Informatica® Intelligent Cloud Services
April 2024

REST API Reference

Informatica Intelligent Cloud Services REST API Reference
April 2024

Table of Contents

Preface

Use *REST API Reference* to learn how to use the Informatica Intelligent Cloud Services℠ REST API to interact with your Informatica Intelligent Cloud Services organization.

Informatica Resources

Informatica provides you with a range of product resources through the Informatica Network and other online portals. Use the resources to get the most from your Informatica products and solutions and to learn from other Informatica users and subject matter experts.

Informatica Documentation

Use the Informatica Documentation Portal to explore an extensive library of documentation for current and recent product releases. To explore the Documentation Portal, visit https://docs.informatica.com.

If you have questions, comments, or ideas about the product documentation, contact the Informatica Documentation team at infa_documentation@informatica.com.

Informatica Intelligent Cloud Services web site

You can access the Informatica Intelligent Cloud Services web site at http://www.informatica.com/cloud. This site contains information about Informatica Cloud integration services.

Informatica Intelligent Cloud Services Communities

Use the Informatica Intelligent Cloud Services Community to discuss and resolve technical issues. You can also find technical tips, documentation updates, and answers to frequently asked questions.

Access the Informatica Intelligent Cloud Services Community at:

https://network.informatica.com/community/informatica-network/products/cloud-integration

Developers can learn more and share tips at the Cloud Developer community:

https://network.informatica.com/community/informatica-network/products/cloud-integration/cloud-developers

Informatica Intelligent Cloud Services Marketplace

Visit the Informatica Marketplace to try and buy Data Integration Connectors, templates, and mapplets:

https://marketplace.informatica.com/

Data Integration connector documentation

You can access documentation for Data Integration Connectors at the Documentation Portal. To explore the Documentation Portal, visit https://docs.informatica.com.

Informatica Knowledge Base

Use the Informatica Knowledge Base to find product resources such as how-to articles, best practices, video tutorials, and answers to frequently asked questions.

To search the Knowledge Base, visit https://search.informatica.com. If you have questions, comments, or ideas about the Knowledge Base, contact the Informatica Knowledge Base team at KB_Feedback@informatica.com.

Informatica Intelligent Cloud Services Trust Center

The Informatica Intelligent Cloud Services Trust Center provides information about Informatica security policies and real-time system availability.

You can access the trust center at https://www.informatica.com/trust-center.html.

Subscribe to the Informatica Intelligent Cloud Services Trust Center to receive upgrade, maintenance, and incident notifications. The Informatica Intelligent Cloud Services Status page displays the production status of all the Informatica cloud products. All maintenance updates are posted to this page, and during an outage, it will have the most current information. To ensure you are notified of updates and outages, you can subscribe to receive updates for a single component or all Informatica Intelligent Cloud Services components. Subscribing to all components is the best way to be certain you never miss an update.

To subscribe, on the Informatica Intelligent Cloud Services Status page, click **SUBSCRIBE TO UPDATES**. You can choose to receive notifications sent as emails, SMS text messages, webhooks, RSS feeds, or any combination of the four.

Informatica Global Customer Support

You can contact a Global Support Center through the Informatica Network or by telephone.

To find online support resources on the Informatica Network, click **Contact Support** in the Informatica Intelligent Cloud Services Help menu to go to the **Cloud Support** page. The **Cloud Support** page includes system status information and community discussions. Log in to Informatica Network and click **Need Help** to find additional resources and to contact Informatica Global Customer Support through email.

The telephone numbers for Informatica Global Customer Support are available from the Informatica web site at https://www.informatica.com/services-and-training/support-services/contact-us.html.

Informatica Intelligent Cloud Services REST API

Use the Informatica Intelligent Cloud Services REST API to access information from your Informatica Intelligent Cloud Services organization. You can also perform tasks such as create, update, and delete connections and configure permissions.

To use the Informatica Intelligent Cloud Services REST API, you need a valid Informatica Intelligent Cloud Services login and an understanding of REST API guidelines.

To configure a request using the REST API, use the appropriate resource and method, along with the applicable objects. Informatica Intelligent Cloud Services returns the requested information, performs the requested task, or returns an error and related messages.

Informatica Intelligent Cloud Services REST API supports the Transport Layer Security (TLS) version 1.2 protocol.

Note that some of the features and functionality mentioned in this guide might not be available to your organization due to licensing.

Platform and service-specific REST APIs

Informatica Intelligent Cloud Services includes common functionality that is the platform on which the services in Informatica Intelligent Cloud Services are built. Each Informatica Intelligent Cloud service has functionality that is only applicable to that service, in addition to platform functionality.

For example, tasks are applicable to most services in Informatica Intelligent Cloud Services. To get a list of tasks in your organization, you use the platform resource, task. A mapping task is a type of task that is only applicable to the Data Integration service. To get details about a mapping task or create a mapping task, you use the Data Integration resource, mttask.

There are two versions of the Informatica Intelligent Cloud Services platform REST API. Use the version that includes the resource you need. You can use both versions in the same session however the base URL and headers are slightly different. For more information, see Chapter 2, "Platform REST API version 2 resources" on page 26 and Chapter 3, "Platform REST API version 3 resources" on page 105.

For information about Data Integration resources, see Chapter 4, "Data Integration REST API" on page 257.

REST API versions

Informatica Intelligent Cloud Services supports the platform REST API version 2 and version 3 resources, and service-specific resources.

You can log in to Informatica Intelligent Cloud Services using the platform REST API version 2 or version 3 login resource. The version of any subsequent resource that you use does not need to match the version of the login resource that you use to log in.

Note the following differences between REST API version 2 and version 3 calls:

Format

You can use the following formats depending upon which API version you use:

- Version 2 supports XML and JSON calls.
- Version 3 supports JSON calls.

Login URL

Use one of the following login URLs:

- For version 2, use `https://dm-<POD region>.informaticacloud.com/ma/api/v2/user/login`.
- For version 3, use `https://dm-<POD region>.informaticacloud.com/saas/public/core/v3/login`.

Your POD (Point of Deployment) region is based on the location of your Informatica Intelligent Cloud Services data center. Use one of the following POD regions:

- For North America, use `us`
- For Europe, use `em`
- For Asia, use `ap`

The POD region is included in the URL you receive when you register with Informatica Intelligent Cloud Services.

Base URL

The login response includes the base URL that you must include in subsequent calls.

The base URL includes the following components:

- The name and region of the POD that your organization uses, for example, usw3.dm-us.
- The Informatica Intelligent Cloud Services domain, informaticacloud.com.
- The internal service that manages the API calls, for example, saas.

The following example is a base URL for an organization on the usw3.dm-us POD:

`https://usw3.dm-us.informaticacloud.com/saas`

In the login response, the attribute that provides the base URL depends on the API version that you use to log in. For example:

- In a version 2 response, the attribute name is `serverUrl`.
- In a version 3 response, the attribute name is `baseApiUrl`.

Request URL

The URL that you use in requests differs between the version 2 and version 3 resources, for example:

- For version 2 resources, use `<serverUrl>/api/v2/<REST API resource>`, for example:

    `https://usw3.dm-us.informaticacloud.com/saas/api/v2/activityLog`

- For version 3 resources, use `<baseApiUrl>/public/core/v3/<REST API resource>`, for example:

  `https://usw3.dm-us.informaticacloud.com/saas/public/core/v3/schedule`

Session ID

The login response includes a session ID that you must include in headers during the session. You can use the same session ID for version 2 and version 3 resources. In the login response, the name of the attribute for session ID depends on the API version that you use to log in. Use one of the following attributes:

- For version 2 resources, use `icSessionId` in the header.

- For version 3 resources, use `INFA-SESSION-ID` in the header.

Header and body configuration

Configure the request header and request body as required, taking into consideration the format of the call and the resource version that you use.

Request header

The request header is slightly different for version 2 and version 3 resources.

For version 2 calls, use the following format in the REST API request header:

```
<METHOD> <serverUrl>/<URI> HTTP/<HTTP version>
Content-Type: application/<json | xml>
Accept: application/<json | xml>
icSessionId: <SessionId>
```

For version 3 calls, use the following format in the REST API request header:

```
<METHOD> <baseApiUrl>/<URI> HTTP/<HTTP version>
Content-Type: application/json
Accept: application/json
INFA-SESSION-ID: <SessionId>
```

Note that if you use a tool such as Postman, requests automatically include the HTTP version. If you enter the HTTP version in the URL, the request is not successful because the HTTP version occurs twice in the URL.

The following list describes the attributes of the version 2 and version 3 request header formats:

Attribute	Required	Description
METHOD	Yes	Method you want to use, such as GET, POST, or DELETE.
serverUrl	Required for most v2 resources	Base URL for all version 2 resources except login and register. Use a placeholder for serverUrl, and replace the placeholder with the Informatica Intelligent Cloud Services URL returned by the login resource. For the login and register resources, use the URL listed in the resource.
baseApiUrl	Required for most v3 resources	Base URL for all version 3 resources except login. Use a placeholder for baseApiUrl, and replace the placeholder with the Informatica Intelligent Cloud Services URL returned by the login resource. For the login resource, use the URL listed in the resource definition.

Attribute	Required	Description
URI	Required for most resources	Resource URI. The URI includes the base URL and the resource name and can also include parameters. For the login and register resources, use the URL listed in the resource definitions.
HTTP version	Yes	HTTP version that you are using. Some tools such as Postman automatically include the HTTP version in the header.
Content-Type	Required for POST requests	Format of the request. Use one of the following options: - application/json. Reads request as JSON. - application/xml. Reads request as XML. Only applicable to version 2 resources. Default is json.
Accept	No	Request format that you want to receive. Use one of the following options: - application/json. Sends response as JSON. - application/xml. Sends response as XML. Only applicable to version 2 resources. Default is json.
icSessionId	Required for most v2 resources	Informatica Intelligent Cloud Services session ID. Required for all version 2 resources except login and register. Use a placeholder for sessionId, and replace the placeholder with the session ID returned by the login resource.
INFA-SESSION-ID	Required for most v3 resources	Informatica Intelligent Cloud Services session ID. Required for all version 3 resources except login. Use a placeholder for sessionId, and replace the placeholder with the session ID returned by the login resource.

Request body

Use the request body to pass additional attributes for the resource. When you pass attributes in a request body, you pass the attributes as part of an object.

For example, to log in with the login resource, you pass the required username and password attributes in a login object.

Some requests include sub-objects for attributes. Declare the sub-objects before listing the related attributes.

JSON format

When you use the JSON format for version 2 REST API calls, you can optionally define a request object with the @type attribute, as shown in the following examples:

```
{
  "@type": "<request object>",
  "<attribute1>": "<value1>",
  "<attribute2>": "<value2>",
}
```

When an attribute includes an object, state the attribute and use the object name as follows:

```
{
  "@type": "<request object>",
  "<attribute1>": "<value1>",
  "<attribute2>": {
     "@type": "<attribute object>",
     "<attributeA>": "<valueA>",
     "<attributeB>": "<valueB>",}
     "@type": "<attribute object>",
     "<attributeD>": "<valueD>",
     "<attributeE>": "<valueE>",}
  "<attribute3>": "<value3>",
}
```

Note: For version 3 REST API calls, do not use the @type attribute.

XML format

When you use the XML format, define a request object as an enclosing set of tags, as follows:

```
<request object>
    <attribute1>value1</attribute1>
    <attribute2>value2</attribute2>
</request object>
```

When an attribute includes an object, enclose the attribute object within the attribute tags as follows:

```
<request object>
    <attribute1>value1</attribute1>
    <attribute2>
      <attribute object>
        <attributeA>valueA</attributeA>
        <attributeB>valueB</attributeB>
      </attribute object>
      <attribute object>
        <attributeC>valueC</attributeC>
        <attributeD>valueD</attributeD>
      </attribute object>
    </attribute2>
    <attribute3>value3</attribute3>
</request object>
```

Return lists

When the REST API returns a series of objects in XML, it encloses the list in the root tag, as follows:

```
<root>
  <return object 1>
    <attribute1>value1</attribute1>
    <attribute2>value2</attribute2>
  </return object 1>
  <return object 2>
    <attribute1>value1</attribute1>
    <attribute2>value2</attribute2>
  </return object 2>
</root>
```

In JSON, no additional attributes are used. The REST API encloses the list in square brackets ([ ]), as follows:

```
[
 {
    "<attribute1>": "<value1>",
    "<attribute2>": "<value2>",
 }{
    "<attribute1>": "<value1>",
    "<attribute2>": "<value2>",
 }
]
```

JSON format example

To log in using JSON, you might use the following request header and body:

```
POST https://dm-us.informaticacloud.com/saas/public/core/v3/login
Content-Type: application/json
Accept: application/json

{
    "username": "user@informatica.com",
    "password": "mypassword"
}
```

The login might return the following information:

```
{
    "products": [
        {
            "name": "Integration Cloud",
            "baseApiUrl": "https://pod.clouddev.informaticacloud.com/saas"
        }
    ],
    "userInfo": {
        "sessionId": "9KA11tLGqxVcGeul8SQBK3",
        "id": "9L1GFroXSDHe2IIg7QhBaT",
        "name": "user",
        "parentOrgId": "52ZSTB0IDK6dXxaEQLUaQu",
        "orgId": "0cuQSDTq5sikvN7x8r1xm1",
        "orgName": "MyOrg_INFA",
        "groups": {},
        "status": "Active"
    }
}
```

You can then use the sessionId and the baseapiUrl to construct a request to obtain your organization's license information, for example:

```
GET https://pod.clouddev.informaticacloud.com/saas/public/core/v3/license/org/{orgId}
Content-Type: application/json
Accept: application/json
INFA-SESSION-ID: IV4wOrJmd6YUtmKa8t
```

XML format example

You can use XML calls with version 2 resources.

To log in using XML, you might use the following header and body:

```
POST https://dm-us.informaticacloud.com/ma/api/v2/user/login
Content-Type: application/xml
Accept: application/xml

<login>
        <username>useremail@company.com</username>
        <password>mypassword</password>
</login>
```

The login might return the following information:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<user>
  <id>00000B03000000000001</id>
  <orgId>00000B</orgId>
  <name>user@company.com</name>
  <createTime>2012-06-14T15:00:00.000Z</createTime>
  <updateTime>2012-06-14T15:00:00.000Z</updateTime>
  <createdBy>System</createdBy>
  <updatedBy>user@company.com</updatedBy>
  <firstName>Firstname</firstName>
```

```
        <lastName>Lastname</lastName>
        <title>Senior Software Engineer</title>
        <password>********</password>
        <phone>111111111111111111111</phone>
        <timezone>America/Los_Angeles</timezone>
        <serverUrl>http://example.informatica.com/saas</serverUrl>
        <icSessionId>IV4wOrJmd6YUtmKa8t</icSessionId>
</user>
```

You can then use the icSessionId and the serverUrl to construct a request to delete a schedule as follows. The schedule ID is 000001D0000000000001.

```
DELETE http://example.informatica.com/saas/api/v2/schedule/000001D0000000000001
Accept: application/xml
icSessionId: IV4wOrJmd6YUtmKa8t
```

Note that Content-Type is not required because the DELETE method does not have additional attributes to pass in the request body.

Update modes

For Data Integration calls, you can submit a POST request using full update mode or partial update mode.

Use partial mode to submit a POST request that only includes the changed object fields, instead of including all of the object fields. For example, if you want to update the connection in an mttask object, you can submit a POST request using partial mode that might look like the following example:

```
POST api/v2/mttask/<taskId>
Content-Type: application/json
Accept: application/json
icSessionId: <icSessionId>
Update-Mode: PARTIAL
{
  "@type": "mtTask",
  "parameters": [
    {
      "@type": "mtTaskParameter",
      "name": "$NewSource$",
      "type": "EXTENDED_SOURCE",
      "sourceConnectionId": "<sourceConnectionId>"
    }
  ]
}
```

If you do not use partial mode, you need to include the entire object in the request. By default, the REST API uses full mode.

Partial mode is available for the following resources:

- connection
- fwConfig
- masterTemplate
- mttask
- schedule
- workflow

When you submit a POST request in partial mode, format the request using JSON and include the following line in the header:

```
Update-Mode=PARTIAL
```

Include the @type attribute for the updated object in the body.

Some fields are grouped in collections. To update a field that resides in a collection, include the key field for the collection in the POST request. The following table lists the collections and corresponding key fields:

Resource/object	Collection	Key Field
fwConfig	fwColumn	name
masterTemplate	mtParameter	name type
mttask	mtTaskInOutParameter	name
mttask	sequenceDefinition	txName
mttask	mtTaskOverriddenField	name
mttask	mtTaskParameter	name type
extendedObject (object in mttask)	objects	name
workflow	workflowTask	taskId

Date/time values

With the REST API, Informatica Intelligent Cloud Services uses the UTC date format to pass all date/time values.

Use the following UTC date format for all date/time values that you pass in requests. The same format is used for all date/time values returned from Informatica Intelligent Cloud Services.

```
<yyyy>-<MM>-<dd>T<HH>:<mm>:<ss>.<SSS>Z
```

The following list describes the attributes of the UTC date format:

yyyy

Year expressed in four digits.

MM

Month expressed in two digits.

dd

Date of the month expressed in two digits.

T

Indicates the time portion of the format.

HH

Hour in the 24-hour format. For example, 0 for 12:00:00 a.m. and 23 for 11:00:00 p.m.

mm

Minutes expressed in two digits.

ss

> Seconds expressed in two digits.

SSS

> Microseconds expressed in three digits.

Z

> UTC time indicator.

For example, the following date string represents 3:00 pm on December 14, 2012:

```
2012-12-14T15:00:00.000Z
```

Object IDs

Many requests require an object ID, such as a connection ID or linear taskflow ID. To find the object ID that you need, you can use the related GET request.

For example, to determine the linear taskflow ID that you need to update a linear taskflow, you can use a workflow GET request to view the details of all linear taskflows in the organization. The return list of linear taskflow details includes the linear taskflow ID. Similarly, to determine the ID of a user, you can perform a user GET request.

Object IDs are not always readily available through the Informatica Intelligent Cloud Services user interface.

Session IDs

When you log in to an Informatica Intelligent Cloud Services organization using the REST API, the login resource returns the REST API session ID. You include this session ID in most subsequent REST API requests during the session. The session ID expires after 30 minutes of inactivity. A grace period of up to 2 minutes might be included to account for any network or unexpected delays.

You can use the same session ID for version 2 and version 3 resources. For example, if you log in using the version 2 login resource, you can use the session ID that was returned in the login response in a request that uses a version 3 resource.

To make a call that uses a REST API version 2 resource, use the icSessionId attribute to include the session ID in the header. To make a call that uses a REST API version 3 resource, use the INFA-SESSION-ID attribute to include the session ID in the header.

The following example shows how icSessionId is used in the header for a REST API version 2 call::

```
GET https://app.informaticacloud.com/saas/api/v2/licenseInfo/org/<id>
Content-Type: application/xml
Accept: application/xml
icSessionId: IV4wOrJmd6YUtmKa8t
```

The following example shows how INFA-SESSION-ID is used in the header for a REST API version 3 call:

```
GET https://app.informaticacloud.com/saas/public/core/v3/license/org/{orgId}
Content-Type: application/json
Accept: application/json
INFA-SESSION-ID: 9KA11tLGqxVcGeul8SQBK3
```

You can submit a POST request to determine the status of a session ID. Use the following URI to submit the request:

```
/api/v2/user/validSessionId
```

Include the following attributes in the request:

- userName. Your Informatica Intelligent Cloud Services user name.
- icToken. The session ID.

For example, you might use the following request:

```
POST https://app.informaticacloud.com/saas/api/v2/user/validSessionId
Content-Type: application/json
Accept: application/json

{
"@type": "validatedToken",
"userName": "user@informatica.com",
"icToken": "9KA11tLGqxVcGeul8SQBK3"
}
```

The response returns whether the session ID is valid or not and the number of minutes left before the session ID expires. For example, you might receive the following response:

```
{
"@type": "validatedToken",
"timeUntilExpire": 29,
"isValidToken": true
}
```

Note: The timeUntilExpire value changes to "0" when the session will expire in less than 1 minute. The isValidToken value changes to "false" when the session expires. To avoid the risk of authentication failure, reauthenticate when the timeUntilExpire value changes to 0.

REST API responses

A successful REST API response returns the requested object, an array of objects, or an HTTP success code. A failed response returns an HTTP error code and can also return an error object.

Successul responses

The following table describes common successful responses to REST API requests:

Request type	Response
GET	For an information request, returns the requested object or an array of objects when applicable. For an action request, returns the HTTP 200 success code. Can also return the REST API success object.
POST	Returns the object that you created or updated. Can also return the HTTP 201 success code.
DELETE	HTTP 200 success code. Can also return the REST API success object.

For example, if you use a GET request to view a schedule, a successful response is the schedule object that you requested. Or, if you use a POST request to update the time that the schedule runs, a successful

response is the schedule object that you updated, including the update. If you use a DELETE request to delete a schedule that is no longer being used, a successful response is the 200 success code.

Failed responses

For a failed response, the REST API returns a standard HTTP error code and can also return an error object. The following table describes some of the HTTP error codes that might be returned:

HTTP error code	Description
400	Bad request. Can be a field or format issue such as an invalid character or incorrect string length.
401	Authentication issue. For example, invalid credentials or invalid session IDs.
403	Generic error. For example, user doesn't have required permissions for a resource.
404	Resource can't be found.
500	Unexpected error. Generic error message that appears when an unexpected condition was encountered.
502	Network issue.
503	Service unavailable. The server cannot handle the request.

Success object

When the REST API successfully performs an action, it returns a 200 or 201 success response. It might also return a success object.

The success object has the following structure:

```
<xs:complexType name="success">
    <xs:sequence>
      <xs:element name="description" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
```

Error object

When the REST API encounters an error, it returns a REST API error object.

For REST API version 2 calls, the error object has the following structure:

```
{
    "code": "UI_10000",
    "description": "User name or password is not valid.",
    "statusCode": 403,
    "@type": "error"
}
```

For REST API version 3 calls, the error object has the following structure:

```
{
    "error": {
        "code": "IDS_085",
        "message": "User name or password is not valid.",
        "requestId": "9hr8e2ObIcChbwYftgDui7",
        "details": null
    }
}
```

REST API rules and guidelines

The following list is a summary of rules and guidelines to follow when working with the Informatica Intelligent Cloud Services REST API:

- Use the login resource to start a REST API session. Reuse the session ID that's returned in the login response in all of the requests that you make during the session.
  You can use a placeholder for the session ID in request headers. Replace the placeholder with the session ID data that's returned in the login response.

- A REST API session expires after 30 minutes of inactivity. However, best practice is to log out before the session expires. To continue work with the REST API, start a new session.

- Do not include icSessionId or INFA-SESSION-ID in the request header for login and register resources.

- Don't log out of a REST API session until all the requests that you've sent have been executed.

- For all resources except login, use a placeholder for the base URL. For version 2 resources, replace the placeholder with the URL returned in the serverUrl. For version 3 resources, replace the placeholder with the URL returned in the baseApiUrl.

- All resources and attributes are case-sensitive.

- Specify the format of the request and response in the header. Use the Content-Type attribute to specify the request format and the Accept attribute to specify the response format.

- If a request or response type isn't configured, Informatica Intelligent Cloud Services uses JSON by default.

- For requests in JSON that use version 2 resources, you can optionally use the @type attribute to define an object. For requests in JSON that use version 3 resources, do not use the @type attribute.

- For requests in XML, use an enclosing <object name> tag to define an object.

- XML responses that include a list of objects return the objects enclosed in the <root> tag.

- For POST requests, you must include all fields in the request object unless you submit the request in JSON format using partial mode. Partial mode is not applicable to most resources. By default, the REST API uses full mode.

- Where indicated, enclose POST request attributes in the specified object. When no object is specified, include attributes in the request body.

Documentation conventions

Informatica Intelligent Cloud Services REST API documentation uses the following conventions:

- Methods are in capital letters, such as GET.

- Request syntax uses the following conventions:
  - Variables are enclosed in angle brackets (< >), such as <id> for a user ID.
  - When listing a choice of attribute values, options are separated by a pipe (|).
  - Optional attributes are in *italics*.

CHAPTER 2

Platform REST API version 2 resources

The REST API version 2 resources in this section apply to multiple services in Informatica Intelligent Cloud Services.

When you use version 2 resources, note the following rules:

- Use JSON or XML format.
- Use the serverUrl value from the login response as the base URL. For example:

  ```
  https://na4.dm-us.informaticacloud.com/saas
  ```
- Use the following URI:

  ```
  /api/v2/<API name>
  ```
- Use the following request header format:

  ```
  <METHOD> <serverUrl>/<URI> HTTP/<HTTP version>
  Content-Type: application/json
  Accept: application/json
  icSessionId: <SessionId>
  ```

In the following example, the serverUrl is `https://na4.dm-us.informaticacloud.com/saas` and the URI is `/api/v2/agent`:

```
<METHOD> https://na4.dm-us.informaticacloud.com/saas/api/v2/agent HTTP/1.1
Content-Type: application/json
Accept: application/json
icSessionId: IV4wOrJmd6YUtmKa8t
```

Note: If you use a tool such as Postman that automatically includes the HTTP version, do not enter the HTTP version in the URL. If the HTTP version appears twice in the URL, the request fails.

activityLog

Use this resource to request log information for completed jobs from the Monitor service. You can also request error logs and session logs. To request log information for jobs that are running, use the activityMonitor resource.

GET Request

You can request all of the log information or filter the log response. To request information from the log, use the following URI:

```
/api/v2/activity/activityLog
```

To request information for a specific log ID, use the following URI:

```
/api/v2/activity/activityLog/<id>
```

To request information for a specific run ID, use the following URI:

```
/api/v2/activity/activityLog?runId=<runId>
```

To request information for a specific task, include the task ID in the following URI:

```
/api/v2/activity/activityLog?taskId=<taskId>
```

To specify the number of rows to skip, use the following URI:

```
/api/v2/activity/activityLog?offset=<offset>
```

To specify a row limit, use the following URI:

```
/api/v2/activity/activityLog?rowLimit=<rowLimit>
```

You can use any combination of these options. For example, you can use the following URI:

```
/api/v2/activity/activityLog?
offset=<offset>&rowLimit=<rowLimit>&taskId=<taskId>&runId=<runId>
```

You can also use the activityLog resource to get a session log. To get a session log, use the following URI:

```
/api/v2/activity/activityLog/<id>/sessionLog
```

You can use the following optional attributes in the activityLog GET URI:

Field	Description
id	Log entry ID. Include this attribute if you want to receive information for a specific ID.
runId	Job ID associated with the log entry ID.
taskId	Task ID associated with the log entry ID. If taskId is not specified, all activityLog entries for all tasks are returned.
offset	The number of rows to skip. For example, you might want to skip the first three rows.
rowLimit	The maximum number of rows to return. The maximum number you can specify is 1000. If you omit this attribute, the activityLog returns all available rows, up to a maximum of 200 rows.

GET Response

Returns an activityLogEntry object for each row in the log or returns an activityLogEntry object for the specified ID. Returns the error object if errors occur.

When you request information for each row in the log, the activityLogEntry object includes the following attributes:

Field	Type	Description
id	String	Log entry ID.
type	String	The type of task. For Data Integration, returns one of the following codes: - DMASK. Masking task. - DRS. Replication task. - DSS. Synchronization task. - MTT. Mapping task. - PCS. PowerCenter task. - WORKFLOW. Linear taskflow.
objectId	String	Task ID.
objectName	String	Name of the task.
runId	Long	ID for the task run.
agentId	String	Agent that runs the task.
runtimeEnvironmentId	String	Runtime environment where the task runs.
startTime	Date/time	Start time for the task or linear taskflow. Uses Eastern Time Zone (ET).
endTime	Date/time	End time for the task or linear taskflow. Uses Eastern Time Zone (ET).
startTimeUtc	Date/time	Start time for the task or linear taskflow. Uses Coordinated Universal Time (UTC).
endTimeUtc	Date/time	End time for the task or linear taskflow. Uses Coordinated Universal Time (UTC).
state	Short	Whether the task completed successfully. Returns one of the following codes: - 1. The task completed successfully. - 2. The task completed with errors. - 3. The task failed to complete.
failedSourceRows	Long	Number of rows that were not read from the source.
successSourceRows	Long	Number of rows that were successfully read from the source.
failedTargetRows	Long	Number of rows that were not written to the target.
successTargetRows	Long	Number of rows that were successfully written to the target.
scheduleName	String	Schedule name, if task was initiated by a schedule.
entries		Indicates the start of information for a child object.

When you request log information for a specific ID, the activityLogEntry object includes the following attributes:

Field	Type	Description
id	String	Log entry ID.
type	String	The type of task. For Data Integration, returns one of the following codes: - DMASK. Masking task. - DRS. Replication task. - DSS. Synchronization task. - MTT. Mapping task. - PCS. PowerCenter task. - WORKFLOW. Linear taskflow.
objectId	String	Task ID.
objectName	String	Name of the task.
runId	String	ID for the task run.
agentId	String	Agent that runs the task.
runtimeEnvironmentId	String	Runtime environment where the task runs.
startTime	Date/time	Start time for the task or linear taskflow. Uses Eastern Time Zone (ET).
endTime	Date/time	End time for the task or linear taskflow. Uses Eastern Time Zone (ET).
startTimeUtc	Date/time	Start time for the task or linear taskflow. Uses Coordinated Universal Time (UTC).
endTimeUtc	Date/time	End time for the task or linear taskflow. Uses Coordinated Universal Time (UTC).
state	Short	Whether the task completed successfully. Returns one of the following codes: - 1. The task completed successfully. - 2. The task completed with errors. - 3. The task failed to complete. - 4. The task has not started.
UIState	Short	Used for internal purposes only.
isStopped	Boolean	Whether the task was stopped by the user. A value of `true` is possible if the task completed with errors or failed to complete.
failedSourceRows	Long	Number of rows that were not read from the source.
successSourceRows	Long	Number of rows that are successfully read from the source.
failedTargetRows	Long	Number of rows that were not written to the target.
successTargetRows	Long	Number of rows that were successfully written to the target.
errorMsg	String	Error message associated with the job.
startedBy	String	User who started the task.

Field	Type	Description
runContextType	String	Method through which the task was initiated. Includes the following values: - UI. Task was initiated through the user interface. - SCHEDULER. Task was initiated through the task scheduler. - REST-API. Task was initiated through the REST API. - OUTBOUND MESSAGE. Task was initiated through an outbound message.
scheduleName	String	Schedule name, if task was initiated by a schedule.
orgId	String	Organization ID.
totalSuccessRows	Long	Total number of rows that were successfully read from the source and written to the target.
totalFailedRows	Long	Total number of rows that were not read from the source and written to the target.
logFilename	String	The name of the generated log file.
errorFilename	String	The name of the generated error file.
errorFileDir	String	The location of the error file on the Secure Agent machine.
connType	String	Connection type.
stopOnError	Boolean	Determines the runtime environment action to take when an nonfatal error occurs. Includes the following values: - True. The linear taskflow stops when an error occurs. - False. The linear taskflow continues to process when an error occurs.
items		Includes an activityLogEntryItem object for each task.
type	String	Included in the activityLogEntryItem object. The type of task. For Data Integration, returns one of the following codes: - DMASK. Masking task. - DRS. Replication task. - DSS. Synchronization task. - MTT. Mapping task. - PCS. PowerCenter task. - WORKFLOW. Linear taskflow.
objectId	String	Included in the activityLogEntryItem object. Task ID.
objectName	String	Included in the activityLogEntryItem object. Name of the task.
runId	String	Included in the activityLogEntryItem object. ID for the task run.
agentId	String	Included in the activityLogEntryItem object. Agent that ran the task.

Field	Type	Description
runtimeEnvironmentId	String	Included in the activityLogEntryItem object. Runtime environment where the task ran.
startTime	Date/time	Included in the activityLogEntryItem object. Start time for the task or linear taskflow. Uses Eastern Time Zone (ET).
endTime	Date/time	Included in the activityLogEntryItem object.End time for the task or linear taskflow. Uses Eastern Time Zone (ET).
state	String	Included in the activityLogEntryItem object.Whether the task completed successfully. Returns one of the following codes: - 1. The task completed successfully. - 2. The task completed with errors. - 3. The task failed to complete.
isStopped	Boolean	Whether the task was stopped by the user. A value of `true` is possible if the task completed with errors or failed to complete.
errorMsg	String	Included in the activityLogEntryItem object. Error message associated with the job.
connType	String	Included in the activityLogEntryItem object. Connection type.
children	String	Included in the activityLogEntryItem object. Returns an activityLogEntryItem object for each table in the linear taskflow.
startedBy	String	User who started the task.
runContextType	String	Method through which the task was initiated. Includes the following values: - UI. Task was initiated through the user interface. - SCHEDULER. Task was initiated through the task scheduler. - REST-API. Task was initiated through the REST API. - OUTBOUND MESSAGE. Task was initiated through an outbound message.
scheduleName	String	Schedule name, if task was initiated by a schedule.
transformationEntries		Includes information in a transformationLogEntry object for each transformation.
id	String	Included in the transformationLogEntry object. Transformation ID.
txName	String	Included in the transformationLogEntry object. Transformation name. For target transformations, returns the target object name.
txType	String	Included in the transformationLogEntry object. Transformation type.
successRows	Long	Included in the transformationLogEntry object. Number of successful rows for the transformation.

Field	Type	Description
affectedRows	Long	Included in the transformationLogEntry object. Number of rows affected by the defined task operation. Applies to target transformations in data transfer tasks and mapping tasks based on mappings that are not configured for full SQL ELT optimization. Doesn't apply to mappings in SQL ELT mode or advanced mappings.
failedRows	Long	Included in the transformationLogEntry object. Number of failed rows for the transformation.
sequenceValues		Returns information generated from a task that includes the sequence generator transformation. Includes a sequenceValueLogEntry object for each transformation.
txName	String	Included in the sequenceValueLogEntry object. Transformation name. For target transformations, returns the target object name.
nextValue	String	Included in the sequenceValueLogEntry object. The last value generated by the task.
inOutParameterValues		The in-out parameter values used in the task. Includes an inOutParameterValueLogEntry for each parameter.
name	String	Included in the inOutParameterValueLogEntry object. Parameter name.
value	String	Included in the inOutParameterValueLogEntry object. Parameter value.

GET Example

To request 20 rows of information returned from the log in JSON format, you might use the following request:

```
GET <serverUrl>/api/v2/activity/activityLog?rowLimit=20 HTTP/1.0
Accept:application/json
icSessionId: <icSessionId>
```

A successful request returns a list: an activityLogEntry object for each entry returned from the log.

The following text is a sample return in JSON:

```
[
    {
        "@type": "activityLogEntry",
        "id": "000001C100000000000C",
        "type": "DRS",
        "objectName": "drstask1",
        "runId": 1,
        "runtimeEnvironmentId": "00000C25000000000002",
        "startTime": "2012-07-30T13:30:30.000Z",
        "endTime": "2012-07-30T13:32:30.000Z",
        "state": 1,
        "failedSourceRows": 0,
        "successSourceRows": 39,
        "failedTargetRows": 0,
        "successTargetRows": 39,
        "errorMsg": null,
        "entries": [
```

```
                            {
                                "@type": "activityLogEntry",
                                "id": "128964732",
                                "type": "DRS",
                                "objectName": "Contact",
                                "runId": 0,
                                "runtimeEnvironmentId": "00000C25000000000002",
                                "agentId: "01000008000000000006",
                                "startTime": "2012-07-30T13:32:31.000Z",
                                "endTime": "2012-07-30T13:35:31.000Z",
                                "state": 1,
                                "isStopped": FALSE,
                                "failedSourceRows": 0,
                                "successSourceRows": 39,
                                "failedTargetRows": 0,
                                "successTargetRows": 39,
                                "errorMsg": "No errors encountered.",
                                "entries": []
                            },
                    ]
            },
            {
                "@type": "activityLogEntry",
                "id": "010000C1000000000PGP",
                "type": "MTT_TEST",
                "objectId": "0100000Z00000000001N",
                "objectName": "Mapping-MultiSource",
                "runId": 12,
                "startTime": "2020-03-27T08:05:56.000Z",
                "endTime": "2020-03-27T08:06:07.000Z",
                "startTimeUtc": "2020-03-27T12:05:56.000Z",
                "endTimeUtc": "2020-03-27T12:06:07.000Z",
                "state": 2,
                "failedSourceRows": 0,
                "successSourceRows": 800,
                "failedTargetRows": 200,
                "successTargetRows": 600,
                "startedBy": "di@infa.com",
                "runContextType": "ICS_UI",
                "entries": [
                        {
                            "@type": "activityLogEntry",
                            "id": "118964723",
                            "type": "MTT_TEST",
                            "objectName": "",
                            "runId": 12,
                            "agentId": "01000008000000000004",
                            "runtimeEnvironmentId": "01000025000000000004",
                            "startTime": "2020-03-27T08:05:56.000Z",
                            "endTime": "2020-03-27T08:06:07.000Z",
                            "startTimeUtc": "2020-03-27T12:05:56.000Z",
                            "endTimeUtc": "2020-03-27T12:06:07.000Z",
                            "state": 2,
                            "failedSourceRows": 0,
                            "successSourceRows": 800,
                            "failedTargetRows": 200,
                            "successTargetRows": 600,
                            "errorMsg": null,
                            "startedBy": "di@infa.com",
                            "runContextType": "ICS_UI",
                            "entries": [],
                            "subTaskEntries": [],
                            "logEntryItemAttrs": {
                                "CONSUMED_COMPUTE_UNITS": "0.0",
                                "ERROR_CODE": "0",
                                "IS_SERVER_LESS": "false",
                                "REQUESTED_COMPUTE_UNITS": "0.0",
                                "Session Log File Name": "s_mtt_0Sr7LdcbAG2ldG33Lp8koQ_2.log"
                            },
                            "totalSuccessRows": 0,
```

```
                        "totalFailedRows": 0,
                        "stopOnError": false,
                        "hasStopOnErrorRecord": false,
                        "contextExternalId": "0100000Z00000000001N",
                        "transformationEntries": [
                            {
                                "@type": "transformationLogEntry",
                                "id": "141332309",
                                "txName": "FFSource2",
                                "txType": "SOURCE",
                                "successRows": 600,
                                "failedRows": 0
                            },
                            {
                                "@type": "transformationLogEntry",
                                "id": "141332310",
                                "txName": "FFSource1",
                                "txType": "SOURCE",
                                "successRows": 200,
                                "failedRows": 0
                            },
                            {
                                "@type": "transformationLogEntry",
                                "id": "141332311",
                                "txName": "FFTarget.csv",
                                "txType": "TARGET",
                                "successRows": 600,
                                "affectedRows": 600,
                                "failedRows": 0
                            },
                            {
                                "@type": "transformationLogEntry",
                                "id": "141332312",
                                "txName": "MYSQLTarget",
                                "txType": "TARGET",
                                "successRows": 0,
                                "affectedRows": 0,
                                "failedRows": 200
                            }
                        ]
                    }
                ]
            }
    ]
```

Error Log Requests

You can request an error log from the server.

To request an error log from the server for a specific log ID, use the following URI:

```
/api/v2/activity/errorLog/id
```

To retrieve an error log from the server, you might use the following request:

```
GET <server URL>/api/v2/activity/errorLog/000002C10000000002BG  HTTP/1.0
Accept:application/json
icSessionId: <icSessionId>
```

The server returns the error log as a string, as shown in the following example:

```
"Col1","Col2","Error"

"05/11/2015 00:00:00.000000000","05/11/2015 00:00:00.000000000","ERROR: Target table
[test] has no keys specified."

"05/11/2015 00:00:00.000000000","05/11/2015 00:00:00.000000000","ERROR: Target table
[test] has no keys specified."
```

```
"05/11/2015 00:00:00.000000000","05/11/2015 00:00:00.000000000","ERROR: Target table
[test] has no keys specified."

"05/11/2015 00:00:00.000000000","05/11/2015 00:00:00.000000000","ERROR: Target table
[test] has no keys specified."

"05/11/2015 00:00:00.000000000","05/11/2015 00:00:00.000000000","ERROR: Target table
[test] has no keys specified."
```

Session Log Requests

You can download session logs for all task types using the sessionLog API. For tasks that have subtasks such as replication tasks and linear taskflows, you can download a ZIP file that contains all of the session logs in the hierarchy. For replication tasks, which have two levels of tasks, you can specify an itemId to return a session log for a subtask if you do not want all of the session logs. For linear taskflows, which have three levels of tasks, you can specify an itemId or childItemId to return a session log for a particular subtask.

Use the following URI to download session logs:

```
/saas/api/v2/activity/activityLog/<Top_Level_Log_Entry_Id>/sessionLog?itemId=<child-log-
entry-item-id>&childItemId=<child-log-entry-item-id>
```

The following example requests include a request for a specific session log and requests for session logs for subtasks:

- To request a session log, which may return a ZIP file if the task is a replication task or linear taskflow, you might use the following request:

  ```
  /saas/api/v2/activity/activityLog/000001C1000000000591/sessionLog
  ```

- To request a session log for a particular subtask for a replication task or linear taskflow, you might use the following request:

  ```
  /saas/api/v2/activity/activityLog/000001C1000000000591/sessionLog?itemId=233
  ```

- To request a session log for a sub-subtask in a linear taskflow, you might use the following request:

  ```
  /saas/api/v2/activity/activityLog/000001C1000000000591/sessionLog?
  itemId=233&childItemId=234
  ```

activityMonitor

Use this resource to request log information for running jobs from the Monitor service. To request log information for completed jobs, use the activityLog resource.

GET Request

To request log information about running jobs, use the following URI:

```
/api/v2/activity/activityMonitor?details=<true|false>
```

You can use the following attribute in the activityMonitor GET URI:

details

Optional.

Log detail to be returned from Informatica Intelligent Cloud Services. Use one of the following options:

- true. Returns log information for tasks, linear taskflows, and child objects. Child objects can include tasks within linear taskflows, and objects within replication tasks.

- false. Returns log information for tasks and linear taskflows.

Default is false. If you omit this optional attribute, Monitor does not return additional details.

GET Response

Returns an activityMonitorEntry object for each row in the log. Returns the error object if errors occur.

The activityMonitorEntry object includes the following GET response attributes:

Field	Type	Description
id	String	Log entry ID.
type	String	The type of task. Returns one of the following codes: - DMASK. Masking task. - DRS. Replication task. - DSS. Synchronization task. - MTT. Mapping task. - PCS. PowerCenter task. - WORKFLOW. Linear taskflow.
taskId	String	Task ID.
taskName	String	Task name.
objectName	String	Source object used in the task, or the replication object being processed.
runId	Long	ID for the task run.
startTime	Date/time	Start time for the task or linear taskflow. Uses Eastern Time Zone (ET).
endTime	Date/time	End time for the task or linear taskflow. Uses Eastern Time Zone (ET).
executionState	String	State of the task. Returns one of the following codes: - QUEUED - INITIALIZED - RUNNING - STOPPING - FAILED FAILED can be returned for linear taskflow subtasks only.
failedSourceRows	Long	Number of rows that were not read from the source.
successSourceRows	Long	Number of rows that were successfully read from the source.
failedTargetRows	Long	Number of rows that were not written to the target.
successTargetRows	Long	Number of rows that were successfully written to the target.
errorMsg	String	Error message associated with the job.
entries		Indicates the start of information for a child object. A child object might be a task within a linear taskflow, or an object in a replication task.
agentId	String	Agent used for the activity.
runtimeEnvironmentId	String	Runtime environment used for the activity.
startedBy	String	User who started the task.

Field	Type	Description
runContextType	String	Method through which the task was initiated. Includes the following values: - UI. Task was initiated through the Data Integration user interface. - SCHEDULER. Task was initiated through the task scheduler. - REST-API. Task was initiated through the REST API. - OUTBOUND MESSAGE. Task was initiated through an outbound message.
scheduleName	String	Schedule name, if task was initiated by a schedule.
callbackURL	String	Status of the job.

GET Example

To return log information including details about child objects in XML, you might use the following request:

```
GET <serverUrl>/api/v2/activity/activityMonitor?details=true
Content-Type: application/xml
Accept: application/xml
icSessionId: <icSessionId>
```

A successful request returns an activityMonitorEntry object for each item returned from Monitor.

The following text is a sample return in XML:

```
<root>
 <activityMonitorEntry>
   <id>000001C100000000000D</id>
   <type>DSS</type>
   <objectName>dss-f2f</objectName>
   <runId>0</runId>
   <startTime>2012-07-30T13:30:00.000Z</startTime>
   <endTime></endTime>
   <executionState>RUNNING</executionState>
   <failedSourceRows>0</failedSourceRows>
   <successSourcerows>938</successSourceRows>
   <failedTargetRows>0</failedTargetRows>
   <successTargetRows>596</successTargetRows>
   <errorMsg> </errorMsg>
   <entries> </entries>
   <agentId>00000C08000000000003</agentId>
   <runtimeEnvironmentId>00000C25000000000002</runtimeEnvironmentId>
 </activityMonitorEntry>
 <activityMonitorEntry>
   <id>000001C500000000000L</id>
   <type>PCS</type>
   <objectName>pcs-lookup</objectName>
   <runId>2</runId>
   <startTime>2012-07-30T13:30:03.001Z</startTime>
   <endTime>2012-07-30T13:30:03.010Z</endTime>
   <executionState>COMPLETE</executionState>
   <failedSourceRows>0</failedSourceRows>
   <successSourcerows>688</successSourceRows>
   <failedTargetRows>0</failedTargetRows>
   <successTargetRows>688</successTargetRows>
   <errorMsg> </errorMsg>
   <entries> </entries>
   <agentId>00000C08000000000003</agentId>
   <runtimeEnvironmentId>00000C25000000000002</runtimeEnvironmentId>
 </activityMonitorEntry>
</root>
```

agent

Use this resource to receive an install token to register a Secure Agent, download the checksum of the agent installation program, request the details about Informatica Cloud Secure Agents or Secure Agent services, or delete a Secure Agent.

GET request for Secure Agent install token and checksum

To request an install token so that you can complete the Secure Agent registration process or to get the checksum of the agent installation program, include the platform type in the URI as follows:

```
/api/v2/agent/installerInfo/<platform>
```

Use one of the following values for the platform:

- win64
- linux64

GET response for Secure Agent install token and checksum

A successful request returns the download URL, install token, and checksum download URL, as shown in the following sample response:

```
{
    "@type": "agentInstallerInfo",
    "downloadUrl": "https://pdm.ics.dev/package-manager/files/binary/
agent64_install_ng_ext/6403/win64/agent64_install_ng_ext.6403.exe",
    "installToken":
"PJ7NVrQ0SGpnpbmJ8K5yte18HLDw305DwPgP_jxG1R4KiOY9BL6qxV7jWiv7wSEfg7mQHKRWX6kcEVph1xjswX",
    "checksumDownloadUrl": "https://pdm.ics.dev/package-manager/files/binary/
agent64_install_ng_ext/6403/win64/agent64_install_ng_ext.6403_win64.sha256"
}
```

To verify the checksum on Windows, use a third-party utility, for example, HashMyFiles or 7-Zip, to find the checksum for the Secure Agent installation program and compare it to the checksum in the `checksum.txt` file. The hashing algorithm is CRC-32.

To verify the checksum on Linux, run the command `cksum <Secure Agent installer filename>` and compare the checksum to the checksum in the `checksum.txt` file. The first column of the command output is the checksum.

GET request for agent details

You can request the details about Secure Agents or details about the services that run on Secure Agents.

Secure Agent details

To request the details about all Secure Agents in the organization, use the following URI:

```
/api/v2/agent
```

To request a list of all the Secure Agents that are currently not assigned to any group, use the following URI:

```
/api/v2/agent/?includeUnassignedOnly=true
```

To request the details about a particular Secure Agent, you can include the Secure Agent ID or the Secure Agent name in the URI. Use one of the following URIs:

```
/api/v2/agent/<id>
```

```
/api/v2/agent/name/<name>
```

If you use the Secure Agent name in the request and the Secure Agent name includes a space, replace the space with `%20`. For example:

```
/api/v2/agent/name/special%20agent
```

Secure Agent services details

To request the details about the services that run on all of the Secure Agents in the organization, use the following URI:

```
/api/v2/agent/details
```

To request the details about the services that run on a particular Secure Agent, include the agent ID in the URI as follows:

```
/api/v2/agent/details/<id>
```

GET response for agent details

Returns the agent object for the requested Secure Agent ID or Secure Agent name.

If you request information for all Secure Agents in the organization, returns an agent object for each Secure Agent in the organization.

If you request details for agent services, returns an AgentEngine object in addition to the agent object.

Returns the error object if errors occur.

The agent object includes the following attributes:

Field	Type	Description
id	String	Secure Agent ID.
orgId	String	Organization ID.
name	String	Secure Agent name.
description	String	Description of the Secure Agent.
createTime	Date/time	Time the Secure Agent was created.
updateTime	Date/time	Last time the Secure Agent was updated.
createdBy	String	User who created the Secure Agent.
updatedBy	String	User who updated the Secure Agent.
active	Boolean	Whether the Secure Agent is active. Returns true or false.
readyToRun	Boolean	Whether the Secure Agent is ready to run a task. Returns true or false.
platform	String	Platform of the Secure Agent machine. Returns one of the following values: - win64 - linux64
agentHost	String	Host name of the Secure Agent machine.
proxyHost	String	Host name of the outgoing proxy server that the Secure Agent uses.
proxyPort	Int	Port number of the outgoing proxy server.

Field	Type	Description
proxyUser	String	User name to connect to the outgoing proxy server.
agentVersion	String	Secure Agent version.
spiUrl	String	This field is no longer applicable and has been deprecated.
upgradeStatus	String	Upgrade status.
lastUpgraded	Date/time	Last time the Secure Agent was upgraded.
lastUpgradeCheck	Date/time	Last time the Secure Agent was checked for upgrade.
lastStatusChange	Date/time	Last time the Secure Agent status was updated.
packages	String	This field is no longer applicable and has been deprecated.
configUpdateTime	Date/time	Last time a user updated Secure Agent properties.
agentGroupId	String	ID of the Secure Agent group.
agentConfigs		This object is no longer applicable and has been deprecated.
name	String	Included in the agentConfig object. Deprecated.
type	String	Included in the agentConfig object. Deprecated.
subtype	String	Included in the agentConfig object. Deprecated.
value	String	Included in the agentConfig object. Deprecated.
customized	Boolean	Included in the agentConfig object. Deprecated.
overridden	Boolean	Included in the agentConfig object. Deprecated.
defaultValue	String	Included in the agentConfig object. Deprecated.
platform	String	Included in the agentConfig object. Deprecated.

If you request details for the services that run on Secure Agents, the agent object also includes the AgentEngine object. The AgentEngine object includes the following attributes:

Field	Type	Description
agentEngineStatus		Status of the agent service, which includes information in the AgentEngineStatus object.
appname	String	Included in the AgentEngineStatus object. The service name that is used internally.
appDisplayName	String	Included in the AgentEngineStatus object. The service name that displays in the user interface.
appversion	String	Included in the AgentEngineStatus object. The service version. The version number changes each time you modify the service.
replacePolicy	-	Used for internal purposes only.
status	String	Included in the AgentEngineStatus object. The status of the service. Possible values include: - RUNNING. The service is running and ready to accept jobs. - DEPLOYING: The service is being provisioned. - STOPPED. The service has been stopped. - ERROR. The service is in an error state. - NEED_RUNNING. The start process is set to begin or has started. - NEED_STOP. The stop process is set to begin or has started.
desiredStatus	-	Used for internal purposes only.
subState	String	Included in the AgentEngineStatus object. A value of 0 indicates that the service is operational. All other values indicate that the service is not operational.
createTime	Date/time	Included in the AgentEngineStatus object. The time the service was created.
updateTime	Date/time	Included in the AgentEngineStatus object. The last time the service was updated.
agentEngineConfigs		Defines agent service properties. Includes information in an engineConfig object for each agent service property.
type	String	Included in the engineConfig object. Configuration type.
name	String	Included in the engineConfig object. Configuration property name.
value	String	Included in the engineConfig object. Value of the property.

Field	Type	Description
platform	String	Included in the engineConfig object. Platform. Returns one of the following values: - win64 - linux64
customized	Boolean	Included in the engineConfig object. Whether the property is in the custom configuration details. Returns true or false.

A successful response might look similar to the follow example:

GET details example

To request the details about the Secure Agent with an ID of 10044030000000000GC, to be returned in JSON format, you might use the following request:

```
GET <serverUrl>/api/v2/agent/10044030000000000GC
Accept:application/json
icSessionId: <icSessionId>
```

The following example shows a successful response:

```
{
"@type": "agent",
"id": "10044030000000000GC",
"orgId": "010025",
"name": "MyAgent",
"createTime": "2021-02-25T00:42:39:000Z",
"updateTime": "2021-02-25T00:42:39:000Z",
"createdBy": "larry104",
"updatedBy": "larry104",
"active": "false",
"readyToRun": "false",
"platform": "linux64",
"agentHost": "agentHost5",
"serverUrl": "https://na4.dm-us.informaticacloud.com/saas",
"proxyPort": "0",
"upgradeStatus": "NotUpgrading",
"federatedId": "6iPQuOsH1YAfnJvhZWPZjI",
"createTimeUTC": "2021-02-25T00:42:39:000Z",
"updateTimeUTC": "2021-02-25T00:42:39:000Z",
"agentGroupId": "01000125000000000002"
}
```

DELETE request

You can delete a Secure Agent if it is not associated with any connections. Before you delete a Secure Agent, update associated connections to use another Secure Agent.

To delete a Secure Agent, use the Secure Agent ID in the following URI:

```
/api/v2/agent/<id>
```

DELETE response

Returns the 200 response code if the request is successful.

Returns the error object if errors occur.

auditlog

Use this resource to request entries from the audit log.

GET Request

To request the most recent 200 entries in the audit log, use the following URI:

```
/api/v2/auditlog
```

To request a specific batch of audit log entries, define the batch size and request a batch number with the following URI:

```
/api/v2/auditlog?batchId=<batchId>&batchSize=<batchSize>
```

Include the following information in the GET URI:

Field	Required	Description
batchSize	Yes	Number of entries to include in a batch.
batchId	Yes	The batch that you want to view. Use 0 for the first batch, which contains the most recent audit log entries. For example, to view entries 26-50, use a batch size of 25, and request batch 1.

GET Response

Returns an auditLogEntry object for each audit log entry returned. Returns the error object if errors occur.

The auditLogEntry object includes the following attributes:

Field	Type	Description
id	String	Audit log entry ID.
version	Int	Version.
orgId	String	Organization ID.
username	String	User who performed the action.
entryTime	Date/time	Time the action occurred. Uses Eastern Time Zone (ET).
entryTimeUTC	Date/time	Time the action occurred. Uses Coordinated Universal Time (UTC).
objectId	String	ID of the object used.
objectName	String	Name of the object used.

Field	Type	Description
category	String	Category of audit log entry. Returns one of the following codes: - AUTH. Authorization. - AGREEMENT. Subscription agreement. - SYSTEM_INFO - ADMIN_REPORT - ORG. Organization. - USER - AGENT. Secure Agent. - CONNECTION - SCHEDULE - DRS. Replication. - DQA. Data assessment. - DMASK.Masking. - DSS. Synchronization. - DATA_FILE. File. - WORKFLOW. Linear taskflow. - PCS. PowerCenter. - MTT. Mapping task. - MI_TASK. File ingestion task. - CUSTOM_FUNC. Mapplet. - MIGRATE. Migration. - CUSTOM_SOURCE. Saved query. - SUBSCRIPTION_BILLING - USER_GROUP - SUB_ORG. Sub-organization. - OBJECT_ACL. Object permissions. - PACKAGE - TEMPLATE. Visio template. - DTEMPLATE. Mappings. - CONNECTOR. Informatica Cloud Connector. - EDITION. Informatica Cloud edition. - SCHEDULE_BLACKOUT. Schedule blackout period. - EXT_CONNECTION. Connections stored on a local Secure Agent. - BUNDLE. - ORG_EDITION. Information about changes to organization edition association. For example, when the organization is reassigned a new edition. - RUNTIME_ENVIRONMENT. - B2BGW_MONITOR - B2BGW_CUSTOMER - B2BGW_SUPPLIER - TASKFLOW - PROCESS - GUIDE - AI_CONNECTION - AI_SERVICE_CONNECTOR - PROCESS_OBJECT - DOMAIN

Field	Type	Description
event	String	Type of action performed. Returns one of the following codes: - CREATE - UPDATE - DELETE - DISABLE - RUN - VERSION1 - VERSION2 - VERSION3 - VERSION4 - VERSION5 - VERSION6 - VERSION7 - DOWNLOAD - EXPORT - IMPORT - FETCHSTATE - LOADSTATE - MAKE_DEFAULT - LINK - ENCRYPT - MOVE_CONNS_TO_AGENT - MOVE_CONNS_TO_IOD - STOP - REGISTER - EDIT - DEREGISTER - RETRY_REGISTRATION - RECONCILE
eventParam	String	Objects related to the action. **Note:** The value can be a maximum of 1024 characters. If the value exceeds 1024 characters, it is truncated and prefixed with "Full Value Truncated."
message	String	Additional information.

GET Example

To view rows 21-40, you might use the following URI:

```
/api/v2/auditlog?batchId=1&batchSize=20
```

bundleObject

Use this resource to request the details for a specific bundle or the details for all bundles published by the organization or installed by the organization. You can also push a published private bundle to sub-organizations.

GET Request

To request the details of a particular bundle, you can include the bundle ID or the bundle name in the URI. Use one of the following URIs:

```
/api/v2/bundleObject/<id>
```
```
/api/v2/bundleObject/name/<name>
```

If you use the bundle name in the URI and the bundle name includes a space, replace the space with `%20`. For example:

```
/api/v2/bundleObject/name/first%20bundle
```

To request the details for all bundles published by the organization, use one of the following URIs:

```
/api/v2/bundleObject/?published=true
```
```
/api/v2/bundleObject/?published=true&installed=false
```

To request the details for all bundles installed by the organization, use one of the following URIs:

```
/api/v2/bundleObject/?installed=true
```
```
/api/v2/bundleObject/?published=false&installed=true
```

GET Response

When you request the details for a bundle, returns the bundleObject for the bundle.

When you request a list of published bundles, returns a bundleObject for each bundle that the organization published.

When you request a list of installed bundles, returns a bundleObject for each bundle that the organization installed.

Returns the error object if errors occurred.

The bundleObject includes the following attributes:

Field	Type	Description
id	String	Bundle ID.
orgId	String	Organization ID.
name	String	Bundle name.
description	String	Description.
createTime	Date/time	Time the bundle was created.
updateTime	Date/time	Time the bundle was updated.
createdBy	String	User who created the bundle.

Field	Type	Description
updatedBy	String	User who last updated the bundle.
lastVersion	String	The current published version of the bundle.
revokeTime	Date/time	This attribute is not used at this time.
paid	Boolean	Whether the bundle was purchased. Returns true for paid, false for free.
copyable	Boolean	Determines whether users can download the contents of the bundle locally. Returns true or false.
accessType	String	Access type for the bundle. Returns the following codes in the BundleObjectAccessType object: - PUBLIC. Available to all Informatica Intelligent Cloud Services organizations. - SUBORGS. Available to sub-organizations of the publishing organization. - ACCESS_LIST. Available to the organization IDs in the sharedWith attribute.
objects		Objects in the bundle. Includes information for each object in the bundleRefObject object.
objectTypeCode	String	Included in the bundleRefObject object. The type of bundle. Includes the following values: - 0V. Visio template. - 17. Mapping. - 0L. Mapplet.
objectId	String	Included in the bundleRefObject object. Object identified in the bundle.
objectName	String	Included in the bundleRefObject object. Name of the object in the bundle.
objectUpdateTime	String	Included in the bundleRefObject object. The date and time that the object in the bundle was last updated.
publishOrgId	String	ID of the organization that published the bundle.
publishOrgName	String	Name of the organization that published the bundle.
externalId	String	External ID for the bundle.

POST Request

As part of a parent organization, you can share a private bundle with sub-organizations.

You can push a published private bundle to install the bundle on all sub-organizations. Push a published private bundle when you want the objects in the bundle to be immediately available to all sub-organizations.

To push a bundle to a sub-organization, use the ID of the bundle object in the following URI:

```
/api/v2/bundleObject/push/<bundleId>
```

POST Response

Returns the success response if the request is successful. Returns the error object if errors occur.

bundleObjectLicense

Use this resource to request license information about bundles installed on or available to the organization. You can also install a bundle and uninstall a bundle.

GET Request

To request license information for a bundle associated with to the organization, use the bundle ID in the following URI:

```
/api/v2/bundleObjectLicense/<bundleObjectId>
```

To request license information for all bundles associated with the organization, omit the optional bundle ID.

GET Response

If successful, returns the BundleObjectLicenseType for the requested bundle.

If you request license information for all bundles, returns the bundleObjectLicense object for all bundles associated with the organization.

Returns the error object if errors occur.

The bundleObjectLicense object includes the following attributes:

Field	Type	Description
bundleObjectId	String	Bundle ID.
orgId	String	Organization ID.
updateOption	String	This attribute is not used at this time.
licenseType	String	Bundle type. Returns one of the following values: - Free - Trial - Subscription
endDate	Date/time	Date the license expires. Returns NULL for free public bundles.
numberOfDaysToApply	Int	Not used at this time.
numberOfMonthsToApply	Int	Not used at this time.
beginDate	Date/time	Publish date for the bundle.
bundleVersion	String	Version number for the bundle.
createTime	Date/time	Creation date for the bundle.
installed	Boolean	Indicates if the organization installed the bundle. Returns TRUE for installed bundles and FALSE for available bundles.
active	Boolean	Indicates that the bundle is available and active. Returns TRUE.
accessCode	String	Required to install a licensed bundle. Used for sharing private bundles. Read only.

POST Request

To install a bundle on the organization, use the following URI:

```
/api/v2/bundleObjectLicense
```

With this URI, use the following attribute in a bundleObjectLicense object:

Field	Type	Required	Description
bundleObjectId	String	Yes	The ID of the bundle.

POST Response

Returns the success response if the request is successful. Returns the error object if errors occur.

DELETE Request

To uninstall a bundle from the organization, use the following URI:

```
/api/v2/bundleObjectLicense?bundleObjectId=<bundleId>&updateOption=<updateOption>
```

Use the following bundleObjectLicense Delete URI attributes:

Field	Type	Required	Description
bundleObjectId	String	Yes	The ID of the bundle.
updateOption	String		Defines what happens if objects in the bundle are used. Use one of the following options: - DELETE_EXISTING_OBJECTS. Deletes the objects that use the bundle object. - UPDATE_EXISTING_OBJECTS. Updates the object that uses the bundle object. - EXCEPTION_IF_IS_USED. Returns a message when a bundle object is used and cancels the uninstallation.

DELETE Response

Returns the success response if the request is successful. Returns the error object if errors occur.

job

Use this resource to start or stop a task based on ID or name. You can also retrieve job completion status.

Start POST Request

If your organization uses projects and folders, use the REST API version 3 lookup resource to retrieve the task ID. This ID is the federated task ID, which you must include in the POST request.

Alternatively, you can use the task resource to retrieve the task ID. However, the task resource returns a task ID that you can only use to run tasks located in the Default folder.

Do not use this resource for a Mass Ingestion file ingestion task. Instead, use the file ingestion job resource. For more information, see .

To start a task, use the following URI:

```
/api/v2/job
```

With this URI, use the following attributes in a job object:

Field	Type	Required	Description
taskId	String	Required if taskName or taskFederatedId is not included.	Task or linear taskflow ID. Use taskId or taskName in the URI. You can include this task ID when the task is located in the Default folder.
taskFederatedId	String	Required if the task is not located in the Default folder.	Global unique identifier for the task, which includes the folder path to the task.
taskName	String	Required if taskId or taskFederatedId is not included.	Task or linear taskflow name. Use taskId or taskName in the URI.
taskType	String	Yes	The type of task. For Data Integration, use one of the following codes: - DMASK. Masking task. - DRS. Replication task. - DSS. Synchronization task. - MTT. Mapping task. - PCS. PowerCenter task. - WORKFLOW. Linear taskflow.
callbackURL	String	-	A valid, publicly available URL. The service posts the job status to the callbackURL.

Start POST Response

Returns the job object if the request is successful. Returns an error object if errors occur.

The job object includes the following attributes:

Field	Type	Description
taskId	String	Task or linear taskflow ID.
taskFederatedId	String	Global unique identifier for the task, which includes the folder path to the task.
taskName	String	Task or linear taskflow name.
taskType	String	The type of task. Returns one of the following codes for Data Integration: - DMASK. Masking task. - DRS. Replication task. - DSS. Synchronization task. - MTT. Mapping task. - PCS. PowerCenter task. - WORKFLOW. Linear taskflow.

Field	Type	Description
runId	Long	ID of the job.
callbackURL	String	Status of the job.

Start POST Request Example

To start a linear taskflow with an ID of 0034J90000000M in JSON, you might use the following request:

```
POST <serverUrl>/api/v2/job HTTP/1.0
Content-Type: application/json
Accept: application/json
icSessionId: <icSessionId>

{
"taskId": "0034J90000000M",
"taskType": "Workflow",
"callbackURL": "https://MyIICSJobStatus.com",
}
```

To start a mapping task with the ID of 0100000Z000009, you might use the following request. To run multiple instances of the task simultaneously, the request includes the runtime object.

```
POST <serverUrl>/api/v2/job HTTP/1.0
Content-Type: application/json
Accept: application/json
icSessionId: <icSessionId>

{
    "@type": "job",
    "taskId": "0100000Z000009",
    "taskType": "MTT",
    "runtime": {
        "@type": "mtTaskRuntime"
    }
}
```

Stop POST Request

To stop a task or linear taskflow, use the following URI:

```
/api/v2/job/stop
```

With this URI, use the following attributes in a job object:

Field	Type	Required	Description
taskId	String	Required if taskName not included.	Task or linear taskflow ID. Use taskId or taskName in the URI.
taskFederatedId	String	Required if the task is not located in the Default folder.	Global unique identifier for the task, which includes the folder path to the task.

Field	Type	Required	Description
taskName	String	Required if taskId not included.	Task or linear taskflow name. Use taskId or taskName in the URI.
taskType	String	Yes	The type of task. For Data Integration, use one of the following codes: - DMASK. Masking task. - DRS. Replication task. - DSS. Synchronization task. - MTT. Mapping task. - PCS. PowerCenter task. - WORKFLOW. Linear taskflow.

Stop POST Response

Returns the success object if the request is successful. Returns the error object if errors occur.

Stop POST Example

To stop a linear taskflow with an ID of 0034J90000000M in JSON, you might use the following request:

```
POST <serverUrl>/api/v2/job/stop HTTP/1.0
Content-Type: application/json
Accept: application/json
icSessionId: <icSessionId>

{
"@type": "job",
"taskId": "0034J90000000M",
"taskType": "Workflow"
}
```

Clean Stop POST Request

To cleanly stop a task or linear taskflow, use the following URI:

```
/api/v2/job/stop?cleanStop=true
```

With this URI, use the following attributes in a job object:

Field	Type	Required	Description
taskId	String	Required if taskName not included.	Task or linear taskflow ID. Use taskId or taskName in the URI.
taskFederatedId	String	Required if the task is not located in the Default folder.	Global unique identifier for the task, which includes the folder path to the task.

Field	Type	Required	Description
taskName	String	Required if taskId not included.	Task or linear taskflow name. Use taskId or taskName in the URI.
taskType	String	Yes	The type of task. For Data Integration, use one of the following codes: - DMASK. Masking task. - DRS. Replication task. - DSS. Synchronization task. - MTT. Mapping task. - PCS. PowerCenter task. - WORKFLOW. Linear taskflow.

Clean Stop POST Response

Returns the success object if the request is successful. Returns the error object if errors occur.

Clean Stop POST Example

To cleanly stop a mapping task named MappingTask1, you might use the following request:

```
POST <serverUrl>/api/v2/job/stop?cleanStop=1

{
"@type": "job",
    "taskName": "MappingTask1",
    "taskType": "MTT"
}
```

Job Status

When you include the callbackURL in the job request, the service sends a request to the callback URL when the job completes. The service always uses a JSON request for callbacks.

A callback might be called multiple times. For example, multiple callbacks might occur in the following situations:

- Your callback server returns an HTTP status code other than 200.

- Your callback server doesn't respond within 30 seconds.

- Your callback server is down.

- There is a transient network failure.

In any of these situations, the URL connection breaks and the service counts the break as a failed attempt. The service will make three immediate attempts to receive a successful response. Afterward, the attempts will occur in exponential increments. For example, the attempts might begin with a 30-second interval and progress up to a maximum 3-minute interval, until the total time reaches 30 minutes.

The service executes the POST request from the callback URL. The following text is a sample return:

```
   {
     @type:"callbackUrlResponse"
     endTime: "2013-02-27T18:57:52.000Z",
     objectId: "0034J90000000M",
     objectName: "taskName",
     runId: 5,
     status: "COMPLETED" // or "FAILED"
}
```

login

You can use this resource to log in to your organization using your Informatica Intelligent Cloud Services user account.

The login response includes the base URL and session ID values that you include in requests that you send during the REST API session.

Use the logout resource to end the session.

To log in using SAML single sign-on, see .

To log in using Salesforce credentials, see .

Note: Your team might use multiple organizations such as an organization for development and an organization for testing. The user credentials that you use to log in determine the organization that you access.

POST Request

To log in, use the following URL:

```
https://<cloud provider>-<region>.informaticacloud.com/ma/api/v2/user/login
```

The values for cloud provider and region correspond to the name of the POD (Point of Deployment) that your organization uses. The following table lists the POD names and the corresponding cloud provider and region to use in the login URL

POD name	Cloud provider-region
USW1	dm-us
USE2	dm-us
USW3	dm-us
USE4	dm-us
USW5	dm-us
USE6	dm-us
USW1-1	dm1-us
USW3-1	dm1-us
USW1-2	dm2-us
CAC1	dm-na
APSE1	dm-ap
APSE2	dm1-apse
APNE1	dm1-ap
APAUC1	dm1-apau

POD name	Cloud provider-region
EMW1	dm-em
EMC1	dm1-em
UK1	dm-uk

For example, if your organization uses the APNE1 POD, use the following URL:

```
https://dm1-ap.informaticacloud.com/ma/api/v2/user/login
```

If you don't know the name of the POD that your organization uses, contact your organization administrator or Informatica Global Customer Support.

For more information about the POD names and corresponding cloud providers and regions, see the Product Availability Matrix (PAM) for Informatica Intelligent Cloud Services on the Knowledge Base.

Use the following attributes in a login object:

Field	Type	Required	Description
username	String	Yes	Informatica Intelligent Cloud Services user name for the organization that you want to log in to. Maximum length is 255 characters.
password	String	Yes	Informatica Intelligent Cloud Services password. Maximum length is 255 characters.

POST Response

Returns the user object if the request is successful. Returns the error object if errors occur.

The response includes the following information that you need to include in the header of subsequent REST API calls:

- icSessionId. A REST API session ID that you include in the header for version 2 REST API calls. The session ID expires after 30 minutes of inactivity. If the session ID expires, log in again to continue working with the REST API.
  For information on retrieving session status details, see "Session IDs" on page 21.

- serverUrl. The base URL that you use in all version 2 resource URIs except for login, for example:
  `<serverUrl>/api/v2/job`

The user object includes the following attributes:

Field	Type	Description
id	String	User ID.
orgId	String	ID of the organization the user belongs to. 22 characters. **Note:** Organizations that were created in legacy Informatica Cloud might have an organization ID of 6 characters.

Field	Type	Description
orgUuid	String	Unique identifier for the organization.
name	String	Informatica Intelligent Cloud Services user name.
description	String	Description of the user.
createTime	String	When the user account was created.
updateTime	String	When the user account was last updated
createdBy	String	Informatica Intelligent Cloud Services user who created the user account.
updatedBy	String	Informatica Intelligent Cloud Services user who last updated the user account.
sfUsername	String	Salesforce user name. Included when user is configured to authenticate through Salesforce.
password	String	Salesforce user password. Included when user is configured to authenticate through Salesforce.
firstName	String	First name for the user account.
lastName	String	Last name for the user account.
title	String	Title of the user.
phone	String	Phone number for the user.
securityQuestion	String	Security question. Returns one of the following codes: - SPOUSE_MEETING_CITY - FIRST_JOB_CITY - CHILDHOOD_FRIEND - MOTHER_MAIDEN_NAME - PET_NAME - CHILDHOOD_NICKNAME - CUSTOM_QUESTION:"<question>"
securityAnswer	String	Answer to the security question.
roles		Object that includes a role object for each role assigned to the user.
name	String	Included in role object. Role name. Returns one of the following codes: - Service Consumer - Designer - Admin
description	String	Included in role object. Role description.
emails	String	Email address to be notified when the user changes the account password.
timezone	String	Time zone of the user. Time zone honors Daylight Saving Time. For more information, see "Time zone codes" on page 543.

Field	Type	Description
serverUrl	String	Informatica Intelligent Cloud Services URL for the organization the user belongs to. Use the serverUrl as a base for most version 2 REST API resource URIs.
spiUrl	String	This field is no longer applicable and has been deprecated.
uuId	String	Unique identifier for the user.
icSessionId	String	Informatica Intelligent Cloud Services session ID for version 2 REST API session. Use in most version 2 REST API request headers.
forceChangePassword	Boolean	Determines if the user must reset the password after the user logs in for the first time. Includes the following values: - True. The user must reset the password. - False. The user is not forced to reset the password.

POST Example

To log in to your Informatica Intelligent Cloud Services organization, you might use the following request:

```
POST https://dm-us.informaticacloud.com/ma/api/v2/user/login
Content-Type: application/json
Accept: application/json

{
"@type": "login",
"username": "John@infa.com",
"password": "mypassword"
}
```

The response returns the user object which contains the serverUrl and icSessionId values to use in subsequent calls, as shown in the following example:

```
{
    "id": "0101TQ03000000000007",
    "orgId": "0101TQ",
    "orgUuid": "3FNFLs1uHe2IIgTs8tRjSJ",
    "name": "John@infa.com",
    "description": "",
    "createTime": "2018-02-16T00:20:07.000Z",
    "updateTime": "2018-07-17T22:45:50.000Z",
    "createdBy": "System built-in user",
    "updatedBy": "John@infa.com",
    "sfUsername": null,
    "firstName": "John",
    "lastName": "Randall",
    "title": "IICS Admin",
    "password": "**********",
    "phone": "123-456-7899",
    "emails": "John@infa.com",
    "timezone": null,
    "serverUrl": "https://usw3.dm-us.informaticacloud.com/saas",
    "icSessionId": "1Ckv5VDHe2IICHi2hq04EF",
    "securityQuestion": "In what city were you born?",
    "securityAnswer": "********",
    "uuid": "a51jk7TB0IDcnWLwJdLaW2",
    "forceChangePassword": false,
    "roles": [
        {
            "name": "Admin",
            "description": "Role for performing administrative tasks for an
organization. Has full access to all licensed services."
        },
        {
```

```
            "name": "Data Preview",
            "description": "Role to preview data"
        },
        {
            "name": "Designer",
            "description": "Role for creating assets, tasks, and processes. Can
configure connections, schedules, and runtime environments. Has access to the
Application Integration Console."
        }
    ],
}
```

Using the session ID and server URL values in the above response as an example, to send a GET request to obtain Secure Agent information, you might use the following request:

```
GET https://usw3.dm-us.informaticacloud.com/saas/api/v2/agent
Content-Type: application/json
Accept: application/json
icSessionId: 1Ckv5VDHe2IICHi2hq04EF
```

loginOAuth

Use this resource to log in to Informatica Intelligent Cloud Services using a JSON Web Token (JWT) access token.

The JWT access token is provided by your identity provider (IDP). Retrieve the token from your identity provider and include it in the loginOAuth request. The token can be used for one login request.

Before you can use the loginOAuth resource, the following prerequisites must be met:

- The organization is licensed and configured to use SAML.

- Users are set up as SAML users and are activated in the organization.

- The organization's identity provider is registered.

- The JWT access token's audience claim maps to the SAML entity ID configured in your organization's SAML setup. The audience claim can be one of the following values:

  - api://<orgId>.<cloud provider>-<region>.informaticacloud.com

  - api://<cloud provider>-<region>.informaticacloud.com.<orgId>

  - api://<cloud provider>-<region>.informaticacloud.com/<orgId>

  - https://<orgId>.<cloud provider>-<region>.informaticacloud.com

  - https://<cloud provider>-<region>.informaticacloud.com.<orgId>

  - https://<cloud provider>-<region>.informaticacloud.com/<orgId>

Note: Optionally, you can include the resource in the audience claim. The resource name is customizable and can be set to any value, for example, LoginResource1, or UserLoginResource.

For more information about SAML configuration, see "User Administration" in *Administrator*.

For more information about registering identity providers, see .

Use values from the following fields returned in the response:

- icSessionId. A 30-minute REST API session ID that you include in the header for REST API calls. After the session ID expires, log in again to continue working with the REST API.
  For information on retrieving session status details, see .

- serverUrl. The base URL that you use in all resource URIs.

Use the logout resource to end the session.

POST Request

The login request must include a JWT access token. To get a JWT access token, see the documentation provided by your identity provider.

To log in, use the following URL:

```
https://<cloud provider>-<region>.informaticacloud.com/ma/api/v2/user/loginOAuth
```

For a list of cloud provider and region values, see .

Include the following attributes in the request:

Field	Type	Required	Description
oauthToken	String	Yes	JWT access token.
orgId	String	Yes	ID of the organization the user belongs to.

POST Response

Returns the user object if the request is successful. Returns the error object if errors occur.

Use the base URL and session ID returned in the response for subsequent requests during this session.

The user object includes the following attributes:

Field	Type	Description
id	String	User ID.
orgId	String	ID of the organization the user belongs to.
orgUuid	String	Unique identifier for the organization.
name	String	Informatica Intelligent Cloud Services user name.
description	String	Description of the user.
createTime	String	When the user account was created.
updateTime	String	When the user account was last updated
createdBy	String	Informatica Intelligent Cloud Services user who created the user account.
updatedBy	String	Informatica Intelligent Cloud Services user who last updated the user account.
sfUsername	String	Salesforce user name. Included when user is configured to authenticate through Salesforce.
password	String	Salesforce user password. Included when user is configured to authenticate through Salesforce.
firstName	String	First name for the user account.

Field	Type	Description
lastName	String	Last name for the user account.
title	String	Title of the user.
password	String	User password, obfuscated.
phone	String	Phone number for the user.
emails	String	Email address to be notified when the user changes the account password.
timezone	String	Time zone of the user. Time zone honors Daylight Saving Time. For more information, see "Time zone codes" on page 543.
serverUrl	String	Informatica Intelligent Cloud Services URL for the organization that the user belongs to. Use as a base for most version 2 and version 3 REST API resource URIs.
icSessionId	String	Informatica Intelligent Cloud Services Session ID. Use in most version 2 and version 3 REST API request headers.
securityQuestion	String	Security question. Returns one of the following codes: - SPOUSE_MEETING_CITY - FIRST_JOB_CITY - CHILDHOOD_FRIEND - MOTHER_MAIDEN_NAME - PET_NAME - CHILDHOOD_NICKNAME - CUSTOM_QUESTION:"<question>"
securityAnswer	String	Answer to the security question, obfuscated.
uuId	String	Unique identifier for the user.
forceChangePassword	Boolean	Determines if the user must reset the password after the user logs in for the first time. Includes the following values: - True. The user must reset the password. - False. The user is not forced to reset the password.
roles		Object that contains roles assigned to the user.
name	String	Included in role object. Role name. Returns one of the following codes: - Service Consumer - Designer - Admin
description	String	Included in role object. Role description.
usergroups		Object that contains the usergroups assigned to the user.
id	String	Included in the usergroups object. User group ID.

Field	Type	Description
orgId	String	Included in the usergroups object. ID of the organization the user group belongs to.
name	String	Included in the usergroups object. Name of the user group.
description	String	Included in the usergroups object. Description of the user group.
createTime	String	Included in the usergroups object. Date and time the user group was created.
updateTime	String	Included in the usergroups object. Date and time the user group was last updated.
createdBy	String	Included in the usergroups object. User who created the user account.
updatedBy	String	Included in the usergroups object. User who last updated the user account.
spiUrl	String	This field is no longer applicable and has been deprecated.

POST Example

To log in to Informatica Intelligent Cloud Services using a JWT access token, you might use the following request:

```
POST https://dm-us.informaticacloud.com/ma/api/v2/user/loginOAuth
Content-Type: application/json
Accept: application/json

{
"orgId": "6xVpQpzHBAoizhbMOLzty9",
"oauthToken": "<JWT token>"
}
```

The response returns the user object which contains the serverUrl and icSessionId values to use in subsequent calls, as shown in the following example:

```
{
    "id": "01000103000000000002",
    "orgId": "010001",
    "orgUuid": "6xVpQpzHBAoizhbMOLzty9",
    "name": "Larry@infa.com",
    "description": null,
    "createTime": "2023-07-20T15:13:12.000Z",
    "updateTime": "2023-07-20T15:13:32.000Z",
    "createdBy": "ma",
    "updatedBy": "Scott@infa.com",
    "sfUsername": null,
    "firstName": "Larry",
    "lastName": "Felyne",
    "title": "Manager",
    "password": "********",
    "phone": "423435546657652",
    "emails": "Larry@infa.com",
    "timezone": null,
    "serverUrl": "https://na4.dm-us.informaticacloud.com/saas",
```

```
        "icSessionId": "0UNdbRXUXHpfqKZEbfmxoQ",
        "securityQuestion": "In what city did you meet your spouse/significant other?",
        "securityAnswer": "********",
        "uuid": "6qnnXdzBdtUbObUTYhyWO1",
        "forceChangePassword": false,
        "roles": [
            {
                "name": "Admin",
                "description": "Role for performing administrative tasks for an
organization. Has full access to all licensed services."
            },
            {
                "name": "Data Preview",
                "description": "Role to preview data"
            },
            {
                "name": "Designer",
                "description": "Role for creating assets, tasks, and processes. Can
configure connections, schedules, and runtime environments. Has access to the
Application Integration Console."
            }
        ],
        "usergroups": [
            {
                "id": "aRfrqNzCfg7e9SDOJ3y1Yn",
                "orgId": "010001",
                "name": "G2",
                "description": "",
                "createTime": "2023-08-01T05:33:22.000Z",
                "updateTime": "2023-08-01T05:33:23.000Z",
                "createdBy": "Scott@infa.com",
                "updatedBy": "Scott@infa.com"
            }
        ],
        "spiUrl": null
}
```

As an example of using the serverUrl and icSessionId values in subsequent requests, to send a GET request to obtain Secure Agent information, you might use the following request:

```
GET https://na4.dm-us.informaticacloud.com/saas/api/v2/agent
Content-Type: application/json
Accept: application/json
icSessionId: 0UNdbRXUXHpfqKZEbfmxoQ
```

loginSaml

Use this version 2 API resource to log in to Informatica Intelligent Cloud Services using a Security Assertion Markup Language (SAML) token. The SAML token is a SAML response generated by your identity provider that contains a SAML assertion.

The loginSaml response includes the session ID and base URL that you include in subsequent REST API calls. Use values from the following fields returned in the response:

- icSessionId. A 30-minute REST API session ID that you include in the header for version 2 REST API calls. After the session ID expires, log in again to continue working with the REST API.
  For information on retrieving session status details, see "Session IDs" on page 21.

- serverUrl. The base URL that you use in all version 2 resource URIs except for loginSaml, for example: `<serverUrl>/api/v2/job`

Use the logout resource to end the session.

POST Request

The login request must include a SAML token. To get a SAML token, see the documentation provided by your identity provider. To see an example of a SAML token or SAML response, see https://www.samltool.com/generic_sso_res.php.

To log in, use the following URL:

```
https://<cloud provider>-<region>.informaticacloud.com/ma/api/v2/user/loginSaml
```

For a list of cloud provider and region values, see .

With this URL, use the following attributes in a login object:

Field	Type	Required	Description
samlToken	String	Yes	SAML token.
orgId	String	Yes	Informatica Intelligent Cloud Services organization ID.

POST Response

Returns the user object if the request is successful. Returns the error object if errors occur.

Use the base URL and session ID returned in the response for subsequent requests during this session.

The user object includes the following attributes:

Field	Type	Description
id	String	User ID.
orgId	String	ID of the organization the user belongs to.
orgUuid	String	Unique identifier for the organization.
name	String	Informatica Intelligent Cloud Services user name.
description	String	Description of the user.
createTime	String	When the user account was created.
updateTime	String	When the user account was last updated
createdBy	String	Informatica Intelligent Cloud Services user who created the user account.
updatedBy	String	Informatica Intelligent Cloud Services user who last updated the user account.
firstName	String	First name for the user account.
lastName	String	Last name for the user account.
title	String	Title of the user.
phone	String	Phone number for the user.
roles		Object that contains roles assigned to the user.

Field	Type	Description
name	String	Included in role object. Role name. Returns one of the following codes: - Service Consumer - Designer - Admin
description	String	Included in role object. Role description.
email	String	Email address to be notified when the user changes the account password.
timezone	String	Time zone of the user. Time zone honors Daylight Saving Time. For more information, see "Time zone codes" on page 543.
serverUrl	String	Informatica Intelligent Cloud Services URL for the organization the user belongs to. Use the serverUrl as a base for most version 2 REST API resource URIs.
icSessionId	String	Informatica Intelligent Cloud Services session ID for version 2 REST API session. Use in most version 2 REST API request headers.
spiUrl	String	This field is no longer applicable and has been deprecated.
uuId	String	Unique identifier for the user.

POST Example

To log in to Informatica Intelligent Cloud Services using SAML single sign-on, you might use the following request:

```
POST https://dm-us.informaticacloud.com/ma/api/v2/user/loginSaml
Content-Type: application/json
Accept: application/json

{
"@type": "login",
"samlToken": "<SAML token>",
"orgId": "003420"
}
```

The response returns the user object which contains the serverUrl and icSessionId values to use in subsequent calls, as shown in the following example:

```
{
    "id": "0101TQ03000000000007",
    "orgId": "003420",
    "orgUuid": "3FNFLs1uHe2IIgTs8tRjSJ",
    "name": "John@infa.com",
    "description": "",
    "createTime": "2018-02-16T00:20:07.000Z",
    "updateTime": "2018-07-17T22:45:50.000Z",
    "createdBy": "System built-in user",
    "updatedBy": "John@infa.com",
    "sfUsername": null,
    "firstName": "John",
    "lastName": "Randall",
    "title": "IICS Admin",
    "phone": "123-456-7899",
    "emails": "John@infa.com",
    "timezone": null,
    "serverUrl": "https://na4.dm-us.informaticacloud.com/saas",
```

```
    "icSessionId": "1Ckv5VDHe2IICHi2hq04EF",
    "securityQuestion": "In what city were you born?",
    "securityAnswer": "********",
    "uuid": "a51jk7TB0IDcnWLwJdLaW2",
    "forceChangePassword": false,
    "roles": [
        {
            "name": "Admin",
            "description": "Role for performing administrative tasks for an
organization. Has full access to all licensed services."
        },
        {
            "name": "Data Preview",
            "description": "Role to preview data"
        },
        {
            "name": "Designer",
            "description": "Role for creating assets, tasks, and processes. Can
configure connections, schedules, and runtime environments. Has access to the
Application Integration Console."
        }
    ],
}
```

Using the above response as an example, to send a GET request to obtain Secure Agent information, you might use the following request:

```
GET https://na4.dm-us.informaticacloud.com/saas/api/v2/agent
Content-Type: application/json
Accept: application/json
icSessionId: 1Ckv5VDHe2IICHi2hq04EF
```

loginSf

Use this resource to log in to an Informatica Intelligent Cloud Services organization using Salesforce credentials.

The login response includes the session ID and base URL that you need to include in subsequent REST API calls.

Note: You must activate your Informatica Intelligent Cloud Services user account before you can log in using the loginSf resource.

Use the Salesforce Web Services API to generate a Salesforce session ID and to retrieve the Salesforce server URL. For more information, see the Salesforce *Web Services API Developer's Guide*.

Use the logout resource to end the session.

POST Request

To log in using Salesforce credentials, use the following URL:

```
https://<cloud provider>-<region>.informaticacloud.com/ma/api/v2/user/loginSf
```

For a list of cloud provider and region values, see <u>"login" on page 54</u>.

Use the following attributes in a loginSf object:

Field	Type	Required	Description
sfSessionId	String	Yes	Salesforce session ID. For information about generating the Salesforce session ID, see the login resource in the Salesforce *Web Services API Developer's Guide*.
sfServerUrl	String	Yes	Salesforce server URL. Retrieve the Salesforce server URL from the Salesforce API login resource response.

POST Response

Returns the user object if the request is successful. Returns the error object if errors occur.

The response includes the following information that you need to include in the header of subsequent REST API calls:

- icSessionId. A REST API session ID that you include in the header for version 2 REST API calls. The session ID expires after 30 minutes of inactivity. After the session ID expires, log in again to continue working with the REST API.
  For information on retrieving session status details, see "Session IDs" on page 21.

- serverUrl. The base URL that you use in all version 2 resource URIs except for login, for example:
  `<serverUrl>/api/v2/job`

The user object includes the following attributes:

Field	Type	Description
id	String	User ID.
orgId	String	ID of the organization the user belongs to. 22 characters. **Note:** Organizations that were created in legacy Informatica Cloud might have an organization ID of 6 characters.
orgUuid	String	Unique identifier for the organization.
name	String	Informatica Intelligent Cloud Services user name.
description	String	Description of the user.
createTime	String	When the user account was created.
updateTime	String	When the user account was last updated
createdBy	String	Informatica Intelligent Cloud Services user who created the user account.
updatedBy	String	Informatica Intelligent Cloud Services user who last updated the user account.
sfUsername	String	Salesforce user name. Included when user is configured to authenticate through Salesforce.
password	String	Salesforce user password. Included when user is configured to authenticate through Salesforce.

Field	Type	Description
firstName	String	First name for the user account.
lastName	String	Last name for the user account.
title	String	Title of the user.
phone	String	Phone number for the user.
securityQuestion	String	Security question. Returns one of the following codes: - SPOUSE_MEETING_CITY - FIRST_JOB_CITY - CHILDHOOD_FRIEND - MOTHER_MAIDEN_NAME - PET_NAME - CHILDHOOD_NICKNAME - CUSTOM_QUESTION:"<question>"
securityAnswer	String	Answer to the security question.
roles		Object that includes a role object for each role assigned to the user.
name	String	Included in role object. Role name. Returns one of the following codes: - Service Consumer - Designer - Admin
description	String	Included in role object. Role description.
emails	String	Email address to be notified when the user changes the account password.
timezone	String	Time zone of the user. Time zone honors Daylight Saving Time. For more information, see "Time zone codes" on page 543.
serverUrl	String	Informatica Intelligent Cloud Services URL for the organization the user belongs to. Use the serverUrl as a base for most version 2 REST API resource URIs.
spiUrl	String	This field is no longer applicable and has been deprecated.
uuId	String	Unique identifier for the user.
icSessionId	String	Informatica Intelligent Cloud Services session ID for version 2 REST API session. Use in most version 2 REST API request headers.
forceChangePassword	Boolean	Determines if the user must reset the password after the user logs in for the first time. Includes the following values: - True. The user must reset the password. - False. The user is not forced to reset the password.

POST Example

To log in to your Informatica Intelligent Cloud Services organization, you might use the following request:

```
POST https://dm-us.informaticacloud.com/ma/api/v2/user/loginSf
Content-Type: application/json
```

```
Accept: application/json

{
"@type": "loginSf",
"sfSessionId": "00Df40000000coF!ARYAQDO2SvoD3eRXOrNaiOb9a3Pp",
"sfServerUrl": "https://c.na41.visual.force.com/services/Soap/u/27.0/00Df40000000coF"
}
```

The response returns the user object which contains the serverUrl and icSessionId values to use in subsequent calls, as shown in the following example:

```
{
    "id": "0101TQ03000000000007",
    "orgId": "0101TQ",
    "orgUuid": "3FNFLs1uHe2IIgTs8tRjSJ",
    "name": "John@infa.com",
    "description": "",
    "createTime": "2018-02-16T00:20:07.000Z",
    "updateTime": "2018-07-17T22:45:50.000Z",
    "createdBy": "System built-in user",
    "updatedBy": "John@infa.com",
    "sfUsername": "JohnR",
    "firstName": "John",
    "lastName": "Randall",
    "title": "IICS Admin",
    "password": "**********",
    "phone": "123-456-7899",
    "emails": "John@infa.com",
    "timezone": null,
    "serverUrl": "https://na4.dm-us.informaticacloud.com/saas",
    "icSessionId": "1Ckv5VDHe2IICHi2hq04EF",
    "securityQuestion": "In what city were you born?",
    "securityAnswer": "********",
    "uuid": "a51jk7TB0IDcnWLwJdLaW2",
    "forceChangePassword": false,
    "roles": [
        {
            "name": "Admin",
            "description": "Role for performing administrative tasks for an
organization. Has full access to all licensed services."
        },
        {
            "name": "Data Preview",
            "description": "Role to preview data"
        },
        {
            "name": "Designer",
            "description": "Role for creating assets, tasks, and processes. Can
configure connections, schedules, and runtime environments. Has access to the
Application Integration Console."
        }
    ],
}
```

Using the above response as an example, to send a GET request to obtain Secure Agent information, you might use the following request:

```
GET https://na4.dm-us.informaticacloud.com/saas/api/v2/agent
Content-Type: application/json
Accept: application/json
icSessionId: 1Ckv5VDHe2IICHi2hq04EF
```

logout

Use this resource to log out of an organization and end the version 2 REST API session specified in the request.

Don't log out of a REST API session until all the requests that you've sent have been executed.

POST Request

To log out an organization and end the version 2 REST API session, include the session ID in the request header with the following URI:

```
/api/v2/user/logout
```

POST Response

Returns the 200 response code if the request is successful.

Returns the error object if errors occur.

POST Example

To log out of your organization, you might use the following request:

```
POST <serverURL>/api/v2/user/logout
Content-Type: application/json
Accept: application/json
icSessionId: <icSessionId>
```

logoutall

Use this resource to log out of an organization and end all version 2 REST API sessions for the organization.

Don't log out of the sessions until all the API requests have been executed.

POST Request

To log out of an organization and end all version 2 REST API sessions for the organization, use the following URL:

```
https://<cloud provider>-<region>.informaticacloud.com/ma/api/v2/user/logoutall
```

With this URL, use the following attributes in a logout object:

username

Informatica Intelligent Cloud Services user name.

password

Informatica Intelligent Cloud Services password.

POST Response

Returns the success object if the request is successful.

Returns the error object if errors occur.

POST Example

To log out of an organization and all version 2 REST API sessions, you might use the following request:

```
POST https://dm-us.informaticacloud.com/ma/api/v2/user/logoutall
Content-Type: application/json
Accept: application/json
```

```
{
"@type": "logout",
"username": "useremail@company.com",
"password": "mypassword"
}
```

org

Use this resource to request the details of your Informatica Intelligent Cloud Services organization or a related sub-organization. You can use this resource to update an organization or related sub-organization. You can also delete a sub-organization.

GET Request

To request the details of your organization, use the following URI:

```
/api/v2/org
```

To request the details of a sub-organization related to your organization, you can include the sub-organization ID or sub-organization name in the URI. Use one of the following URIs:

```
/api/v2/org/<sub-organization ID>
```

```
/api/v2/org/name/<sub-organization name>
```

If you use an organization name in the URI and the name includes a space, replace the space with `%20`. For example:

```
/api/v2/org/name/my%20suborg
```

GET Response

When you request the details of an organization, Informatica Intelligent Cloud Services returns the org object in list format.

If the organization is a parent organization in an organization hierarchy, the org object includes the IDs and names of all sub-organizations.

Returns the error object if errors occurred.

The **org** object includes the following attributes:

Field	Type	Description
id	String	Organization ID.
orgId	String	Organization ID.
name	String	Organization name.
description	String	Description of the organization.
createTime	Date/time	Time the organization was created.
updateTime	Date/time	Last time the organization was updated.
createdBy	String	ser who created the organization.

Field	Type	Description
updatedBy	String	Last user who updated the organization.
parentOrgId	String	Organization ID for the parent organization. Returns 0 if the organization is a stand-alone or parent organization.
address1	String	Address for the organization.
address2	String	Additional address information for the organization.
address3	String	Additional address information for the organization.
city	String	City where the organization is based.
state	String	State where the organization is based. Returns a state code. For more information, see "State codes" on page 535.
zipcode	String	Postal code of the area where the organization is based.
timezone	String	Time zone of the organization. For more information, see "Time zone codes" on page 543.
country	String	Country where the organization is based. Returns a country code. For more information, see "Country codes" on page 536.
employees	String	Range of employees in the organization.
offerCode	String	Offer code assigned to Informatica Intelligent Cloud Services partners.
successEmails	String	Email address to receive notification of tasks that complete successfully.
warningEmails	String	Email address to receive notification of tasks that complete with errors.
errorEmails	String	Email address to receive notification of tasks that fail to complete.
campaignCode	String	Campaign code.
atlasProjectId	String	Atlas project ID.
zuoraAccountId	String	Zuora account ID.
spiUrl	String	This field is no longer applicable and has been deprecated.
devOrg	Boolean	Indicates the organization is a development organization. Returns 1 for a development organization. Returns 0 for a production organization.
maxLogRows	Int	Maximum number of rows to keep in the activity log.
minPasswordLength	Int	Minimum number of characters for a user account password.

Field	Type	Description
minPasswordCharMix	Int	Mix of characters each password must contain. Passwords can contain a mix of the following character sets: lowercase letters, capital letters, numbers, and special characters. Returns one of the following values: - 1. Contains at least one of the character sets. - 2. Contains at least two of the character sets. - 3. Contains at least three of the character sets. - 4. Contains all four character sets.
passwordReuseInDays	Int	Number of days until a previous password can be used again. A value of 0 means a password can always be reused.
passwordExpirationInDays	Int	Number of days until a password expires. A value of 0 means a password will never expire.
subOrgLimit	Int	Number of sub-organizations allowed. If the limit has been customized, the REST API returns the custom limit. Otherwise, the REST API returns the limit associated with the edition.
restApiSessionLimit	Int	Number of concurrent REST API sessions allowed. If the limit has been customized, the REST API returns the custom limit. Otherwise, the REST API returns the limit associated with the edition.
parentOrgId	String	Organization ID of the parent organization. 0 indicates the organization is a stand-alone or parent organization.
jobExecUserProfile	String	Informatica Intelligent Cloud Services user account configured to run contact validation tasks.
orgUUID	String	Unique identifier for the organization.
subOrgs		Object that contains information for each sub-organization.
id	String	Included in subOrgs object. ID of the sub-organization.
name	String	Included in subOrgs object. Name of the sub-organization.

POST Request

You can update an Informatica Intelligent Cloud Services organization if the user that started the REST API session has the Admin role and belongs to either the organization that you want to update or the parent organization.

You can update a sub-organization if your organization has the appropriate license and if the user that started the REST API session has the Admin role in the parent organization.

To update the details of a sub-organization related to your parent organization, use the organization ID in the following URI. To update the details of your organization, omit the optional ID.

```
/api/v2/org/<id>
```

Note: When you update an organization through the REST API, the action is a full update. If a field isn't included in the request, the value resets to the default.

You cannot update the organization ID, offer code, or organization administrator user account created with the organization.

With this URI, you can use the following attributes in the **org** object:

Field	Type	Required	Description
name	String	Yes	Organization name.
address	String	Yes	Address of organization.
address2	String		Additional address information for the organization.
address3	String		Additional address information for the organization.
city	String	Yes	City where the organization is based.
state	String	Required when Country is US	State where the organization is based. Use the appropriate state code. Required when Country is set to US. For more information, see Appendix A , "State codes" on page 535.
zipcode	String	Required when Country is US	Postal code of the area where the organization is based. Required when Country is set to US
country	String	Yes	Country where the organization is based. Use the appropriate country code. For more information, see Appendix A , "Country codes" on page 536.
timezone	String		Time zone of the organization. Time zone honors Daylight Saving Time. For more information, see Appendix A , "Time zone codes" on page 543.
description	String		Description of the organization. Maximum length is 255 characters.
successEmails	String		Default email address for notification of successful job completion.
warningEmails	String		Default email addresses for warnings about job completion.
errorEmails	String		Default email address for notification about job failure.
employees	String	Yes	Range of employees in the organization. Use one of the following ranges: - "0_10" - "11_25" - "26_50" - "51_100" - "101_500" - "501_1000" - "1001_5000" - "5001_"

Field	Type	Required	Description
offerCode	String		Offer code assigned to Informatica Intelligent Cloud Services partners.
passwordReuseInDays	Int		Number of days until a previous password can be used again. Maximum number of days is 730 (2 years). A value of 0 means a password can always be reused.
passwordExpirationInDays	Int		Number of days until the password expires. Maximum number of days is 180. A value of 0 means a password will never expire.

POST Response

If successful, returns the org request object for the organization that you created or updated.

Returns the error object if errors occur.

DELETE Request

You can delete an Informatica Intelligent Cloud Services sub-organization if the user that started the REST API session has the Admin role and belongs the parent organization.

To delete an Informatica Intelligent Cloud Services organization, use the organization ID with the following URI:

```
/api/v2/org/<id>
```

DELETE Response

Returns the 200 response code if the request is successful.

Returns the error object if errors occur.

POST Example

To update a sub-organization with an ID of 02340000, you might use the following request:

```
POST <serverUrl>/api/v2/org/02340000
Content-Type: application/xml
Accept: application/xml
icSessionId: <icSessionId>

<org>
  <name>Dev Org</name>
  <address1>333 Main Street</address1>
  <city>City</city>
  <state>MD</state>
  <zipcode>90001</zipcode>
  <country>US</country>
  <timezone>America/Chicago</timezone>
</org>
```

A successful request returns the org request object for the sub-organization that you updated.

register

Use this resource to create an Informatica Intelligent Cloud Services sub-organization. For Informatica Intelligent Cloud Services partners only.

You can create an Informatica Intelligent Cloud Services sub-organization if your organization has the appropriate license and if the user that started the REST API session has the Admin role in the parent organization.

register POST Request

To create an Informatica Intelligent Cloud Services sub-organization, use the following URI.

```
/api/v2/user/register
```

Use the session ID from the login response in the request header. Use the serverUrl from the login response as the base URL.

You can use the following attributes in a registration object:

Field	Type	Required	Description
org			Attribute that defines an Informatica Intelligent Cloud Services organization.
offerCode	String		Include in the org object. Offer code assigned to Informatica Intelligent Cloud Services partners.
name	String	Yes	Include in the org object. Name for the new Informatica Intelligent Cloud Services organization.
address1	String	Yes	Include in the org object. Address where the organization is located.
address2	String		Include in the org object. Additional address information for the organization.
address3	String		Include in the org object. Additional address information for the organization.
city	String	Yes	Include in the org object. City where the organization is located.
state	String	Yes	Include in the org object. State where the organization is located. Use the appropriate state code. Required for the United States. For more information, see Appendix A, "State codes" on page 535.
zipcode	String	Yes	Include in the org object. Postal code where the organization is located.

Field	Type	Required	Description
country	String	Yes	Include in the org object. Country where the organization is located. Use the appropriate country code. For more information, see Appendix A, "Country codes" on page 536.
timezone	String		Include in the org object. **Note:** Informatica Intelligent Cloud Services uses America/Los_Angeles as the default time zone. After you create the sub-organization, you can use the org resource to change the time zone or you can change the time zone in Administrator.
employees	String	Yes	Include in the org object. Number of employees in the organization. Use one of the following ranges: - "0_10" - "11_25" - "26_50" - "51_100" - "101_500" - "501_1000" - "1001_5000" - "5001_"
user			Attribute that defines the organization administrator user account.
name	String	Yes	Include in the user object. Email address for the organization administrator account.
password	String	Yes	Include in the user object. Password for the organization administrator account.
firstName	String	Yes	Include in the user object. First name of the organization administrator.
lastName	String	Yes	Include in the user object. Last name of the organization administrator.
title	String	Yes	Include in the user object. Title of the organization administrator.
phone	String	Yes	Include in the user object. Phone number for the organization administrator.
emails	String	Yes	Include in the user object. Email address that receives notification from Informatica Intelligent Cloud Services.

Field	Type	Required	Description
timezone	String		Include in the user object. Time zone of the organization administrator. **Note:** Informatica Intelligent Cloud Services uses America/Los_Angeles as the default time zone. After you create the sub-organization, you can use the org resource to change the time zone or you can change the time zone in Administrator.
securityQuestion	String		Include in the user object. Security question. Use one of the following codes to select the security question: - SPOUSE_MEETING_CITY - FIRST_JOB_CITY - CHILDHOOD_FRIEND - MOTHER_MAIDEN_NAME - PET_NAME - CHILDHOOD_NICKNAME - CUSTOM_QUESTION:"<question>"
securityAnswer	String		Include in the user object. Answer to the security question.
forceChangePassword	Boolean		Include in the user object. Determines if the user must reset the password after the user logs in for the first time. Includes the following values: - True. The user must reset the password. - False. The user is not forced to reset the password.
optOutOfEmails	Boolean		Include in the user object. Whether the user opts in or out of receiving marketing communication from Informatica. TRUE indicates that the user does not want to receive marketing communication.
registrationCode	String		Registration code.
sendEmail	Boolean		When registration completes, sends an email to the user email address with temporary login information. Use TRUE to send an email.

POST Response

Returns the user object if the request is successful. Returns the error object if errors occur.

The user object includes the following attributes.

Field	Type	Description
id	String	User ID.
orgId	String	ID of the organization the user belongs to. 22 characters. **Note:** Organizations that were created in legacy Informatica Cloud might have an organization ID of 6 characters.

Field	Type	Description
orgUuid	String	Unique identifier for the organization.
name	String	Informatica Intelligent Cloud Services user name.
description	String	Description of the user.
createTime	String	When the user account was created.
updateTime	String	When the user account was last updated
createdBy	String	Informatica Intelligent Cloud Services user who created the user account.
updatedBy	String	Informatica Intelligent Cloud Services user who last updated the user account.
sfUsername	String	Salesforce user name. Included when user is configured to authenticate through Salesforce.
password	String	Salesforce user password. Included when user is configured to authenticate through Salesforce.
firstName	String	First name for the user account.
lastName	String	Last name for the user account.
title	String	Title of the user.
phone	String	Phone number for the user.
securityQuestion	String	Security question. Returns one of the following codes: - SPOUSE_MEETING_CITY - FIRST_JOB_CITY - CHILDHOOD_FRIEND - MOTHER_MAIDEN_NAME - PET_NAME - CHILDHOOD_NICKNAME - CUSTOM_QUESTION:"<question>"
securityAnswer	String	Answer to the security question.
roles		Object that includes a role object for each role assigned to the user.
name	String	Included in role object. Role name. Returns one of the following codes: - Service Consumer - Designer - Admin
description	String	Included in role object. Role description.
emails	String	Email address to be notified when the user changes the account password.
timezone	String	Time zone of the user. Time zone honors Daylight Saving Time. For more information, see "Time zone codes" on page 543.

Field	Type	Description
serverUrl	String	Informatica Intelligent Cloud Services URL for the organization the user belongs to. Use the serverUrl as a base for most version 2 REST API resource URIs.
spiUrl	String	This field is no longer applicable and has been deprecated.
uuId	String	Unique identifier for the user.
icSessionId	String	Informatica Intelligent Cloud Services session ID for version 2 REST API session. Use in most version 2 REST API request headers.
forceChangePassword	Boolean	Determines if the user must reset the password after the user logs in for the first time. Includes the following values: - True. The user must reset the password. - False. The user is not forced to reset the password.

POST Examples

To register an organization in JSON, you might use the following request:

```
POST <serverUrl>/api/v2/user/register
Content-Type: application/json
Accept: application/json
{
    "@type" : "registration",
    "user" : {
        "@type" : "user",
        "name" : "useremail@company.com",
        "emails" : "useremail@company.com",
        "firstName" : "firstName",
        "lastName" : "lastName",
        "title" : "jobTitle",
        "phone" : "(0)1234 567 890",
        "timezone" : null,
        "forceChangePassword" : "true"
        "optOutOfEmails" : "true"
    },
    "org" : {
        "@type" : "org",
        "offerCode" : "PPC30daytrial",
        "campaignCode" : "PPC",
        "name" : "myOrg",
        "address1" : "1 Main St",
        "city" : "Mycity",
        "state" : "CA",
        "zipcode" : "90210",
        "country" : "US",
        "employees" : "5001_"
    },
    "registrationCode" : "ics-standard",
    "sendEmail" : true
}
```

A successful request returns the user object that was created, which includes the organization ID for the organization that was created.

Runtime environments

You can use the runtimeEnvironment resource to request runtime environment details and manage Secure Agent groups.

You can perform the following tasks using the REST API:

- Get details about runtime environments.
- Create, update, delete Secure Agent groups.
- Enable and disable Informatica Intelligent Cloud Services and connectors for Secure Agent groups.
- Configure Secure Agent service properties for Secure Agent groups.

Getting runtime environment details

Use the runtimeEnvironment resource to get runtime environment details.

You can request the details for a specific runtime environment or details for all of an organization's runtime environments,

To request runtime environment information for an organization, use the following URI:

```
/api/v2/runtimeEnvironment
```

To request the details of a particular runtime environment, you can include the runtime environment ID or name in the URI. Use one of the following URIs:

```
/api/v2/runtimeEnvironment/<id>
```

```
/api/v2/runtimeEnvironment/name/<name>
```

If you use the runtime environment name in the URI and the runtime environment name includes a space, replace the space with `%20`. For example:

```
/api/v2/runtimeEnvironment/name/my%20runtime%20environment
```

GET Response

Returns runtime environment information for the requested runtime environment. The runtimeEnvironment object includes the following attributes:

Field	Type	Description
id	String	Runtime environment ID.
orgId	String	Organization ID.
name	String	Runtime environment name.
description	String	Description of the runtime environment.
createTime	Date/time	Date and time the runtime environment was created.
updateTime	Date/time	Date and time that the runtime environment was last updated.
createdBy	String	User who created the runtime environment.
updatedBy	String	User who last updated the runtime environment.

Field	Type	Description
agents		Agents assigned to the runtime environment. See the agent resource.
isShared	Boolean	Indicates whether the Secure Agent group is shared. Returns one of the following values: - true. The Secure Agent group is shared. - false. The Secure Agent group is not shared.
federatedId	String	Global unique identifier.
serverlessConfig		Attribute that defines serverless runtime environment properties.

If you get information about a serverless runtime environment, the runtimeEnvironment object also includes the serverlessConfig object. The serverlessConfig object includes the following attributes:

Field	Type	Description
platform	String	Cloud platform that hosts the serverless runtime environment.
applicationType	String	Application that can use the serverless runtime environment.
status	String	Status of the serverless runtime environment.
statusMessage	String	Status message for the serverless runtime environment.
statusMessageDetails	String	Validation error data from AWS for the serverless runtime environment.
cloudProviderConfig		Defines basic properties and Data Integration properties.
infaAccountNumber	String	Included in the cloudProviderConfig object. Informatica's account number on the cloud platform.
externalId	String	Included in the cloudProviderConfig object. External ID to associate with the role.
configurationName	String	Included in the cloudProviderConfig object. Name of the serverless runtime environment.
configurationDescription	String	Included in the cloudProviderConfig object. Description of the serverless runtime environment.
params		Defines the AWS resource configuration properties.
s3	String	Included in the params object. Location on Amazon S3 to store supplementary files.
subnet	String	Included in the params object. ID of the subnet.
role	String	Included in the params object. Name of the role that the serverless runtime environment assumes.

Field	Type	Description
vpc	String	Included in the params object. ID of the VPC (Amazon Virtual Private Cloud).
externalId	String	Included in the params object. External ID that is configured for the role.
securityGroup	String	Included in the params object. Security group ID.
accountNumber	String	Included in the params object. Your account number on the cloud platform.
referenceId	String	Included in the params object. Internal ID used to reference the serverless runtime environment.
computeUnits	String	Included in the params object. Maximum number of serverless compute units that a task can use.
executionTimeout	String	Included in the params object. Amount of time in minutes to wait for a task to complete.
cloudInstanceId	String	Included in the params object. Instance ID.
zone	String	Included in the params object. Availability zone ID.
region	String	Included in the params object. Region on the cloud platform.
infaAccountNumber	String	Included in the params object. Informatica's account number on the cloud platform.
awsTags	String	Included in the params object. AWS tags.
maxComputeUnits	String	Maximum number of serverless compute units that a task can use.
executionTimeout	String	Amount of time in minutes to wait for a task to complete.

Get Example

To request the details of a particular runtime environment, you might use the following request:

```
GET <serverUrl>/api/v2/runtimeEnvironment/00000425000000000004
Accept:application/json
icSessionId: <icSessionId>
```

The following text is a sample return in JSON:

```
{
    "@type": "runtimeEnvironment",
    "id": "00000425000000000004",
    "orgId": "000004",
```

```
            "name": "SUT_Agent",
            "createTime": "2016-12-09T12:34:01.000Z",
            "updateTime": "2016-12-09T17:54:00.000Z",
            "createdBy": "org1@infa.com",
            "updatedBy": "org1@infa.com",
            "agents": [],
            "isShared": true,
            "federatedId": "6iPQuOsH1YAfnJxhZWPZjI"
    }
```

The following text is a sample return for a serverless runtime environment:

```
{
        "@type": "runtimeEnvironment",
        "id": "01000000000000000039",
        "orgId": "010211",
        "name": "Serverless runtime environment 1",
        "description": "My serverless runtime environment",
        "createTime": "2020-08-25T13:21:16.000Z",
        "updateTime": "2020-08-25T13:29:43.000Z",
        "createdBy": "admin",
        "updatedBy": "admin",
        "agents": [],
        "isShared": false,
        "federatedId": "4sddtYsgbpnpTBjSZB12fs",
        "serverlessConfig": {
            "platform": "AWS",
            "applicationType": "CDI",
            "status": "RUNNING",
            "statusMessage": "Serverless runtime is running",
            "cloudProviderConfig": {
                "cloudConfig": [
                    {
                        "infaAccountNumber": "064942996470",
                        "externalId": "7eeafa7c-6dd1-4666-8ac8-7431b1d72def",
                        "configurationName": "Serverless runtime environment 1",
                        "configurationDescription": "My serverless runtime environment",
                        "params": {
                            "s3": "s3://discale-qa-west2/test1",
                            "subnet": "subnet-08123adayy51ed327",
                            "role": "CDI_Serverless_Role",
                            "vpc": "vpc-02ef05yy73fb7f063",
                            "externalId": "7eeafa7c-6dd1-4666-8ac8-7431b1d72def",
                            "securityGroup": "sg-025d67343b0655372",
                            "accountNumber": "778525666549",
                            "referenceId": "4sddtYsgbpnpTBjSZB12fs",
                            "computeUnits": "1",
                            "executionTimeout": "2880",
                            "cloudInstanceId": "i-0e3e6g02r1g1364a3",
                            "zone": "usw2-az3",
                            "region": "us-west-2",
                            "infaAccountNumber": "064942996470",
                            "awsTags": "Key=NAME,Value=test1
Key=EMAIL,Value=test1@informatica.com"
                        }
                    }
                ]
            },
            "maxComputeUnits": "1",
            "executionTimeout": "2880"
        }
}
```

Creating, updating, and deleting Secure Agent groups

You can use the runtimeEnvironment resource to create, update, and delete Secure Agent groups.

Use a POST request to create and update Secure Agent groups. Use a DELETE request to delete a Secure Agent group.

POST request

To create a Secure Agent group, use the following URI:

```
/api/v2/runtimeEnvironment
```

Include the following fields in the request:

Field	Type	Required	Description
name	String	Yes	Name of the Secure Agent group.
isShared	Boolean	-	Whether the Secure Agent group can be shared with sub-organizations.

To update a Secure Agent group, include the agent group ID as shown in the following URI:

```
/api/v2/runtimeEnvironment/<id>
```

You can add or remove agents and rename Secure Agent groups. To move an agent from one Secure Agent group to another group, send a request to remove the agent from its current group and then send a request to add the agent to the new group.

Include the following fields in the request:

Field	Type	Required	Description
name	String	Yes	Name of the Secure Agent group.
isShared	Boolean	-	Whether the Secure Agent group can be shared with sub-organizations.
agents	Array	Yes	Agents assigned to the Secure Agent group.
orgId	String	Yes	Organization ID.
id	String	Yes	Agent ID.

POST response

Field	Type	Description
id	String	Secure Agent group ID.
orgId	String	Organization ID.
name	String	Secure Agent group name.
description	String	Description of the Secure Agent group.
createTime	Date/time	Date and time the Secure Agent group was created.
updateTime	Date/time	Date and time that the Secure Agent group was last updated.
createdBy	String	User who created the Secure Agent group.
updatedBy	String	User who last updated the Secure Agent group.

Field	Type	Description
agents	Array	Agents assigned to the Secure Agent group. For more information, see the agent resource.
isShared	Boolean	Indicates whether the Secure Agent group is shared. Returns one of the following values: - true. The Secure Agent group is shared. - false. The Secure Agent group is not shared.
federatedId	String	Global unique identifier.
serverlessConfig		Attribute that defines serverless runtime environment properties.

POST request examples

The following example shows a request to create a Secure Agent group:

```
POST <serverUrl>/api/v2/runtimeEnvironment
Accept:application/json
icSessionId: <icSessionId>
{
  "@type": "runtimeEnvironment",
  "name": "14402",
  "isShared": true
}
```

The following example shows a request to add an agent to a Secure Agent group:

```
POST <serverUrl>/api/v2/runtimeEnvironment/00000425000000000004
Accept:application/json
icSessionId: <icSessionId>
{
  "@type": "runtimeEnvironment",
  "name": "14401",
  "agents": [
    {
      "@type": "agent",
      "id": "01000308000000000003",
      "orgId": "010003"
    }
  ],
  "isShared": false
}
```

POST response example

You might receive a response similar to the following example:

```
{
  "@type": "runtimeEnvironment",
  "id": "01000325000000000005",
  "orgId": "010003",
  "name": "USW1MJ02W6PP-2",
  "createTime": "2021-11-09T17:20:55.583Z",
  "updateTime": "2021-11-09T17:20:55.583Z",
  "createdBy": "ctan",
  "updatedBy": "ctan",
  "agents": [],
  "isShared": false,
  "federatedId": "9F4z0pY1kKqdlmLp5kYXkF",
  "createTimeUTC": "2021-11-09T22:20:55.583Z",
  "updateTimeUTC": "2021-11-09T22:20:55.583Z",
  "serverlessConfig": {
    "cloudProviderConfig": {
      "cloudConfig": []
```

```
            }
          }
        }
```

DELETE request

To delete a Secure Agent group, include the agent group ID as shown in the following URI:

```
/api/v2/runtimeEnvironment/00000425000000000034
```

Managing Secure Agent group selections

You can use the runtimeEnvironment resource to enable or disable Informatica Intelligent Cloud Services and connectors for a Secure Agent group. You can also get details about the selections.

Use a GET request to get the details about a Secure Agent group's enabled and disabled selections. Use a PUT request to assign a Secure Agent group to services and connectors.

GET request

You can request information about the selections that are enabled for a Secure Agent group. You can also request information about all of the selections that are available for a Secure Agent group based on the organization's license. The response includes the names and IDs of the available services and connectors.

To get information about enabled selections, use the following URI:

```
/api/v2/runtimeEnvironment/<id>/selections
```

To get information about all of the selections that are available to a Secure Agent group, use the following URI:

```
/api/v2/runtimeEnvironment/<id>/selections/details
```

GET response

Returns the Secure Agent group selections. If successful, returns the following fields:

Field	Type	Description
services		Contains selection information for services.
selections		Contains information for enabled services. If the request is to include details, also contains selection information for disabled services.
id	String	Included in the selections object. Service ID.
name	String	Included in the selections object. Service name.
enabled	Boolean	Included in the selections object. Whether the service is enabled for the Secure Agent group.
connectors		Contains selection information for connectors.
selections		Contains information for enabled connectors. If the request is to include details, also contains selection information for disabled connectors.

Field	Type	Description
id	String	Included in the selections object. Connector ID.
name	String	Included in the selections object. Connector name.
enabled	Boolean	Included in the selections object. Whether the connector is enabled for the Secure Agent group.
Additional services		Contains selection information for additional services, for example, Git.
selections		Contains information for additional services that are enabled. If the request is to include details, also contains selection information for additional services that are disabled.
id	String	Included in the selections object. Service ID.
name	String	Included in the selections object. Service name.
enabled	Boolean	Included in the selections object. Whether the service is enabled for the Secure Agent group.

GET request example

The following example shows a request to get information about enabled selections:

```
POST <serverUrl>/api/v2/runtimeEnvironment/00000425000000000004/selections
Accept:application/json
icSessionId: <icSessionId>
```

The following example shows a request to get information about enabled and disabled selections:

```
POST <serverUrl>/api/v2/runtimeEnvironment/00000425000000000004/selections/details
Accept:application/json
icSessionId: <icSessionId>
```

GET response example

If the request is for enabled and disabled selections, you might receive a response similar to the following example:

```
{
  "services": {
    "selections": [
      {
        "id": "00000047000000000002",
        "name": "Data Integration",
        "enabled": true
      }
    ]
  },
  "connectors": {
    "selections": [
      {
        "id": "01000000000005L",
        "name": "Amazon Redshift v2",
```

```
                "enabled": true
            },
           {
               "id": "01000000000005Q",
               "name": "Amazon S3",
               "enabled": false
           }
        ]
      },
      "additionalServices": {
        "selections": [
           {
               "id": "00000048000000000002",
               "name": "Git",
               "enabled": false
           }
        ]
      }
   }
}
```

PUT request

To enable services and connectors for a Secure Agent group, include the agent group ID as shown in the following URI:

```
/api/v2/runtimeEnvironment/<id>/selections
```

To see a list of all of the services and connectors that are available and find the selection IDs, use a GET request to obtain information about enabled and disabled selections.

Include the following fields in the request:

Field	Type	Required	Description
id	String	Yes	The selection object ID.
name	String	-	The selection object name.
enabled	Boolean	-	Whether to enable or disable the selection.

PUT request example

If you want to select the Data Integration service, the Amazon Redshift v2 connector, and the Git service for a Secure Agent group, the request might be similar to the following example:

Note: The IDs shown in the example are examples only. Use a GET request to find the selection IDs and names that are available to the Secure Agent group.

```
PUT <serverUrl>/api/v2/runtimeEnvironment/00000425000000000004/selections
Accept:application/json
icSessionId: <icSessionId>
{
   "services": {
     "selections": [
        {
           "id": "00000047000000000002",
           "name": "Data Integration",
           "enabled": true
        }
     ]
   },
   "connectors": {
     "selections": [
        {
           "id": "01000000000005L",
```

```
            "name": "Amazon Redshift v2",
            "enabled": false
          },
          {
            "id": "01000000000005M",
            "name": "Amazon S3 v2",
            "enabled": false
          }
        ]
      },
      "additionalServices": {
        "selections": [
          {
            "id": "00000047000000000013",
            "name": "Self-hosted Git Repo",
            "enabled": true
          }
        ]
      }
    }
  }
```

Configuring Secure Agent service properties for Secure Agent groups

You can use the runtimeEnvironment resource to configure Secure Agent service properties for a Secure Agent group. When you add a Secure Agent to the Secure Agent group, the agent uses the group-level property settings by default instead of using the Secure Agent service default settings.

The Secure Agent service properties that you can configure for a Secure Agent group depend on which services and connectors are assigned to the group. To see the editable properties for a particular Secure Agent group, you can send a GET request for configuration details.

The following rules apply when you configure Secure Agent service properties for a Secure Agent group:

- When you add a Secure Agent to a Secure Agent group that is configured to use group-level property settings, the group-level property settings replace the agent's property settings. If the agent has custom properties, the custom properties are removed.

- When you move a Secure Agent from a Secure Agent group that is configured to use group-level property settings to a group that doesn't use group-level property settings, the agent retains the property settings that were configured at the group level.

Warning: Do not configure agent-level Secure Agent service property settings for an agent in a Secure Agent group that uses group-level property settings. If you want to configure agent-level property settings, delete the group-level property settings before you configure the agent properties.

GET request

Use a GET request to get a list of editable Secure Agent service properties that you can configure for a Secure Agent group.

You can also get a list of group-level properties that have overridden Secure Agent service default property settings.

Include the Secure Agent group ID in the URI. Optionally, include the platform in the URI. You can use linux64 or win64. The default platform is linux64.

Use one of the following URIs:

- To get a list of editable properties, use the following URI:

      /api/v2/runtimeEnvironment/<id>/configs/details/<platform>

- To get a list of group-level properties that override the Secure Agent service default property settings, use the following URI:

        /api/v2/runtimeEnvironment/<id>/configs/*<platform>*

GET response

A successful response returns the Secure Agent services, property types, and the following fields in the property type object:

Field	Type	Description
name	String	Property name.
value	String	Value used for all agents added to the Secure Agent group.
isCustom	Boolean	Whether the property is a custom property.
isSensitive	Boolean	Whether the property value is sensitive data that is masked.

GET response example

If you request the properties that have overridden the Secure Agent service default property settings, you might receive a response similar to the following example:

```
{
    "Data_Integration_Server": [
        {
         "TOMCAT_CFG": [
             {
              "name": "MySQL_JDBC_DRIVER_JARNAME",
              "value": "mysql-connector-java-8.0.13.jar"
             },
             {
              "name": "NetworkTimeoutPeriod",
              "value": "335"
             }
         ]
        },
        {
         "PMRDTM_CFG": [
             {
              "name": "SalesforceConnectionTimeout",
              "value": "311"
             },
             {
              "name": "customConfig-DTM-DIX-linux",
              "value": "custom-new",
              "isCustom": "true"
             }
         ]
        }
    ]
}
```

PUT request

Use to configure Secure Agent service properties for a Secure Agent group. The properties that you configure will override the default property settings for the specified Secure Agent service. Include the Secure Agent group ID in the URI.

Optionally, include the platform in the URI. You can use linux64 or win64. The default platform is linux64.

Use the following URI:

```
/api/v2/runtimeEnvironment/<id>/configs/<platform>
```

In the body of the request, include the Secure Agent service name, the property type, and a name/value pair for the property, for example:

```
{
    "<Secure Agent service name>": [
        {
         "<Property type 1>": [
            {
             "name": "<property name>",
             "value": "<property value>"
            },
            {
             "name": "<property name>",
             "value": "<property value>"
            }
          ]
        },
        {
         "<Property type 2>": [
            {
             "name": "<property name>",
             "value": "<property value>"
            },
            {
             "name": "<property name>",
             "value": "<property value>"
            }
          ]
        }
      ]
}
```

Use the service and property names as shown in a GET response. For example, for the Data Integration Server service, use `Data_Integration_Server`.

The properties that you don't include in the request will retain their existing settings.

Include the following fields in the property type object:

Field	Type	Required	Description
name	String	Yes	Property name.
value	String	Yes	Value to use for all agents added to the Secure Agent group. To use the system default value, use APP_DEFAULT.
isCustom	Boolean	-	Whether the property is a custom property. Default is False.
isSensitive	Boolean	-	Whether the property value is sensitive data that you want to mask. Default is False.

Returns a success code if successful or errors if unsuccessful.

PUT request example

If you want to configure the TOMCAT_CFG property type for the Data Integration Server service, the request might be similar to the following example:

```
PUT <serverUrl>/api/v2/runtimeEnvironment/00000425000000000004/configs
Accept:application/json
```

```
icSessionId: <icSessionId>
{
    "Data_Integration_Server": [
        {
         "TOMCAT_CFG": [
            {
             "name": "MySQL_JDBC_DRIVER_JARNAME",
             "value": "mysql-connector-java-8.0.14.jar"
            },
            {
             "name": "customConfig-Tomcat-DIS-linux",
             "value": "custom-new",
             "isCustom": "true"
            }
         ]
        }
    ]
}
```

DELETE request

You can delete Secure Agent service property settings that were configured for a Secure Agent group. When you delete the group-level properties, the settings for all of the Secure Agents in the group change to the Secure Agent service default settings. Custom properties that were created for the Secure Agent group are removed.

To delete the property settings for a Secure Agent group, include the Secure Agent group ID in the following URI:

```
/api/v2/runtimeEnvironment/<id>/configs
```

Returns a success code if successful or errors if unsuccessful.

schedule

Use this resource to request the details of a schedule or the details of all schedules in the organization. You can create or update a schedule. You can also delete a schedule.

Note: To leverage full scheduling capabilities, use the version 3 schedule resource instead of the version 2 schedule resource.

GET Request

To view the details of all schedules in the organization, use the following URI:

```
/api/v2/schedule
```

To request the details of a particular schedule, you can include the schedule ID or schedule name in the URI. Use one of the following URIs:

```
/api/v2/schedule/<id>
```

```
/api/v2/schedule/name/<name>
```

If you use the schedule name in the URI and the schedule name includes a space, replace the space with `%20`. For example:

```
/api/v2/schedule/name/my%20schedule
```

GET Response

If successful, returns the schedule object for the requested schedule. Or, if you request the details for all schedules, returns the schedule object for each schedule in the organization.

Returns the error object if errors occur.

The schedule object includes the following attributes:

Field	Type	Description
id	String	Schedule ID.
orgId	String	Organization ID.
name	String	Schedule name.
description	String	Description of the schedule.
createTime	Date/time	Time the schedule was created.
updateTime	Date/time	Last time the schedule was updated.
createdBy	String	User who created the schedule.
updatedBy	String	User who last updated the schedule.
startTime	Date/time	Date and time when the schedule starts running.
startTimeUTC	Date/time	Date and time when the schedule starts running the tasks. Uses Coordinated Universal Time (UTC).
endTime	Date/time	Date and time when the schedule stops running .
interval	String	Interval or repeat frequency at which the schedule runs. Returns one of the following codes: - None. The schedule does not repeat. - Minutely. Tasks run on an interval based on the specified number of minutes, days, and time range. - Hourly. Tasks run on an hourly interval based on the start time of the schedule. - Daily. Tasks run on a daily interval based on the start time of the schedule. - Weekly. Tasks run on a weekly interval based on the start time of the schedule. - Biweekly. Tasks run every two weeks based on the start time of the schedule. - Monthly. Tasks run on a monthly interval based on the start time of the schedule.
frequency	Int	Frequency that the schedule runs for the specified interval. For example, if the interval is Hourly, a frequency of 2 means the task runs every 2 hours. Returned for Minutely, Hourly, and Daily intervals only.
rangeStartTime	Date/time	The start of the time range within a day that tasks run. Returned for Minutely and Hourly intervals only.
rangeEndTime	Date/time	The end of the time range within a day that tasks run. Returned for Minutely and Hourly intervals only.
sun	Boolean	Tasks run on Sunday. Returns one of the following codes: - true - false Returned for Minutely, Hourly, Weekly, and Biweekly intervals only.

Field	Type	Description
mon	Boolean	Tasks run on Monday. See description for sun.
tue	Boolean	Tasks run on Tuesday. See description for sun.
wed	Boolean	Tasks run on Wednesday. See description for sun.
thu	Boolean	Tasks run on Thursday. See description for sun.
fri	Boolean	Tasks run on Friday. See description for sun.
sat	Boolean	Tasks run on Saturday. See description for sun.
weekDay	Boolean	Tasks run on weekdays only. Returns one of the following codes: - true - false Returned for Daily interval only.
dayOfMonth	Int	Date of the month that tasks run. Returns a date between 1-28. Returned for Monthly interval only.
weekOfMonth	String	Week of the month that tasks run. Returns one of the following codes: - First. The tasks run in the first week of the month. - Second. The tasks run in the second week of the month. - Third. The tasks run in the third week of the month. - Fourth. The tasks run in the fourth week of the month. - Last. The tasks run in the last week of the month. Returned for Monthly interval only.
dayOfWeek	String	Day of the week that tasks run. Returns one of the following codes: - Day. Tasks run on the first day or last day of the month, based on the selected weekOfMonth option. - Sunday - Monday - Tuesday - Wednesday - Thursday - Friday - Saturday Returned for Monthly interval only.
timeZone	String	Time zone of the user who last updated the schedule. Time zone honors Daylight Saving Time.

POST Request

To update a schedule, use the schedule ID with the following URI. To create a schedule, omit the optional schedule ID.

```
/api/v2/schedule/<id>
```

You can submit a partial update using partial mode. To submit a request using partial mode, use a JSON request and include the following line in the header:

```
Update-Mode=PARTIAL
```

You can use the following attributes in a **schedule** object:

Field	Type	Required	Description
id	String	Required for updates only	Schedule ID.
orgId	String	Yes	Organization ID.
name	String	Yes	Schedule name.
description	String		Description of the schedule.
startTime	Date/time	Yes	Date and time when the schedule starts running.
startTimeUTC	Date/time	Yes	Date and time when the schedule starts running. Uses Coordinated Universal Time (UTC).
endTime	Date/time		Date and time when the schedule stops running. If you do not use this parameter, the schedule runs indefinitely.

Field	Type	Required	Description
interval	String	Yes	Interval or repeat frequency at which the schedule runs. Use one of the following options: - None. Tasks run at the schedule start time. The schedule does not repeat. - Minutely. Tasks run on an interval based on the specified number of minutes, days, and time range. You can use the following parameters: - frequency. Frequency in minutes that tasks run. - sun, mon, tue, wed, thu, fri, sat. The days of the week that tasks run. - startTimeRange and endTimeRange. The time range within a day tasks should run. Do not use if you want tasks to run all day. - endTime. When the schedule should stop running. Do not use if you want the schedule to run indefinitely. - Hourly. Tasks run on an hourly interval based on the start time of the schedule. You can use the following parameters: - frequency. Frequency in hours that tasks run. - sun, mon, tue, wed, thu, fri, sat. The days of the week that tasks run. - startTimeRange and endTimeRange. The time range within a day tasks should run. Do not use if you want tasks to run all day. - endTime. When the schedule should stop running. Do not use if you want the schedule to run indefinitely. - Daily. Tasks run on a daily interval based on the start time configured for the schedule. You can use the following parameters: - frequency. Frequency in days that tasks run. - weekDay. Runs the tasks every weekday. Do not use if you want the tasks to run every day. - endTime. When the schedule should stop running. Do not use if you want the schedule to run indefinitely. - Weekly. Tasks run on a weekly interval based on the start time of the schedule. You can use the following parameters: - sun, mon, tue, wed, thu, fri, sat. The days of the week that tasks run. - endTime. When the schedule should stop running. Do not use if you want the schedule to run indefinitely. - Biweekly. Tasks run every two weeks based on the start time of the schedule. You can use the following parameters: - sun, mon, tue, wed, thu, fri, sat. The days of the week that tasks run. - endTime. When the schedule should stop running. Do not use if you want the schedule to run indefinitely. - Monthly. Tasks run on a monthly interval based on the start time of the schedule. You can use the following parameters: - dayOfMonth. Day of the month when you want tasks to run, between 1-28. - dayOfWeek. Day of the week when you want tasks to run. - weekOfMonth. Week of the month when you want tasks to run. - endTime. When the schedule should stop running. Do not use if you want the schedule to run indefinitely. To indicate when tasks should run, use dayOfWeek with weekOfMonth, such as the First Monday. Or use dayOfMonth, such as 1. Tip: To run tasks on the last day of the month, use the Last weekOfMonth parameter with the Day dayOfWeek parameter.

Field	Type	Required	Description
frequency	Int	Yes	Repeat frequency for tasks. Use one of the following values: - For the Minutely interval, use one of the following options: 5, 10, 15, 20, 30, 45. - For the Hourly interval, use one of the following options: 1, 2, 3, 4, 6, 8, 12. - For Daily intervals, use number of days between 1 -30. Default is 1. Use with Minutely, Hourly, and Daily intervals only.
rangeStartTime	Date/time		The start of the time range within a day that you want tasks to run. Enter a date and time using standard date/time format. Only the time portion is used. Use with Minutely and Hourly intervals only.
rangeEndTime	Date/time		The end of the time range within a day that you want tasks to run. Enter a date and time using standard date/time format. Only the time portion is used. Use with Minutely and Hourly intervals only.
sun	Boolean		Runs tasks on Sunday at the configured time. You can use the sun - sat parameters to run tasks on several days of the week. Use with Minutely, Hourly, Weekly, and Biweekly intervals only.
mon	Boolean		Runs tasks on Monday at the configured time. See description for sun.
tue	Boolean		Runs tasks on Tuesday at the configured time. See description for sun.
wed	Boolean		Runs tasks on Wednesday at the configured time. See description for sun.
thu	Boolean		Runs tasks on Thursday at the configured time. See description for sun.
fri	Boolean		Runs tasks on Friday at the configured time. See description for sun.
sat	Boolean		Runs tasks on Saturday at the configured time. See description for sun.
weekDay	Boolean		Runs tasks on weekdays. Use one of the following options: - True. Run tasks on Monday through Friday. Does not run tasks on the weekend. - False. Run tasks every day. Use with the Daily interval only.
dayOfMonth	Int		Date of the month that tasks should run. Use a date between 1-28. Use with the Monthly interval only. Tip: To run tasks on the last day of the month, use the Last weekOfMonth parameter with the Day dayOfWeek parameter.

Field	Type	Required	Description
weekOfMonth	String		Week of the month that tasks should run. Use with dayOfWeek to specify the day and week of the month that tasks should run. For example, the First Day or the Last Wednesday of the month. Use one of the following options: - First - Second - Third - Fourth - Last Use with the Monthly interval only.
dayOfWeek	String		Day of the week that tasks should run. Use with weekOfMonth to specify the day and week of the month that tasks should run. For example, the First Day or the Last Wednesday of the month. Use one of the following options: - Day - Sunday - Monday - Tuesday - Wednesday - Thursday - Friday - Saturday Use with the Monthly interval only.
timeZone	String		Time zone to use for the schedule. If no valid time zone is passed, Informatica Intelligent Cloud Services uses the user's time zone. For more information, see Appendix A, "Time zone codes" on page 543

POST Response

Returns the schedule response object for the schedule that you created or updated.

Returns an error object if errors occur.

DELETE Request

To delete a schedule, use the schedule ID with the following URI:

```
/api/v2/schedule/<id>
```

DELETE Response

Returns the 200 response code if the request is successful.

Returns an error object if errors occur.

GET Example

To request information about all schedules in the organization, you might use the following request:

```
GET <serverUrl>/api/v2/schedule
Accept: application/json
icSessionId: <icSessionId>
```

A successful request returns a schedule object for each schedule in the organization.

serverTime

Use this resource to return the local time for the Informatica Intelligent Cloud Services server.

GET Request

To request the local time of the Informatica Intelligent Cloud Services server, use the following URI.

```
/api/v2/server/serverTime
```

GET Response

Returns the serverTime object if the request is sucessful. Returns an error object if errors occur.

The serverTime object includes the following attribute:
time

> Local time of the Informatica Intelligent Cloud Services server.

GET Example

To check the local time of the Informatica Intelligent Cloud Services server, you might use the following request:

```
GET <serverUrl>/api/v2/server/serverTime
Accept: application/xml
icSessionId: <icSessionId>
```

task

Use this resource to request a list of tasks of a specified type. You can use this resource to retrieve the name and ID for a task.

Do not use this resource to obtain a task ID to run a job. Instead, use the lookup resource. The lookup resource returns the federated task ID which is required to run a task that is not located in the Default folder.

Do not use this resource for a file ingestion task. Instead, use the file ingestion job resource. For more information see, <u>"tasks resource" on page 449</u>.

GET Request

To request a list of tasks of a specified type, use the task type code in the following URI.

```
/api/v2/task?type=<type>
```

Use the following attribute in the URI:

Field	Required	Description
type	Yes	For Data Integration, use one of the following codes: - DMASK. Masking task. - DRS. Replication task. - DSS. Synchronization task. - MTT. Mapping task. - PCS. PowerCenter task.

GET Response

If the request is successful, returns the task object for every task of the requested type. Returns the error object if errors occur.

The task object includes the following attributes:

Field	Type	Description
id	String	Task ID.
orgId	String	Organization ID.
name	String	Task name.
description	String	Description.
createTime	String	Time the task was created.
updateTime	String	Last time the task was updated.
createdBy	String	User who created the task.
updatedBy	String	User who last updated the task.

GET Example

To view a list of all synchronization tasks, use the following request.

```
/api/v2/task?type=DSS
```

user

Use this resource to request the details of an Informatica Intelligent Cloud Services user account or the details of all user accounts in the organization. If you have administrator privileges, you can also use this resource to create or update a user account and to delete a user account. To ensure organization security, this resource does not display or update the password for a user account.

Note: To leverage full user management capabilities, use the version 3 users resource instead of the version 2 user resource. The version 3 users resource supports users, user groups, and roles. The version 2 user resource does not support user groups and roles, and a GET request might not return all users in the organization.

GET Request

To request the details of all Informatica Intelligent Cloud Services user accounts, use the following URI:

```
/api/v2/user
```

To request the details of a particular Informatica Intelligent Cloud Services user account, you can include the user account ID or user name in the URI. Use one of the following URIs:

```
/api/v2/user/<id>
/api/v2/user/name/<name>
```

If you use the user name in the URI and the user name includes a space, replace the space with `%20`. For example:

```
/api/v2/user/name/Fred%20Smith
```

GET Response

When you request the details for a user account, Informatica Intelligent Cloud Services returns the user object for the requested user account. When you request the details of all user accounts, Informatica Intelligent Cloud Services returns the user object for each user account in the organization.

The user object includes the following attributes:

Field	Type	Description
id	String	User ID.
orgId	String	ID of the organization the user belongs to. 22 characters. **Note:** Organizations that were created in legacy Informatica Cloud might have an organization ID of 6 characters.
orgUuid	String	Unique identifier for the organization.
name	String	Informatica Intelligent Cloud Services user name.
description	String	Description of the user.
createTime	String	When the user account was created.
updateTime	String	When the user account was last updated
createdBy	String	Informatica Intelligent Cloud Services user who created the user account.
updatedBy	String	Informatica Intelligent Cloud Services user who last updated the user account.
sfUsername	String	Salesforce user name. Included when user is configured to authenticate through Salesforce.
password	String	Salesforce user password. Included when user is configured to authenticate through Salesforce.
firstName	String	First name for the user account.
lastName	String	Last name for the user account.
title	String	Title of the user.
phone	String	Phone number for the user.
securityQuestion	String	Security question. Returns one of the following codes: - SPOUSE_MEETING_CITY - FIRST_JOB_CITY - CHILDHOOD_FRIEND - MOTHER_MAIDEN_NAME - PET_NAME - CHILDHOOD_NICKNAME - CUSTOM_QUESTION:"<question>"

Field	Type	Description
securityAnswer	String	Answer to the security question.
roles		Object that includes a role object for each role assigned to the user.
name	String	Included in role object. Role name. Returns one of the following codes: - Service Consumer - Designer - Admin
description	String	Included in role object. Role description.
emails	String	Email address to be notified when the user changes the account password.
timezone	String	Time zone of the user. Time zone honors Daylight Saving Time. For more information, see "Time zone codes" on page 543.
serverUrl	String	Informatica Intelligent Cloud Services URL for the organization the user belongs to. Use the serverUrl as a base for most version 2 REST API resource URIs.
spiUrl	String	This field is no longer applicable and has been deprecated.
uuId	String	Unique identifier for the user.
icSessionId	String	Informatica Intelligent Cloud Services session ID for version 2 REST API session. Use in most version 2 REST API request headers.
forceChangePassword	Boolean	Determines if the user must reset the password after the user logs in for the first time. Includes the following values: - True. The user must reset the password. - False. The user is not forced to reset the password.

POST Request

You must be logged in as an administrator in order to create users or update user details. To update the details of an existing user account, use the user account ID in the following URI.

```
/api/v2/user/<id>
```

To create a new Informatica Intelligent Cloud Services user account, omit the optional user account ID in the URI.

To create a SAML single sign-on user account, do not include password.

You can use the following attributes in a **user** object:

Field	Type	Required	Description
orgId	String	Yes	ID of the organization the user will belong to. 22 characters. **Note:** Organizations that were created in legacy Informatica Cloud might have an organization ID of 6 characters. You can find the organization ID using the login resource.
name	String	Yes	Informatica Intelligent Cloud Services user name. Must be a valid email address when creating a user account using this resource. Maximum length is 255 characters.
sfUsername	String	-	Salesforce user name.
password	String	Yes, unless the user is a SAML single sign-on user.	Informatica Intelligent Cloud Services password. Do not include if the user is a SAML single sign-on user and the organization is SAML enabled. Maximum length is 255 characters.
firstName	String	Yes	First name for the user account.
lastName	String	Yes	Last name for the user account.
title	String	Yes	Title of the user.
phone	String	Yes	Phone number for the user.
emails	String	-	Email address to be notified when the user changes the account password.
description	String	-	Description of the user.
timezone	String	-	Time zone of the user. Time zone honors Daylight Saving Time. Use the appropriate time zone code. If no valid time zone is passed, Informatica Intelligent Cloud Services uses America/Los_Angeles by default. For more information, see .
securityQuestion	String	-	Security question. Use one of the following codes to select the security question: - SPOUSE_MEETING_CITY - FIRST_JOB_CITY - CHILDHOOD_FRIEND - MOTHER_MAIDEN_NAME - PET_NAME - CHILDHOOD_NICKNAME - CUSTOM_QUESTION:"<question>"
securityAnswer	String	-	Answer to the security question.
roles		Yes	Object that includes a role object for each role to be assigned to the user.

Field	Type	Required	Description
name	String	Yes	Include in the role object. Role to assign to the user. Use one of the following values: - Service Consumer - Designer - Admin
description	String	-	Include in the role object. Description of the role.
forceChangePassword	String	-	Determines if the user must reset the password after the user logs in for the first time. Includes the following values: - True. The user must reset the password. - False. The user is not forced to reset the password.

POST Response

Returns the user response object for the requested user account. Or, if you requested information for all user accounts, returns the user response object for each user account in the organization.

Returns an error object if errors occur.

DELETE Request

To delete a user, use the user account ID in the following URI.

```
/api/v2/user/<id>
```

DELETE Response

Returns the 200 response code if the request is successful.

Returns an error object if errors occur.

POST Example

To create a new user, you might use the following request:

```
POST <serverUrl>/api/v2/user/
Content-Type: application/xml
Accept: application/xml
icSessionId: <icSessionId>

<user>
 <orgId>00342000</orgId>
 <name>username@company.com</name>
 <firstName>User</firstName>
 <lastName>Name</lastName>
 <title>developer</title>
 <password>UserPassword</password>
 <phone>5555555555</phone>
 <email>larry@company.com</email>
 <roles>
     <role>
        <name>Designer</name>
     </role>
 </roles>
 <timezone>America/Chicago</timezone>
</user>
```

CHAPTER 3

Platform REST API version 3 resources

The REST API version 3 resources in this section apply to multiple services under Informatica Intelligent Cloud Services.

When you use version 3 resources, note the following rules:

- Use JSON format.

- Use the baseApiUrl value from the login response as the base URL. For example:

  ```
  https://na4.dm-us.informaticacloud.com/saas
  ```

- Use the following URI:

  ```
  /public/core/v3/<API name>
  ```

- Use the following request header format:

  ```
  <METHOD> <baseApiUrl>/<URI> HTTP/<HTTP version>
  Content-Type: application/json
  Accept: application/json
  INFA-SESSION-ID: <SessionId>
  ```

In the following example, the baseApiUrl is `https://na4.dm-us.informaticacloud.com/saas` and the URI is `/public/core/v3/schedule`:

```
<METHOD> https://na4.dm-us.informaticacloud.com/saas/public/core/v3/schedule HTTP/1.1
Content-Type: application/json
Accept: application/json
INFA-SESSION-ID: 9KA11tLGqxVcGeul8SQBK3
```

Note: If you use a tool such as Postman that automatically includes the HTTP version, do not enter the HTTP version in the URL. If the HTTP version appears twice in the URL, the request fails.

Export and import

Use the export and import resources to migrate assets and other objects from one organization to another.

To use the export and import resources, the source and target organizations must have the appropriate license.

To migrate objects, you export them from the source organization and then import them into the target organization.

You can include up to 1000 objects in an export job or import job.

When you export assets, you can choose whether to include dependent objects. During the import operation, you can choose which assets to import.

Informatica Intelligent Cloud Services does not include schedule information when you export an asset. After the import operation, you can associate the imported assets with schedules. Also, when you export and import schedules, the schedules are not associated with any assets.

The export and import resources do not migrate the associated state of the objects from the source organization to the target organization. To migrate the state of the objects that you migrate, use the fetchState and loadState resources.

Secure Agent configuration

You can export the configuration of a Secure Agent and import the configuration at the Secure Agent or Secure Agent group level. You might want to migrate the configuration of a Secure Agent to apply the same configuration to individual Secure Agents or all Secure Agents in a Secure Agent group.

After you export the Secure Agent configuration, you can make revisions to the configuration in the JSON file that's included in the export package before you import it.

Data Integration objects

For Data Integration, you can export and import the following types of objects:

- Mappings
- Tasks
- Advanced taskflows
- Linear taskflows
- Business services
- Fixed-width configuration files
- Hierarchical schemas
- Mapplets
- Saved queries
- Visio templates
- Connections
- Schedules
- Secure Agent configuration

Application Integration objects

For Application Integration, you can export and import the following types of objects:

- Processes
- Guides
- Connections
- Service connectors
- Process objects
- Secure Agent configuration

Exporting objects

Use this resource with the import resource to migrate objects from one organization to another.

Exporting objects includes a series of requests and responses. The end result is a ZIP file that contains the exported objects. To export objects, you perform the following tasks:

1. Log in to the source organization.
2. Send an objects GET request with query parameters to get a list of objects to export. Or, if you already know which objects you want to export, send a lookup GET request to get the object IDs for the objects that you want to export.
   Informatica Intelligent Cloud Services returns the object IDs.

   See "Objects" on page 163 and "Lookup" on page 139.
3. Send an export POST request to start the export job, using the object IDs returned in the objects or lookup response.
   Informatica Intelligent Cloud Services returns the job ID for the export job.

   See "Starting an export job" on page 107.
4. Send an export GET request to get the status of the export job, using the export job ID for the export package.
   Informatica Intelligent Cloud Services returns the job ID and status. The response can also include a list of the objects in the export package.

   See "Getting the export job status" on page 109.
5. Send an export GET request to download the export package.
   Informatica Intelligent Cloud Services returns the export package in a ZIP file.

   See "Downloading an export package" on page 114.

Starting an export job

Use a POST request to start an export job.

POST request

You can export objects such as assets, connections, Secure Agent configurations, and schedules. To specify the objects to export and start the export job, use the following URI:

```
/public/core/v3/export
```

Include the following fields in the request:

Field	Type	Required	Description
name	String		Name of the export job. If a name is not specified, the default name will be used in the following format: job-<currentTimeInMilliseconds>
objects	Collection <complex type>	Yes	Object IDs for objects to export. **Note:** Informatica recommends that you include no more than 1000 objects in an export file.

Field	Type	Required	Description
id	String	Yes	Included in the objects object. Global unique identifier for the export object. This can be a project, folder, or asset ID.
includeDependencies	Boolean		Included in the objects object. Determines whether to include dependent objects for the assets in the export. Default is True.

POST request example

You might use a request similar to the following example:

```
POST <baseApiUrl>/public/core/v3/export
Content-Type: application/json
Accept: application/json
INFA-SESSION-ID: <sessionId>

{
    "name" : "testJob1",
    "objects" : [
        {
            "id": "l7bgB85m5oGiXObDxwnvK9",
            "includeDependencies" : true
        },
        {
            "id": "1MW0GDAE1sFgnvWkvom7mK",
            "includeDependencies" : false
        },
        {
            "id": "iIVBNZSpUKFg4N6g2PKUox"
        }
    ]
}
```

POST response

If successful, returns the following information for the export job:

Field	Type	Description
id	String	ID of the export job.
createTime	String	Time export package was created.
updateTime	String	Time export package was last updated.
name	String	Name of the import job.
startTime	String	Time the export job was started.
endTime	String	Time the export job ended.
status	Complex type	Status of the export.
state	String	Returned in the status object. Status of the export job, such as In Progress, Success, or Failed.

Field	Type	Description
message	String	Returned in the status object. Export job status message.
objects	Collection	Collection of objects. Returns null if blank.

POST response examples

If successful, you might receive a response similar to the following example:

```
{
  "id": "7evG9CokA1whk8ehF3opKM",
  "createTime": "2017-10-26T08:15:48.502Z",
  "updateTime": "2017-10-26T08:15:48.502Z",
  "name": "testJob1",
  "startTime": "2017-10-26T08:15:48.501Z",
  "endTime": null,
  "status": {
    "state": "IN_PROGRESS",
    "message": "In Progress"
  },
  "objects": null
}
```

If you receive an error, you might see a response similar to the following example:

```
{
  "error": {
    "code": "MigrationSvc_034",
    "message": "Invalid object id/s [[242973wgfscbwasd23]]. Object resolution failed.",
    "requestId": "2ataXVlgw3ydI1Yb2MA4sq"
  }
}
```

Getting the export job status

Use a GET request to get the status of an export job or download an export job log.

GET request

To obtain status of the export job, use one of the following URIs:

- To receive status of the export job, use the following URI, where <id> is the export job ID:

  `/public/core/v3/export/<id>`

- To receive status for each object in the export job, use the following URI:

  `/public/core/v3/export/<id>?expand=objects`

Continue polling the request until the state is SUCCESSFUL.

To download the export job log, use the following URI:

`/public/core/v3/export/<id>/log`

GET response

A request for an export job log returns the log in a text file.

A request for status returns the following export status information:

Field	Type	Description
id	String	ID of the export job.
createTime	String	Time the export job was created.
updateTime	String	Last time the export job was updated.
name	String	Name of the export job.
startTime	String	Start time of the export job.
endTime	String	End time of the export job.
status	Complex type	Status of the export job.
state	String	Returned in the status object. State of the export job, such as In Progress, Success, or Failed.
message	String	Returned in the status object. Export job status message.
objects	Collection	Objects in the export job. Returned only when the URI includes `?expand=objects`
id	String	Returned in the objects object. Global unique identifier for the export object requested.
name	String	Returned in the objects object. Name of the object to export.
path	String	Returned in the objects object. Complete path of the object to export.
description	String	Returned in the objects object. Description of the object to export.
status	Complex type	Returned in the objects object. Export status of the individual object.
state	String	Returned in the status object. Export state of the individual object, such as IN PROGRESS, SUCCESS, or FAILED.
message	String	Returned in the status object. Export status message for the individual object.

GET response example

If your request for an export job's status is successful, you might receive a response similar to the following example:

```
{
  "id": "7evG9CokA1whk8ehF3opKM",
  "createTime": "2017-10-26T08:15:48.502Z",
  "updateTime": "2017-10-26T08:15:48.502Z",
```

```
      "name": "testJob1",
      "startTime": "2017-10-26T08:15:48.501Z",
      "endTime": null,
      "status": {
        "state": "IN_PROGRESS",
        "message": "In Progress."
      },
      "objects": null
}
```

If your request included import status for individual objects, a successful response might be similar to the following example:

```
{
  "id": "7evG9CokA1whk8ehF3opKM",
  "createTime": "2017-10-26T08:15:49.000Z",
  "updateTime": "2017-10-26T08:15:50.000Z",
  "name": "testJob1",
  "startTime": "2017-10-26T08:15:49.000Z",
  "endTime": "2017-10-26T08:15:50.000Z",
  "status": {
    "state": "SUCCESSFUL",
    "message": "Export completed successfully."
  },
  "objects": [
    {
      "id": "1MW0GDAE1sFgnvWkvom7mK",
      "name": "Linear Taskflow",
      "path": "/ICS Taskflow",
      "type": "SAAS_LINEAR_TASKFLOW",
      "description": null,
      "status": {
        "state": "SUCCESSFUL",
        "message": null
      }
    },
    {
      "id": "1SuZ9Gf8LtphrJn9EdHCod",
      "name": "SQL Server",
      "path": "/DSS",
      "type": "Folder",
      "description": "",
      "status": {
        "state": "SUCCESSFUL",
        "message": null
      }
    },
    {
      "id": "1Uf9PTj6kTjbsVYMk55OC6",
      "name": "Synchronization Task Multi Source",
      "path": "/Default/SQL Server",
      "type": "SAAS_DSS",
      "description": "",
      "status": {
        "state": "SUCCESSFUL",
        "message": null
      }
    },
    {
      "id": "2UL9ZO9Z3OJeuxbL2cYbaX",
      "name": "Synchronization Task Simple Filter",
      "path": "/Default/SQL Server",
      "type": "SAAS_DSS",
      "description": "",
      "status": {
        "state": "SUCCESSFUL",
        "message": null
      }
    },
    {
```

```
                  "id": "31SzqpeEEKacy7OaXXCfaD",
                  "name": "Synchronization Task Multi Source",
                  "path": "/DSS/SQL Server",
                  "type": "SAAS_DSS",
                  "description": "",
                  "status": {
                    "state": "SUCCESSFUL",
                    "message": null
                  }
                },
                {
                  "id": "5FA0DnMzeuDbYZnn3hdto9",
                  "name": "Default",
                  "path": "/",
                  "type": "Project",
                  "description": "Auto-generated Default Project",
                  "status": {
                    "state": "SUCCESSFUL",
                    "message": null
                  }
                },
                {
                  "id": "5V5VpaoJGTNkWCB2f2t4MG",
                  "name": "Synchronization Task Simple Filter",
                  "path": "/DSS/SQL Server",
                  "type": "SAAS_DSS",
                  "description": "",
                  "status": {
                    "state": "SUCCESSFUL",
                    "message": null
                  }
                },
                {
                  "id": "7udJJF48H5Iizzry8gjUAb",
                  "name": "SQL Server",
                  "path": "/Default",
                  "type": "Folder",
                  "description": "",
                  "status": {
                    "state": "SUCCESSFUL",
                    "message": null
                  }
                },
                {
                  "id": "fIQLvhNnsqBjXKNfjyZFaH",
                  "name": "ICS Taskflow",
                  "path": "/",
                  "type": "Project",
                  "description": "",
                  "status": {
                    "state": "SUCCESSFUL",
                    "message": null
                  }
                },
                {
                  "id": "hGrgtrajWMUjNIsnLKQCAi",
                  "name": "SQL Server Linux",
                  "path": null,
                  "type": "SAAS_CONNECTION",
                  "description": null,
                  "status": {
                    "state": "SUCCESSFUL",
                    "message": null
                  }
                },
                {
                  "id": "iIVBNZSpUKFg4N6g2PKUox",
                  "name": "abc_map",
                  "path": "/Default",
                  "type": "MAPPING",
```

```
          "description": "",
          "status": {
            "state": "SUCCESSFUL",
            "message": null
          }
        },
        {
          "id": "l7bgB85m5oGiXObDxwnvK9",
          "name": "DSS",
          "path": "/",
          "type": "Project",
          "description": "",
          "status": {
            "state": "SUCCESSFUL",
            "message": null
          }
        },
        {
          "id": "lOqsFQE4OSWeyg77AeWwK2",
          "name": "Linux",
          "path": null,
          "type": "SAAS_RUNTIME_ENVIRONMENT",
          "description": null,
          "status": {
            "state": "SUCCESSFUL",
            "message": null
          }
        }
      ]
    }
```

If you requested an export job log, the contents of the text file might be similar to the following example:

```
> OIE_002 INFO 2019-02-05T22:50:08.788Z Starting export operation.
Execution Client: API
Job Name: m_RegionTotalNew-1549407002393
Organization: infa2.doc
RequestId: iklHoZTokKAiNO95Cw9NG3
User: janer2
> OIE_004 INFO 2019-02-05T22:50:09.042Z Successfully exported object [/SYS/
_SYSTEM_PROJECT] of type [Project] id [5UrdDrgV5yKerYgtJAA4IU]> OIE_004 INFO
2019-02-05T22:50:09.042Z Successfully exported object [/Explore/Accounts] of type
[Project] id [8Uyq1wiZ9lye2Sou5OCqOa]
> OIE_004 INFO 2019-02-05T22:50:09.126Z Successfully exported object [/SYS/
_SYSTEM_FOLDER] of type [Folder] id [b98UuC0ADGEkXxF9EIlUCZ]
> OIE_004 INFO 2019-02-05T22:50:09.126Z Successfully exported object [/Explore/Accounts/
February2018] of type [Folder] id [cojSZpHcqcafFy6YkCBgIl]
> OIE_004 INFO 2019-02-05T22:50:09.354Z Successfully exported object [/SYS/TMS26W0864]
of type [SAAS_RUNTIME_ENVIRONMENT] id [6TKTNZ3wfIIjV5yBTJmYWO]
> OIE_004 INFO 2019-02-05T22:50:09.504Z Successfully exported object [/SYS/ff] of type
[SAAS_CONNECTION] id [7GgahDJzE9GbYb75xQ35GM]
> OIE_004 INFO 2019-02-05T22:50:09.765Z Successfully exported object [/Explore/Accounts/
February2018/m_RegionTotalNew] of type [MAPPING] id [4LiKwGKgegAixI2awqWgK1]
> OIE_003 INFO 2019-02-05T22:50:09.843Z Finished export operation.
Job Name: m_RegionTotalNew-1549407002393
Start Time: 2019-02-05T22:50:03.000Z
End Time: 2019-02-05T22:50:09.765Z
Started by: janer2
Start Method: API
Source Organization: infa.doc
Status: SUCCESSFUL
```

Downloading an export package

Use a GET request to download an export package.

GET request

To download the export package, use the following URI:

```
/public/core/v3/export/<id>/package
```

The <id> is the export job ID.

GET response

If successful, you receive the ZIP stream in the response body and the response type will be application/zip.

If unsuccessful, you might receive a response similar to the following example:

```
{
  "error": {
    "code": "MigrationSvc_017",
    "message": "Export request with identifier [asdasduguyvasd8347] doesn't exist.",
    "requestId": "2ataXVlgw3ydI1Yb2MA4sq"
  }
}
```

Identity providers

You can use a JSON web token (JWT) as an OAuth access token to log in to Informatica Intelligent Cloud Services. To log in using a JWT access token, your organization must have a registered identity provider. Use the IdentityProviders resource to register and manage your identity provider.

You can use the IdentityProviders resource to accomplish the following tasks:

- Register an identity provider.
- Get details about an identity provider.
- Update an identity provider.
- Delete an identity provider.

Registering an identity provider

Use a POST request to register an identity provider.

Note: An organization can have no more than one registered identity provider.

POST request

To register an identity provider for an organization, use the following URI:

```
/public/core/v3/Orgs/<orgId>/IdentityProviders
```

Include the following fields in the request:

Field	Type	Required	Description
type	String	Yes	Type of identity provider. Supported type is OIDC.
endPoints	Object	Yes	Object that includes the URLs for the identity provider issuer and the JWT tokens.
issuer	String	Yes	Include in the endPoints object. Absolute URL of the identity provider issuer. Must use the same HTTPS scheme as the key URL and be a subset of the key URL. Maximum length is 255 characters.
keys	String	Yes	Include in the endPoints object. Absolute URL of the JWT tokens. Must use the same HTTPS scheme as the issuer URL. Maximum length is 255 characters.
accountPolicy	Object	Yes	Object that defines the account policy.
link	String	Yes	Include in the accountPolicy object. Object that includes properties for identifying the user.
tokenClaim	String	-	Include in the link object. Name of the claim to be used to identify the user for JWT validation. Default is sub. Maximum length is 64 characters.
matchType	String	Yes	Include in the link object. The Informatica Cloud attribute name to identify the Informatica Intelligent Cloud Services user. Use one of the following values: - aliasName - uid Maximum length is 64 characters.
signingAlgorithm	String	-	Token signing algorithm. Use one of the following values: - HS256 - HS 384 - HS512 - RS256 - RS384 - RS512 - ES256 - ES384 - ES412 - PS256 - PS384 - PS512 - EdDSA - ES256K Default is RS256.

POST request example

You might use a request similar to the following example:

```
https://dm-us.informaticacloud.com/saas/public/core/v3/Orgs/ppbg1kOl2Jo13b/
IdentityProviders
{
  "type": "OIDC",
  "endPoints": {
    "issuer": "https://myidp.mycompany.com/oauth2/",
    "keys": "https://myidp.mycompany.com/oauth2/keys"
  }
  {
   "accountPolicy": {
      "link": {
          "tokenClaim": "sub",
          "matchType": "uid"
      }
  }
  "signingAlgorithm": "HS256"
}
```

POST response

If unsuccessful, returns an error object. If successful, returns the following information:

Field	Type	Description
id	String	User ID.
orgId	String	ID of the organization the user belongs to.
type	String	Type of identity provider. Supported type is OIDC.
endPoints	Object	Object that includes the URLs for the identity provider issues and the JWT tokens.
issuer	String	Included in the endPoints object. Absolute URL of the identity provider issuer.
keys	String	Included in the endPoints object. Absolute URL of the JWT tokens.
accountPolicy	Object	Object that defines the account policy.
link	String	Included in the accountPolicy object. Object that includes properties for identifying the user.
tokenClaim	String	Included in the link object. Name of the claim to be used to identify the user for JWT validation.
matchType	String	Included in the link object. The Informatica Cloud attribute name to identify the Informatica Intelligent Cloud Services user.
signingAlgorithm	String	Token signing algorithm.
createTime	String	When the identity provider was registered.

Field	Type	Description
createdBy	String	Informatica Intelligent Cloud Services user who registered the identity provider.
updateTime	String	When the identity provider was last updated.
updatedBy	String	Informatica Intelligent Cloud Services user who last updated the identity provider.

POST response example

If successful, you might receive a response similar to the following example:

```
{
  "id" : "4aoljyi4ppbg1kOl2Jo13b",
  "orgId" : "ppbg1kOl2Jo13b"
  "type": "OIDC",
  "endPoints": {
    "issuer": "https://myidp.mycompany.com/oauth2/",
    "keys": "https://myidp.mycompany.com/oauth2/keys"
  }    {
  "accountPolicy": {
      "link": {
            "tokenClaim": "sub",
            "matchType": "uid"
      }
  }
  "signingAlgorithm": "HS256",
  "createTime": "2019-03-06T22:04:00.000Z",
  "createdBy": "a@abc.com",
  "updateTime": "2019-03-06T22:04:00.000Z",
  "updatedBy": "a@abc.com"
}
```

Getting identity provider details

Use a GET request to get the information about an identity provider.

GET request

To get information about the identity provider for an organization, use the following URI:

```
/public/core/v3/Orgs/<orgId>/IdentityProviders
```

GET response

If unsuccessful, returns an error object. If successful, returns the following information for the identity provider:

Field	Type	Description
id	String	User ID.
orgId	String	ID of the organization the user belongs to.
type	String	Type of identity provider. Supported type is OIDC.
endPoints	Object	Object that includes the URLs for the identity provider issuer and the JWT tokens.

Field	Type	Description
issuer	String	Included in the endPoints object. Absolute URL of the identity provider issuer.
keys	String	Included in the endPoints object. Absolute URL of the JWT tokens.
accountPolicy	Object	Object that defines the account policy.
link	String	Included in the accountPolicy object. Object that includes properties for identifying the user.
tokenClaim	String	Included in the link object. Name of the claim to be used to identify the user for JWT validation. Default is sub.
matchType	String	Included in the link object. The Informatica Cloud attribute name to identify the Informatica Intelligent Cloud Services user.
signingAlgorithm	String	Token signing algorithm.
createTime	String	When the identity provider was registered.
createdBy	String	Informatica Intelligent Cloud Services user who registered the identity provider.
updateTime	String	When the identity provider was last updated.
updatedBy	String	Informatica Intelligent Cloud Services user who last updated the identity provider.

GET response example

If successful, you might receive a response similar to the following example:

```
[
 {
  "id" : "4aoljyi4ppbg1kOl2Jo13b",
  "orgId" : "ppbg1kOl2Jo13b"
  "type": "OIDC",
  "endPoints": {
    "issuer": "https://myidp.mycompany.com/oauth2/",
    "keys": "https://myidp.mycompany.com/oauth2/keys"
  }     {
  "accountPolicy": {
     "link": {
          "tokenClaim": "sub",
          "matchType": "aliasName"
       }
  }
  "signingAlgorithm": "HS256",
  "createTime": "2019-03-06T22:04:00.000Z",
  "createdBy": "a@abc.com",
  "updateTime": "2019-03-06T22:04:00.000Z",
  "updatedBy": "a@abc.com"
 },
]
```

Updating an identity provider

Use a PUT request to update an identity provider.

PUT request

To update an identity provider, use the following URI:

```
/public/core/v3/Orgs/<orgId>/IdentityProviders/<identity provider ID>
```

You can include the following fields in the request:

Field	Type	Required	Description
endPoints	Object	Yes	Object that includes the URLs for the identity provider issuer and the JWT tokens.
issuer	String	Yes	Include in the endPoints object. Absolute URL of the identity provider issuer. Must use the same HTTPS scheme as the key URL and be a subset of the key URL. Maximum length is 255 characters.
keys	String	Yes	Include in the endPoints object. Absolute URL of the JWT tokens. Must use the same HTTPS scheme as the issuer URL. Maximum length is 255 characters.
accountPolicy	Object	Yes	Object that defines the account policy.
link	String	Yes	Include in the accountPolicy object. Object that includes properties for identifying the user.
tokenClaim	String	-	Include in the link object. Name of the claim to be used to identify the user for JWT validation. Default is sub. Maximum length is 64 characters.

Field	Type	Required	Description
matchType	String	Yes	Include in the link object. The Informatica Cloud attribute name to identify the Informatica Intelligent Cloud Services user. Use one of the following values: - aliasName - uid Maximum length is 64 characters.
signingAlgorithm	String	-	Token signing algorithm. Use one of the following values: - HS256 - HS 384 - HS512 - RS256 - RS384 - RS512 - ES256 - ES384 - ES412 - PS256 - PS384 - PS512 - EdDSA - ES256K Default is RS256.

Returns the 204 response code if successful. Returns an error object if errors occurs.

PUT request example

You might send a request similar to the following example:

```
https://dm-us.informaticacloud.com/saas/public/core/v3/Orgs/ppbg1kOl2Jo13b/
IdentityProviders/6qnnXazBltUbOcUTYgyWO1
{
  "endPoints": {
    "issuer": "https://myidp.mycompany.com/oauth2/",
    "keys": "https://myidp.mycompany.com/oauth2/keys"
  },
  "accountPolicy": {
      "link": {
            "tokenClaim": "user",
            "matchType": "uid"
        }
    }
  "signingAlgorithm": "RS512",
}
```

Deleting an identity provider

To delete an identity provider, include the identity provider ID in a DELETE request.

To delete an identity provider, use the following URI:

```
/public/core/v3/Orgs/<orgId>/IdentityProviders/<identity provider ID>
```

For example,

```
DELETE http://dm-us.informaticacloud.com/saas/public/core/v3/Orgs/ppbg1mOj8Jo14c/
IdentityProviders/6qnnXdzBdtUbObUTYhyWO1
```

Returns the 204 response code if successful. Returns an error object if errors occurs.

Importing objects

Use this resource with the export resource to migrate objects from one organization to another.

Importing objects includes a series of requests and responses. To import objects, you perform the following tasks:

1. Log in to the target organization.
2. Send an import POST request to upload the ZIP file.
   Informatica Intelligent Cloud Services returns the job ID for the import job.

   See "Uploading an import package" on page 121.
3. Send an import POST request to import objects.
   Informatica Intelligent Cloud Services returns the status of the import such as In Progress or Success, or returns an error message. The response also includes the source organization ID for the organization that created the export package.

   See "Starting an import job" on page 122.
4. Send an import GET request to get the status of the import job. You can also request status at the object level.
   Informatica Intelligent Cloud Services returns the status of the import job and if requested, status of each object in the package.

   See "Getting the import job status" on page 126.

Uploading an import package

Use a POST request to upload an import package.

POST request

To upload the import package, use the following URI:

```
/public/core/v3/import/package
```

For Content-Type, use

```
multipart/form-data
```

In the request body, include a part with the name of `package`. For its content, use the export ZIP file that you want to import.

By default, Informatica Intelligent Cloud Services uses checksum validation to verify that no changes were made to the contents of the export ZIP file after it was created. If you want to upload an import package that contains a modified export ZIP file, include the relaxChecksum parameter and set the value to True.

The following image shows an example of the request body in Postman:
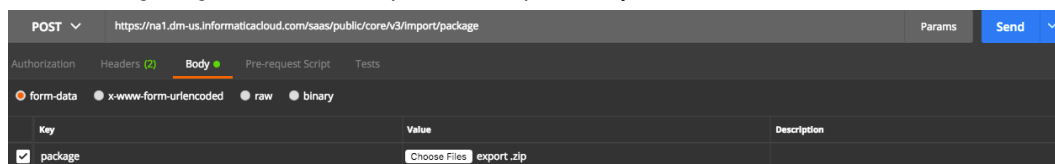
POST response

If successful, returns the following information for the import job:

Field	Type	Description
jobId	String	ID of the import job.
jobStatus	Collection	Status of the package upload.
state	String	Returned in the status object. Status of the import job, such as In Progress, Success, or Failed.
message	String	Returned in the status object. Import job status message.
checksumValid	Boolean	Indicates whether the import package has valid checksum.

POST response example

You might receive a response similar to the following example:

```
{
  "jobId": "2oZb7vFI2QQg4ncd4AyCGn",
  "jobStatus": {
    "state": "NOT_STARTED",
    "message": null
  }
  "checksumValid": true
}
```

Starting an import job

Use a POST request to specify and start an import job. You can import objects such as assets, connections, Secure Agent configurations, and schedules.

POST request

In the request, include a list of objects to import and parameters to resolve any conflict resolution that might occur. An example of a conflict resolution might be if you try to import an asset that has the same name as another asset in the target organization.

You can specify a runtime environment that exists in the target organization to use instead of the source runtime environment provided that the connector types and versions are the same. To find a list of the runtime environments in the target organization, you can use the lookup resource.

To specify the import objects and start the import job, use the following URI:

```
/public/core/v3/import/<id>
```

The <id> is the import job ID received in the POST response for the import package upload.

To get the object IDs that you want to include in the request, you can use the lookup resource. For more information, see .

Include the following fields in the request:

Field	Type	Required	Description
name	String	Yes	Name of the import job. Default name is job-<currentTimeInMilliseconds>
importSpecification	Complex type	-	Used to specify import specifications. By default, the import includes all objects in the import package with default conflict resolution settings.

defaultConflictResolution	String	-	Include in the importSpecification object.
			Whether to overwrite existing objects with all of the objects in the import file. Includes the following options:
			- OVERWRITE. Overwrite the existing objects with the objects in the import file.
			  Default value for assets.
			- REUSE. Use the existing objects. If an object doesn't exist, it is created.
			  Default value for connections, runtime environments, projects, folders, and schedules.

includeObjects	Collection<String>	-	Include in the importSpecification object. Objects to include in the import. You can use the lookup resource to find the object IDs. By default, the import includes all objects in the import package. If the specified object is a project, the import includes all assets that belong to the project and all dependent objects that are not already present in the target organization. If the specified object is an asset, the import creates the asset's containers (project, folder) if they do not already exist. **Note:** Informatica recommends that you include no more than 1000 objects in an import job.
objectSpecification	Collection <complex type>	-	Include in the importSpecification object. Specifies the object properties. If properties are not specified for a particular object, the import uses the default conflict resolution settings.

Field	Type	Required	Description
conflictResolution	String	-	Include in the objectSpecification object. Whether to overwrite an existing asset with an asset in the import file. Includes the following options: - OVERWRITE. Overwrite the existing asset with the asset in the import file. - REUSE. Use the existing object. If the object doesn't exist, it is created. Applicable to assets, for example, Data Integration mapping tasks, Application Integration guides, B2B Gateway suppliers, and Data Quality dictionaries. Do not use for projects, folders, runtime environments, or connections. **Note:** The import includes connections and runtime environments associated with the asset if they do not exist in the target organization.
sourceObjectId	String	Yes	Include in the objectSpecification object. The object ID in the export package file. Required if objectSpecification is present.
targetObjectId	String	-	Include in the objectSpecification object. Used for Container to Container mapping, as well as some asset to asset mappings. Use to specify a connection or runtime environment that exists in the target organization. The target object must use the same connector type and version that is used in the source object.
additionalProviderFlags	Collection <complex type>	-	Additional object specifications. Use key/value pairs to specify additional object specifications, such as such as in-out parameter values and sequence generator values.

POST request example

You might use a request similar to the following example:

```
POST <baseApiUrl>/public/core/v3/import/2oZb7vFI2QQg4ncd4AyCGn
Content-Type: application/json
Accept: application/json
INFA-SESSION-ID: <sessionId>

{
    "name": "ImportName",
    "importSpecification": {
        "defaultConflictResolution": "REUSE",
        "includeObjects": ["iIVBNZSpUKFg4N6g2PKUox","ejZY66c19YUccBdbGwKG4P"],
        "objectSpecification": [
            {
             "sourceObjectId": "iIVBNZSpUKFg4N6g2PKUox",
             "conflictResolution": "OVERWRITE"
            },
            {
             "sourceObjectId": "5FA0DnMzeuDbYZnn3hdto9",
             "targetObjectId": "5KgUiEkW95NkjLRRefWKiG"
             "additionalProviderFlags": [
               {
                "key": "Sequence",
                "value": "26908"
```

```
                }
              ]
            }
          ]
        }
      }
    }
```

POST response

If successful, returns the following information for the import job:

Field	Type	Description
jobId	String	ID of the import job.
createTime	String	Time the import job was created.
updateTime	String	last time the import job was updated.
name	String	Name of the import job.
startTime	String	Start time of the import job.
endTime	String	End time of the import job.
status	Collection	Status of the package upload.
state	String	Returned in the status object. Import state of the individual object, such as IN PROGRESS, SUCCESS, or FAILED.
message	String	Returned in the status object. Import job status message.
objects	Collection	Objects included in the import job.
sourceOrgId	String	Organization ID of the organization that created the export package that was imported.

POST response examples

If successful, you might receive a response similar to the following example:

```
{
  "id": "2oZb7vFI2QQg4ncd4AyCGn",
  "createTime": "2024-02-26T08:40:09.000Z",
  "updateTime": "2024-02-26T08:55:53.238Z",
  "name": "ImportName",
  "startTime": "2024-02-26T08:55:53.232Z",
  "endTime": "2024-02-26T08:53:03.000Z",
  "status": {
    "state": "IN_PROGRESS",
    "message": "In Progress."
  },
  "objects": null,
  "sourceOrgId": "0VOx1gScNH7dlDyA4tD8yX"
}
```

If you receive an error, you might see a response similar to the following example:

```
{
  "error": {
    "code": "MigrationSvc_040",
    "message": "User does not have required permissions.",
    "requestId": "2ataXVlgw3ydI1Yb2MA4sq"
```

```
            }
       }
```

Getting the import job status

Use a GET request to get the status of an import job or download an import job log.

GET request

To obtain status of the import job, use one of the following URIs, where <id> is the import job ID:

- To receive status of the import job, use the following URI:

    `/public/core/v3/import/<id>`

- To receive status for each object in the import job, use the following URI:

    `/public/core/v3/import/<id>?expand=objects`

To download the import job log, use the following URI:

    `/public/core/v3/import/<id>/log`

GET response

A request for an import job log returns the log in a text file.

A request for status returns the following import status information:

Field	Type	Description
id	String	ID of the import job.
createTime	String	Time the import job was created.
updateTime	String	Last time the import job was updated.
name	String	Name of the import job.
startTime	String	Start time of the import job.
endTime	String	End time of the import job.
status	Complex type	Status of the package upload.
state	String	Returned in the status object. Status of the import job, such as IN PROGRESS, SUCCESS, or FAILED.
message	String	Returned in the status object. Import job status message.
sourceOrgId	String	ID of the organization that created the export package that was imported.
objects	Collection	Objects included in the import.
sourceObject	Collection	Returned in the objects object. Object included in the import.
id	String	Returned in the sourceObject object. Global unique identifier for the object included in the import.

Field	Type	Description
name	String	Returned in the sourceObject object. Name of the objectincluded in the import.
path	String	Returned in the sourceObject object. Complete path of the object included in the import.
type	String	Returned in the sourceObject object. Type of object included in the import.
description	String	Returned in the sourceObject object. Description of object included in the import.
targetObject	Collection	Returned in the objects object. Target object.
id	String	Returned in the targetObject object. Global unique identifier for the target object.
name	String	Returned in the targetObject object. Name of the target object.
path	String	Returned in the targetObject object. Complete path of the target object.
type	String	Returned in the targetObject object. Type of target object.
description	String	Returned in the targetObject object. Description of target object.
status	String	Returned in the targetObject object. Status of the target object.
status	Complex type	Returned in the objects object. Import status of the individual object.
state	String	Returned in the status object. Import state of the individual object, such as IN PROGRESS, SUCCESS, or FAILED.
message	String	Returned in the status object. Import status message for the individual object.

GET response example

If your request for an import job's status is successful, you might receive a response similar to the following example:

```
{
  "id": "2oZb7vFI2QQg4ncd4AyCGn",
  "createTime": "2017-10-26T08:40:09.000Z",
  "updateTime": "2017-10-26T08:55:56.000Z",
  "name": "ImportName",
  "startTime": "2017-10-26T08:55:53.000Z",
```

```
      "endTime": "2017-10-26T08:55:56.000Z",
      "status": {
        "state": "SUCCESSFUL",
        "message": "Import completed successfully."
      },
      "objects": null,
      "sourceOrgId": "0VOx1gScNH7dlDyA4tD8yX"
    }
```

If your request included import status for individual objects, a successful response might be similar to the following example:

```
    {
      "id": "2oZb7vFI2QQg4ncd4AyCGn",
      "createTime": "2017-10-26T08:40:09.000Z",
      "updateTime": "2017-10-26T08:55:56.000Z",
      "name": "ImportName",
      "startTime": "2017-10-26T08:55:53.000Z",
      "endTime": "2017-10-26T08:55:56.000Z",
      "status": {
        "state": "SUCCESSFUL",
        "message": "Import completed successfully."
      },
      "objects": [
        {
          "sourceObject": {
            "id": "ejZY66c19YUccBdbGwKG4P",
            "name": "M1",
            "path": "/Default",
            "type": "MAPPING",
            "description": "ab"
          },
          "targetObject": {
            "id": null,
            "name": "M1",
            "path": "/default1",
            "type": "MAPPING",
            "description": null,
            "status": null
          },
          "status": {
            "state": "SUCCESSFUL",
            "message": "Reuse existing."
          }
        },
        {
          "sourceObject": {
            "id": "iIVBNZSpUKFg4N6g2PKUox",
            "name": "abc_map",
            "path": "/Default",
            "type": "MAPPING",
            "description": ""
          },
          "targetObject": {
            "id": null,
            "name": "abc_map",
            "path": "/default1",
            "type": "MAPPING",
            "description": null,
            "status": null
          },
          "status": {
            "state": "SUCCESSFUL",
            "message": "Overwrite existing."
          }
        }
      ],
      "sourceOrgId": "0VOx1gScNH7dlDyA4tD8yX"
    }
```

If you requested an import job log, the contents of the text file might be similar to the following example:

```
> OIE_002 INFO 2019-02-07T01:02:24.986Z Starting import operation.
Execution Client: API
Job Name: ImportExportMapping2-1541009746833
Organization: infa.doc
RequestId: 68srkYNhdSkdKCKfLBGxyd
User: janer2
> OIE_006 INFO 2019-02-07T01:02:25.416Z Successfully imported object [/Explore/
ImportExport] of type [Project] id [3z0FL8tjqEbizNwVBV9LWR] to [/Explore/ImportExport]
> OIE_006 INFO 2019-02-07T01:02:25.931Z Successfully imported object [/SYS/CustFF] of
type [SAAS_CONNECTION] id [76c7oud5pBzlyAC3tdfVK2] to [/SYS/CustFF]
> OIE_006 INFO 2019-02-07T01:02:26.598Z Successfully imported object [/Explore/
ImportExport/ImportExportMapping2] of type [MAPPING] id [09wsnChCzUYl9OWCy6PKIe] to [/
Explore/ImportExport/ImportExportMapping2]
> OIE_003 INFO 2019-02-07T01:02:26.598Z Finished import operation.
Job Name: ImportExportMapping2-1541009746833
Start Time: 2019-02-07T01:02:24.915Z
End Time: 2019-02-07T01:02:26.598Z
Started by: janer2
Start Method: API
Source Organization: infadoc2
Status: SUCCESSFUL
```

IP addresses

Use the Orgs resource to get a list of trusted IP address ranges and add trusted IP address ranges. You can also enable or disable trusted IP address filtering.

Note: A sub-organization's trusted IP ranges are independent of the parent organization's trusted IP ranges.

GET request

To request a list of trusted IP address ranges for an organization or a sub-organization, use the following URI:

```
/public/core/v3/Orgs/<organization ID>/TrustedIP
```

GET response

If the request is successful, the response includes the following information for the organization:

Field	Type	Description
orgId	String	Organization ID.
enableIP	Boolean	Whether IP address filtering is enabled.
ipRanges	List	Trusted IP address ranges for the organization.
startIP	String	Included in the ipRanges object. The first IP address in a range of trusted IP addresses.
endIP	String	Included in the ipRanges object. The last IP address in a range of trusted IP addresses.

GET example

To get a list of trusted IP ranges for an organization, you might send a request similar to the following example:

```
GET <baseApiUrl>/public/core/v3/Orgs/6MRgiMIfvdRfUuCCCLICcI/TrustedIP
```

You might receive a response similar to the following example:

```
{
    "id": "6MRgiMIfvdRfUuCCCLICcI",
    "enableIP": false,
    "ipRanges": [
        {
            "startIP": "10.29.5.1",
            "endIP": "10.29.5.2"
        }
    ]
}
```

PUT request

To add values of trusted IP ranges for an organization or a sub-organization and enable or disable trusted IP ranges, send a PUT request using the following URI:

```
/public/core/v3/Orgs/<organization ID>/TrustedIP
```

Note: If you add trusted IP address ranges for an organization, existing trusted IP address ranges are overwritten.

Include the following information:

Field	Type	Required	Description
enableIP	Boolean	No	Whether to enable IP address filtering. If enabled, at least one IP address range must be specified.
ipRanges	List	No	IP address ranges for the organization.
startIP	String	No	Include in the ipRanges object. The first IP address in a range of trusted IP addresses.
endIP	String	No	Include in the ipRanges object. The last IP address in a range of trusted IP addresses.

PUT response

If the request is successful, the response includes trusted IP address information for the specified organization.

PUT example

To enable the trusted IP addresses feature for an organization and add a range of trusted IP addresses, you might send a request similar to the following example:

```
PUT <baseApiUrl>/public/core/v3/Orgs/6MRgiMIfvdRfUuCCCLICcI/TrustedIP
Content-Type: application/json
Accept: application/json
INFA-SESSION-ID: 9KA11tLGqxVcGeul8SQBK3
{
    "enableIP": true,
    "ipRanges": [
        {
            "startIP": "10.29.4.5",
```

```
            "endIP": "10.29.5.2"
        }
    ]
}
```

You might receive a response similar to the following example:

```
{
    "id": "6MRgiMIfvdRfUuCCCLICcI",
    "enableIP": true,
    "ipRanges": [
        {
            "startIP": "10.29.4.5",
            "endIP": "10.29.5.2"
        }
    ]
}
```

To add multiple ranges of trusted IP addresses, you might send a request similar to the following example:

```
PUT <baseApiUrl>/public/core/v3/Orgs/6MRgiMIfvdRfUuCCCLICcI/TrustedIP
Content-Type: application/json
Accept: application/json
INFA-SESSION-ID: 9KA11tLGqxVcGeul8SQBK3
{
    "enableIP": true,
    "ipRanges": [{"startIP": "10.29.4.5", "endIP":"10.29.5.2"}, {"startIP":
"10.29.10.1", "endIP":"10.29.10.5"}, {"startIP": "10.29.11.1", "endIP":"10.29.11.5"}]
}
```

You might receive a response similar to the following example:

```
{
    "id": "6MRgiMIfvdRfUuCCCLICcI",
    "enableIP": true,
    "ipRanges": [
        {
            "startIP": "10.29.4.5",
            "endIP": "10.29.5.2"
        },
        {
            "startIP": "10.29.10.1",
            "endIP": "10.29.10.5"
        }},
        {
            "startIP": "10.29.11.1",
            "endIP": "10.29.11.5"
        }
    ]
}
```

Key rotation

Use the key resource to get information about the organization's encryption key rotation settings and to change the settings.

You must have the Key Admin role to view or change key rotation settings.

Getting key rotation interval settings

You can use the key resource to see the current key rotation interval for your organization and valid key rotation intervals.

GET request

To get key rotation interval details, use the following URI:

```
/public/core/v3/key/rotationSettings
```

GET response

If successful, returns the following information:

Field	Type	Description
orgId	String	ID of the organization the user belongs to.
validRotationIntervals	List <String>	Valid key rotation intervals. To change the current key rotation interval to one of these values, send a PATCH request.
rotationInterval	String	The current key rotation interval used for the organization.

GET response example

If successful, you might receive a response similar to the following example:

```
{
    "orgId": "52ZSTB0IDK6dXxaEQLUaQu",
    "validRotationIntervals": [
        "90_DAYS",
        "120_DAYS",
        "180_DAYS",
        "365_DAYS"
    ],
    "rotationInterval": "365_DAYS"
}
```

Changing key rotation intervals

You can use the key resource to change the key rotation interval for the organization.

PATCH request

To change the key rotation interval, send a PATCH request using the following URI:

```
/public/core/v3/key/rotationSettings
```

Include the following information:

Field	Type	Required	Description
rotationInterval	String	Yes	The key rotation interval to use for the organization. Use one of the following values: - 90_DAYS - 120_DAYS - 180_DAYS - 365_DAYS Default is 365_DAYS.

PATCH response

Returns a success code if successful or an error object if errors occur.

PATCH example

To change the key rotation interval for an organization, you might send a request similar to the following example:

```
POST <baseApiUrl>/public/core/v3/key/rotationSettings
Content-Type: application/json
Accept: application/json
INFA-SESSION-ID: <sessionId>
{
    "rotationInterval": "120_DAYS"
}
```

Licenses

Use the license resource to get license information about organizations and assign licenses to sub-organizations. In order to assign licenses to a sub-organization, you must log in to the parent organization as an administrator.

You can use the license resource to send the following requests:

- GET request to obtain an organization's editions, custom licenses, and custom limits.

- PUT request to update a sub-organization's license information.

Getting license details

Use a GET request to obtain an organization's editions, custom licenses, and custom limits.

GET request

To request license information for an organization or sub-organization, use the following URI:

```
/public/core/v3/license/org/<orgId>
```

GET response

Returns requested license information if successful or an error object if errors occur.

If successful, returns the following license information for the specified organization ID:

Field	Type	Description
customLicenses	List	Information about the organization's custom licenses.
licenseDef	String	Included in the customLicenses object. The unique identifier for the custom license.
expirationDate	String	Included in the customLicenses object. Time at which the license expires.
licenseType		Included in the custom license object. Type of license. Includes the following values: - TRIAL - SUBSCRIPTION - FREE - NONE
assignedEditions	List	Information about the organization's editions in the edition object.
edition	String	Included in the edition object. Unique identifier for the limit.
expirationDate	String	Included in the edition object. Expiration date for the edition.
customLimits	List	Information about the organization's custom limits.
limitDefinition	String	Included in the customLimit object. Unique identifier for the limit.
value	Integer	Included in the customLimit object. Maximum uses of the limit. Use -1 to indicate there is no maximum for the limit.

GET example

The following example shows a request for an organization's license information:

```
GET <baseApiUrl>/public/core/v3/license/org/1ax3wad2FEsz35asd2892s
Content-Type: application/json
Accept: application/json
INFA-SESSION-ID: <sessionId>
```

The response includes license information for the organization as shown in the following example:

```
{
    "id": "1ax3wad2FEsz35asd2892s",
    "parentOrg": null,
    "customLicenses": [
        {
            "licenseType": "SUBSCRIPTION",
            "expirationDate": "2017-11-05T18:01:24Z",
            "licenseDef": "a5Xjp3VF3sjcyZUDa6UaWh"
        }
    ],
    "assignedEditions": [
        {
```

```
            "expirationDate": "2017-11-05T18:01:24Z",
            "edition": "4sdvnCrYEjfcKjTvAoigEF"
        },
        {
            "expirationDate": "2018-10-06T18:00:08Z",
            "edition": "5SPzPwEFvBEds8LzVwXX4K"
        }
    ],
    "customLimits": [
        {
            "value": -1,
            "limitDefinition": "09cX4Tmi1qSfrS997ORMYl"
        }
    ]
}
```

Updating a sub-organization's licenses

Use a PUT request to update a sub-organization's license information.

PUT request

In order to update licenses for a sub-organization, you must log in to the parent organization as an administrator.

This request overwrites the sub-organization's licenses with the licenses in the request. To make changes to a sub-organization's licenses, first request license information for the sub-organization, make your modifications in the object, and then use it as the request body.

To update license information for a sub-organization, use the following URI:

```
/public/core/v3/license/org/<orgId>
```

PUT response

Returns a success code if successful or an error object if errors occur.

PUT example

To change a sub-organization's licenses, you might use the following request:

```
PUT <baseApiUrl>/public/core/v3/license/org/<orgId>
Accept:application/json
INFA-SESSION-ID: <sessionId>

{
    "customLicenses": [
        {
            "licenseType": "SUBSCRIPTION",
            "expirationDate": "2017-11-05T18:01:24Z",
            "licenseDef": "a5Xjp3VF3sjcyZUDa6UaWh"
        }
    ],
    "assignedEditions": [
        {
            "expirationDate": "2017-11-05T18:01:24Z",
            "edition": "4sdvnCrYEjfcKjTvAoigEF"
        },
        {
            "expirationDate": "2018-10-06T18:00:08Z",
            "edition": "5SPzPwEFvBEds8LzVwXX4K"
        }
    ],
    "customLimits": [
        {
            "value": -1,
            "limitDefinition": "09cX4Tmi1qSfrS997ORMYl"
        }
```

```
        ]
    }
```

Login

Use the login resource to log in to Informatica Intelligent Cloud Services to use version 3 REST API resources.

The login response includes the session ID and base URL that you need to include in the REST API calls that you make during the session.

Use values from the following fields:

- sessionId. A REST API session ID that you include in the header for REST API calls. The session ID expires after 30 minutes of inactivity. After the session ID expires, log in again to continue working with the REST API.
  For information on retrieving session status details, see .

- baseApiUrl. The base URL that you use in all version 3 resource URIs except for login, for example:
  `<baseApiUrl>/public/core/v3/lookup`

Note: Your team might use multiple organizations such as an organization for development and an organization for testing. The user credentials that you use to log in determine the organization that you access.

POST request

To log in, use the following URL:

```
https://<cloud provider>-<region>.informaticacloud.com/saas/public/core/v3/login
```

The values for cloud provider and region correspond to the name of the POD (Point of Deployment) that your organization uses. The following table lists the POD names and the corresponding cloud provider and region to use in the login URL

POD name	Cloud provider-region
USW1	dm-us
USE2	dm-us
USW3	dm-us
USE4	dm-us
USW5	dm-us
USE6	dm-us
USW1-1	dm1-us
USW3-1	dm1-us
USW1-2	dm2-us
CAC1	dm-na

POD name	Cloud provider-region
APSE1	dm-ap
APSE2	dm1-apse
APNE1	dm1-ap
APAUC1	dm1-apau
EMW1	dm-em
EMC1	dm1-em
UK1	dm-uk

For example, if your organization uses the APNE1 POD, use the following URL:

```
https://dm1-ap.informaticacloud.com/saas/public/core/v3/login
```

If you don't know the name of the POD that your organization uses, contact your organization administrator or Informatica Global Customer Support.

For more information about the POD names and corresponding cloud providers and regions, see the Product Availability Matrix (PAM) for Informatica Intelligent Cloud Services on the Knowledge Base.

Use the following fields in a login object:

Field	Type	Required	Description
username	String	Yes	Informatica Intelligent Cloud Services user name for the organization that you want to log in to. Maximum length is 255 characters.
password	String	Yes	Informatica Intelligent Cloud Services password. Maximum length is 255 characters.

POST response

Returns user information if the request is successful. Returns the error object if errors occur.

Use the base URL and session ID returned in the response for subsequent requests during this session.

A successful request returns the following objects:

Field	Type	Description
products	Collection	Subscribed Informatica products.
name	String	Product name.
baseApiUrl	String	Returned in the product object. Base API URL for the product. Use in REST API requests.
userInfo	Collection	User information.

Field	Type	Description
sessionId	String	Returned in the userInfo object. REST API session ID for the current session. Use in most REST API request headers.
id	String	Returned in the userInfo object. User ID.
name	String	User name.
parentOrgId	String	Organization ID for the parent organization.
orgId	String	Returned in the userInfo object. ID of the organization the user belongs to. 22 characters.
orgName	String	Returned in the userInfo object. Organization name.
groups	Collection	User group information for the user.
status	String	Status of the user. Returns one of the following values: - Active - Inactive

POST example

To log in to your Informatica Intelligent Cloud Services organization, you might use the following request:

```
POST https://dm-us.informaticacloud.com/saas/public/core/v3/login
Content-Type: application/json
Accept: application/json

{
    "username": "user@informatica.com",
    "password": "mypassword"
}
```

If successful, the response includes the products and userInfo objects which contain the baseApiUrl and sessionId values to use in subsequent calls, as shown in the following example:

```
{
    "products": [
        {
            "name": "Integration Cloud",
            "baseApiUrl": "https://usw3.dm-us.informaticacloud.com/saas"
        }
    ],
    "userInfo": {
        "sessionId": "9KA11tLGqxVcGeul8SQBK3",
        "id": "9L1GFroXSDHe2IIg7QhBaT",
        "name": "user",
        "parentOrgId": "52ZSTB0IDK6dXxaEQLUaQu",
        "orgId": "0cuQSDTq5sikvN7x8r1xm1",
        "orgName": "MyOrg_INFA",
        "groups": {},
        "status": "Active"
    }
}
```

Using the session ID and base URL values in the above response as an example, to send a GET request to obtain license information, you might use the following request:

```
GET https://usw3.dm-us.informaticacloud.com/saas/public/core/v3/license/org/
0cuQSDTq5sikvN7x8r1xm1
Content-Type: application/json
Accept: application/json
INFA-SESSION-ID: 9KA11tLGqxVcGeul8SQBK3
```

Logout

Use the logout resource to log out of an organization and end the version 3 REST API session specified in the request.

Don't log out of a REST API session until all the requests that you've sent have been executed.

POST request

To log out of an organization and end the version 3 REST API session, include the session ID in the request header with the following URI:

```
https://<cloud provider>-<region>.informaticacloud.com/saas/public/core/v3/logout
```

POST response

Returns the 200 response code if the request is successful or an error object if errors occur.

POST example

To log out of your organization, use the following request:

```
POST https://dm-us.informaticacloud.com/saas/public/core/v3/logout
Content-Type: application/json
Accept: application/json
INFA-SESSION-ID: <sessionId>
```

Lookup

Use the lookup resource to look up an object's ID, name, path, or type attributes.

POST request

This resource is usually used to obtain an object's ID to use in an export request or job request. When you use this resource to obtain an object's ID, include the object path and type in the lookup request.

For a job request, use the value of the id field returned in the lookup response for the federated task ID field in the job request.

To request lookup information, use the following URI:

```
/public/core/v3/lookup
```

You can use the following fields in the objects object:

Field	Type	Required	Description
id	String	Required if object path and type not included.	Global unique identifier for the object.
path	String	Required with type if object ID not included.	Full path of the object including project, folder, and object name.
type	String	Required with path if object ID not included.	Type of object. Use one of the following values: - PROJECT - FOLDER Or use one of the following asset types: For Data Integration, the object can be one of the following types: - DTEMPLATE. Mapping. - MTT. Mapping task. - DSS. Synchronization task. - DMASK. Masking task. - DRS. Replication task. - CONNECTION. - AGENT. Secure Agent. - AGENTGROUP. Runtime environment. - MAPPLET. - BSERVICE. Business service definition. - HSCHEMA. Hierarchical schema. - PCS. PowerCenter task. - FWCONFIG. Fixed width configuration. - CUSTOMSOURCE. Saved query. - MI_FILE_LISTENER. File listener. - MI_TASK. Mass ingestion task. - WORKFLOW. Linear taskflow. - VISIOTEMPLATE - SCHEDULE - SCHEDULE_JOB - SCHEDULE_BLACKOUT. Schedule blackout period. - TASKFLOW For Application Integration, the object can be one of the following types: - PROCESS - GUIDE - AI_CONNECTION - AI_SERVICE_CONNECTOR - PROCESS_OBJECT For B2B Gateway, the object can be one of the following types: - B2BGW_MONITOR - B2BGW_CUSTOMER - B2BGW_SUPPLIER Object types are not case sensitive.

POST response

Returns object information if successful or an error object if errors occur.

If successful, returns the following lookup information for each object:

Field	Type	Description
objects	Collection	Collection of objects for which lookup is requested.
id	String	Global unique identifier for the object. Use the value of this field as the value for taskFederatedId when you submit a job request.
path	String	Full path of the object including project, folder, and object name.
type	String	Type of object.
description	String	Description of the object.
createdBy	String	User who created the object.
updateTime	String	Last time the object was modified.

POST example

The following example shows a lookup request for eight objects:

```
POST <baseApiUrl>/public/core/v3/lookup
Content-Type: application/json
Accept: application/json
INFA-SESSION-ID: 9KA11tLGqxVcGeul8SQBK3

{
    "objects": [{
        "id" : "2iXOKghGpySlgv6ifQImyl"
    }, {
        "path" : "Default/Synchronization Task1",
        "type" : "DSS"
    }, {
        "id" : "hTrrjm1kawScIm1BGEj6UV"
    }, {
        "path" : "My Project",
        "type" : "Project"
    }, {
        "path" : "My Project/DSS Tasks Folder",
        "type" : "Folder"
    }, {
        "path": "USW1R90FPZXD",
        "type": "Agent"
    }, {
        "path": "USW1R90FPZXD",
        "type": "AgentGroup"
    }, {
        "path": "FF_Conn_1",
        "type": "Connection"
    }]
}
```

The response includes lookup information for each object as shown in the following example:

```
{
    "objects": [
        {
            "id": "2iXOKghGpySlgv6ifQImyl",
            "path": "Default/Mapping1",
            "type": "DTEMPLATE",
            "description": "My Mapping 1",
            "updateTime": "2018-04-13T20:44:37Z"
        },
        {
```

```
            "id": "1fOqrwpFvLkimAkFFvIiwl",
            "path": "Default/Synchronization Task1",
            "type": "DSS",
            "description": "Sync Data Task",
            "updateTime": "2018-04-13T20:45:44Z"
        },
        {
            "id": "hTrrjm1kawScIm1BGEj6UV",
            "path": "My Project/Linear Taskflow1",
            "type": "WORKFLOW",
            "description": null,
            "updateTime": "2018-04-13T20:50:31Z"
        },
        {
            "id": "0EzsUXQ1RnkbKD6VyOukCb",
            "path": "My Project",
            "type": "Project",
            "description": "",
            "updateTime": "2018-04-13T20:40:07Z"
        },
        {
            "id": "dRNcMcUVou5lh5kihmEAWl",
            "path": "My Project/DSS Tasks Folder",
            "type": "Folder",
            "description": "DSS Tasks",
            "updateTime": "2018-04-13T20:49:17Z"
        },
        {
            "id": "1a8moeCNtm4fh5vGcUhxOj",
            "path": "USW1R90FPZXD",
            "type": "AGENT",
            "description": null,
            "updateTime": "2018-04-12T19:01:16Z"
        },
        {
            "id": "9iJP8TdBOMujA7eH2CTm8l",
            "path": "USW1R90FPZXD",
            "type": "AgentGroup",
            "description": null,
            "updateTime": "2018-04-12T19:01:17Z"
        },
        {
            "id": "5VkwOw6Jd8RglXEkxDu0ya",
            "path": "FF_Conn_1",
            "type": "Connection",
            "description": null,
            "updateTime": "2018-04-12T21:34:11Z"
        }
    ]
}
```

Metering data

If your organization has the Intelligent Cloud Data Management feature, you can download metering usage data using the REST API.

To download the metering data, you send an export job request, check the job status, and then download a ZIP file that contains the data.

You can download the following data:

Summary IPU usage data

You can download summary IPU usage data for production organizations, additional production organizations, sub-organizations, and sandbox organizations for a specified date range. The ZIP file includes data for the requesting organization and the organizations under it.

For example, if the requesting organization is the production organization, the ZIP file contains data for the production organization and its additional production organizations, sub-organizations, and sandbox organizations. If the requesting organization is an additional production organization that has a sub-organization, the ZIP file contains data for the additional production organization and its sub-organization.

You can send a request to receive usage information for all meters in one CSV file or receive individual CSV files for each meter. To receive usage information for all meters in one CSV file, set the combinedMeterUsage field value to TRUE.

Job-level details

If a service supports job-level detail reports, you can download job-level details for a specified date range. You can request job-level details for a particular meter or for all meters. The ZIP file includes data for the requesting organization.

In the export job request, include a date range and either a meter ID for a particular meter or the allMeters field with a value of TRUE to specify the data to include in the ZIP file. To see if you can download job-level details for your service and to find the meter ID to use in the request, see "Meter IDs" on page 148.

Note: An organization can have a maximum of 5 active jobs. A job is active when the status is CREATED or PROCESSING.

Sending an export job request for IPU usage data

You can run a job to export summary IPU usage data for the parent organization and its linked organizations such as additional production organizations, sub-organizations, and sandbox organizations for a specified date range.

After the job starts, you can check the job status. When the job completes successfully, you can send a request to download the data.

POST request

To request and receive summary IPU usage data, you must have access rights to the parent organization and all of the linked organizations.

To send the export job request, use the following URI:

```
/public/core/v3/license/metering/ExportMeteringDataAllLinkedOrgsAcrossRegion
```

Include the following fields in the request:

Field	Type	Required	Description
startDate	Date/time	Yes	Start of the date range.
endDate	Date/time	Yes	End of the date range. The range can be a maximum of 180 days.

Field	Type	Required	Description
combinedMeterUsage	Boolean	-	Whether to return usage information for all meters in one CSV file. Include one of the following values: - TRUE. Return usage information for all meters in one CSV file. - FALSE. Return usage information in a separate CSV file for each meter. Default is FALSE.
callbackUrl	String	-	A valid, publicly available URL. The service posts the job status to the callbackURL.

POST request example

You might use a request similar to the following example:

```
POST <baseApiUrl>/public/core/v3/license/metering/
ExportMeteringDataAllLinkedOrgsAcrossRegion
Content-Type: application/json
Accept: application/json
INFA-SESSION-ID: <sessionId>

{
    "startDate":"2022-08-12T00:00:00Z",
    "endDate":"2022-10-25T00:00:00Z",
    "combinedMeterUsage":"FALSE"
    "callbackUrl":"https://MyExportJobStatus.com"
}
```

POST response

The response includes the following fields:

Field	Type	Description
jobID	String	ID of the export job.
status	String	Status of the export job.
errorMessage	String	Error message, if any, that is associated with the job.
orgId	String	Organization ID for the requesting organization.
userId	String	ID of the user who sent the request.
selectedOrgId	String	Organization ID for the requesting organization.
startDate	Date/time	Start of the date range.
endDate	Date/time	End of the date range.
callbackUrl	String	The URL that the service uses to post the job status.
createTime	String	Time the export job was created.

Field	Type	Description
updateTIme	String	Time the export job was last updated.
combinedMeterUsage	Boolean	Whether usage data for all meters is returned in one CSV file.

POST response example

If successful, you might receive a response similar to the following example:

```
{
    "jobId": "cYjwJNEF7OckWEmkSC1tSI",
    "status": "CREATED",
    "orgId": "4wMFqGlxAKVlIFTixlTSOY",
    "userId": "aMxRLnYzdWtlkJvDbtPZOr",
    "selectedOrgId": "4wMFqGlxAKVlIFTixlTSOY",
    "startDate": "2020-01-01T10:00:00Z",
    "endDate": "2022-12-31T00:00:00Z",
    "callbackUrl": "https://MyExportJobStatus.com",
    "createTime": null,
    "updateTime": null
    "combinedMeterUsage":"FALSE"
}
```

If an error occurs, the response might be similar to the following example:

```
{
    "jobId": "cYjwJNEF7OckWEmkSC1tSI",
    "status": "FAILED",
    "errorMessage": "Exception occurred while processing request for org:
4wMFqGlxAKVlIFTixlTSOY.\ncom.informatica.cloud.errorutil.MicroServiceException: Get
meter usage request failed with exception: I/O error on POST request",
    "orgId": "4wMFqGlxAKVlIFTixlTSOY",
    "userId": "aMxRLnYzdWtlkJvDbtPZOr",
    "selectedOrgId": "4wMFqGlxAKVlIFTixlTSOY",
    "startDate": "2020-01-01T10:00:00Z",
    "endDate": "2022-12-31T00:00:00Z",
    "callbackUrl": "https://MyExportJobStatus.com",
    "createTime": null,
    "updateTime": null
    "combinedMeterUsage":"FALSE"
}
```

Sending an export job request for job-level metering data

For certain meters, you can run a job to export job-level details for a particular service and meter for a specified date range.

After the job starts, you can check the job status. When the job completes successfully, you can send a request to download the data.

POST request

To send the export job request, use the following URI:

```
/public/core/v3/license/metering/ExportServiceJobLevelMeteringData
```

Include the following fields in the request:

Field	Type	Required	Description
startDate	Date/time	Yes	Start of the date range.
endDate	Date/time	Yes	End of the date range. The range can be a maximum of 180 days.
allMeters	Boolean	Required when meterId is blank.	Whether to return usage information for all meters. Include one of the following values: - TRUE. Return usage information for all meters. - FALSE. Return usage information for meter specified in meterId field. Default is FALSE.
meterId	String	Required when allMeters is FALSE.	Meter ID. For allowable values, see "Meter IDs" on page 148
callbackUrl	String	-	A valid, publicly available URL. The service posts the job status to the callbackURL.

POST request example

You might use a request similar to the following example:

```
POST <baseApiUrl>/public/core/v3/license/metering/ExportServiceJobLevelMeteringData
Content-Type: application/json
Accept: application/json
INFA-SESSION-ID: <sessionId>

{
    "startDate":"2022-08-12T00:00:00Z"
    "endDate":"2022-10-25T00:00:00Z"
    "allMeters":"FALSE"
    "meterId":"3TaYTMo6BFYeNIABfVmH0n"
    "callbackUrl":"https://MyExportJobStatus.com"
}
```

POST response

The response includes the following fields:

Field	Type	Description
jobID	String	ID of the export job.
status	String	Status of the export job.
errorMessage	String	Error message, if any, that is associated with the job.
orgId	String	Organization ID for the requesting organization.
userId	String	ID of the user who sent the request.
selectedOrgId	String	Organization ID for the requesting organization.
allMeters	Boolean	Whether usage information for all meters is included.

Field	Type	Description
meterId	String	Meter ID.
startDate	Date/time	Start of the date range.
endDate	Date/time	End of the date range.
callbackUrl	String	The URL that the service uses to post the job status.
createTime	String	Time the export job was created.
updateTIme	String	Time the export job was last updated.

POST response example

If successful, you might receive a response similar to the following example:

```
{
    "jobId": "cYjwJNEF7OckWEmkSC1tSI",
    "status": "CREATED",
    "orgId": "4wMFqGlxAKVlIFTixlTSOY",
    "userId": "aMxRLnYzdWtlkJvDbtPZOr",
    "selectedOrgId": "4wMFqGlxAKVlIFTixlTSOY",
    "allMeters":"FALSE",
    "meterId": 3TaYTMo6BFYeNIABfVmH0n,
    "startDate": "2020-01-01T10:00:00Z",
    "endDate": "2022-12-31T00:00:00Z",
    "callbackUrl": "https://MyExportJobStatus.com",
    "createTime": null,
    "updateTime": null
}
```

If an error occurs, the response might be similar to the following example:

```
{
    "jobId": "cYjwJNEF7OckWEmkSC1tSI",
    "status": "FAILED",
    "errorMessage": "Exception occurred while processing request for org:
4wMFqGlxAKVlIFTixlTSOY.\ncom.informatica.cloud.errorutil.MicroServiceException: Get
meter usage request failed with exception: I/O error on POST request",
    "orgId": "3TaYTMo6BFYeNIABfVmH0n",
    "userId": "aMxRLnYzdWtlkJvDbtPZOr",
    "selectedOrgId": "3TaYTMo6BFYeNIABfVmH0n",
    "allMeters":"FALSE",
    "meterId": 3TaYTMo6BFYeNIABfVmH0n,
    "startDate": "2020-01-01T10:00:00Z",
    "endDate": "2022-12-31T00:00:00Z",
    "callbackUrl": "https://MyExportJobStatus.com",
    "createTime": null,
    "updateTime": null
}
```

Meter IDs

When you create a request to export job-level metering data, include the meter ID in the request to specify the meter.

The following table lists meter IDs for the meters that support exporting job-level metering data.

Meter name	Meter ID
Data Integration	a2nB20h1o0lc7k3P9xtWS8
Data Integration with Advanced Serverless	35m9fB23Tykj4Fb3rN5q2J
Data Integration Elastic	3TaYTMo6BFYeNIABfVmH0n
Data Integration Elastic with Advanced Serverless	8tXWie0ZQLWlG1cyxxLwQM
Application Integration	3uIRkIV5Rt9lBbAPzeR5Kj
Application Integration with Advanced Serverless	bN6mes5n4GGciiMkuoDlCz
Mass Ingestion Streaming	hr7GsCwFFmyfvfZQFn8v81
Mass Ingestion Database	24WXkCWzeSHjFlQvLPDegF
Mass Ingestion Files	lCwc4CfL7EEhv9773egFC8
Mass Ingestion Application	i3H6LcmMIYjhUKa9VCi7Cl
Advanced Pushdown Optimization	dMN0VeTW4cThHyPovp4GEX
Data Integration - Change Data Capture	0sDTANKFZBSbqjzKaXKlmB
Mass Ingestion Database - Change Data Capture	aluxJ8jOKmzdXwD0JuHRS1
Mass Ingestion Application - Change Data Capture	4cPkZ5cZxjzc4SK2RHoqgy
Integration Hub	fqttkiGnSaHeXW255z4IcD
Catalog Records	5hflTd89LojbuPtkRvPd8o
Governance Records	3sY0I8UHjyTiXcLzZl1fqj
Marketplace Records	6J969du5ipFiRR3hRANP81
Metadata Record Consumption	xQ7fB0n3HtM8JlZk2W6sX1
Scanner Execution	1XzBBPzFTAlk77lGUvVPIS
Scanner Execution with Serverless	beRTNuPByiheJDCgYuIOlo

Getting the metering data export job status

Use a GET request to get the status of an export job for metering data.

GET request

To get the status of the export job, use the following URI:

```
/public/core/v3/license/metering/ExportMeteringData/<jobId>
```

GET response

The response includes the following fields:

Field	Type	Description
jobID	String	ID of the export job.
status	String	Status of the export job. One of the following values can be returned: - CREATED - PROCESSING - SUCCESS - FAILED - PARTIAL_SUCCESS If the status is PARTIAL_SUCCESS, the job completed but data couldn't be collected for all of the linked organizations due to errors.
errorMessage	String	Error message, if any, that is associated with the job.
orgId	String	Organization ID for the requesting organization.
userId	String	ID of the user who sent the request.
selectedOrgId	String	Organization ID for the requesting organization.
meterId	String	Meter ID. Applicable for jobs requesting job-level details.
startDate	Date/time	Start of the date range.
endDate	Date/time	End of the date range.
callbackUrl	String	The URL that the service uses to post the job status.
createTime	String	Time the export job was created.
updateTIme	String	Time the export job was last updated.

GET response example

If successful, you might receive a response similar to the following example:

```
{
    "jobId": "4l1eKlA9ngChmu6gdv3FKs",
    "status": "SUCCESS",
    "orgId": "4wMFqGlxAKVlIFTixlTSOY",
    "userId": "aMxRLnYzdWtlkJvDbtPZOr",
    "selectedOrgId": "4wMFqGlxAKVlIFTixlTSOY",
    "meterId": null,
    "startDate": "2022-08-01T10:00:00Z",
    "endDate": "2022-12-31T00:00:00Z",
```

```
        "callbackUrl": "https://MyExportJobStatus.com",
        "createTime": "2022-11-23T10:04:37Z",
        "updateTime": "2022-11-23T10:04:37Z"
    }
```

If the job is partially successful, the response might be similar to the following example:

```
{
    "jobId": "cYjwJNEF7OckWEmkSC1tSI",
    "status": "PARTIAL_SUCCESS",
    "errorMessage": "Short error message: [fetchEligibleJobForExecution] Exception in
executing
job. org.springframework.web.client.HttpServerErrorException$ServiceUnavailable: 503
Service
Unavailable: \"<html><body><h1>503 Service Unavailable</h1><EOL>No server is available
to handle
this request.<EOL></body></html><EOL>\",
    "orgId": "4wMFqGlxAKVlIFTixlTSOY",
    "userId": "aMxRLnYzdWtlkJvDbtPZOr",
    "selectedOrgId": "4wMFqGlxAKVlIFTixlTSOY",
    "meterId": null,
    "startDate": "2020-01-01T10:00:00Z",
    "endDate": "2022-12-31T00:00:00Z",
    "callbackUrl": "https://MyExportJobStatus.com",
    "createTime": null,
    "updateTime": null
}
```

Downloading the metering data

When the export job completes successfully, use a GET request to download a ZIP file that contains the requested metering data.

The ZIP file is available to download for 3 days after the job completes with a status of SUCCESS or PARTIAL_SUCCESS.

Note: You can download the ZIP file when the status of the job is PARTIAL_SUCCESS. However, the data in the ZIP file might be not be complete.

GET request

To download the ZIP file, use the following URI:

```
public/core/v3/license/metering/ExportMeteringData/<jobId>/download
```

GET response

If successful, you receive the ZIP stream in the response body and the response type will be application/zip. To download the file, save the response to a file as shown in the following image using Postman:
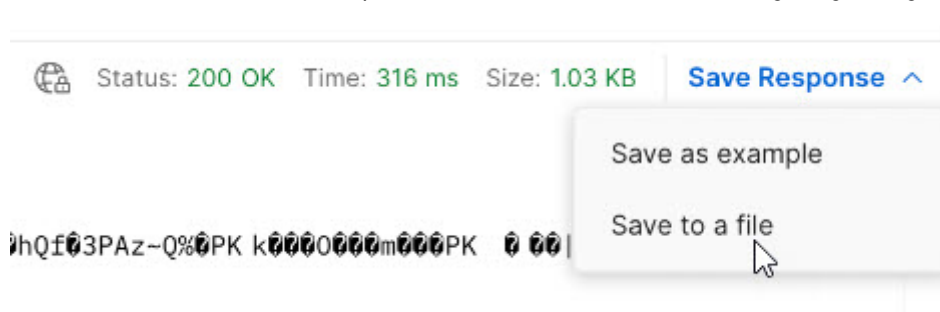
Object state synchronization

If you migrate an object from one organization to another, you can use the fetchState and loadState resources to synchronize the object states and run time attributes between the organizations.

For example, in Organization A, a mapping task with a Sequence Generator transformation has a NEXTVAL value of 3270. The same task was migrated to Organization B, however the NEXTVAL value in Organization B is 0. You want to synchronize the task's state between Organization A and Organization B so that the NEXTVAL value in both organizations has a value of 3270. You use the fetchState and loadState resources to synchronize the NEXTVAL value so that you can run the task in Organization B while preserving the sequence of numbers.

You can make up to 100 fetchState and 100 loadState calls each day.

To use the fetchState and loadState resources, the organizations must have the appropriate license.

The process of synchronizing object states is similar to the process of migrating objects. To synchronize object states, you fetch the states in the primary organization using the fetchState resource, and you load them into the target organization using the loadState resource.

fetchState

Use this resource with the loadState resource to synchronize object states across organizations.

Use the fetchState resource to create an object states package that you upload into other organizations using the loadState resource.

Creating the object states package includes a series of requests and responses, similar to the process of exporting assets. The end result is a ZIP file that contains the object states that you want to load to other organizations. To create the object states package, you perform the following tasks:

1. Send a lookup GET request to receive the object IDs for the object states you want to synchronize. Informatica Intelligent Cloud Services returns the object IDs.

    See .

2. Send a fetchState POST request to start the job, using the object IDs returned in the lookup response. Informatica Intelligent Cloud Services returns the job ID for the fetchState job.

    See .

3. Send a fetchState GET request to get the status of the job, using the fetchState job ID for the object state package.
    Informatica Intelligent Cloud Services returns the job ID and status. The response can also include a list of the object IDs and associated object states that are in the package.

    See .

4. Send a fetchState GET request to download the package.
    Informatica Intelligent Cloud Services returns the package in a ZIP file.

    See .

The object states package contains state information in a JSON file for each object. Each file name uses the following format:

```
<task name>.<task type>.runtime.json
```

For example, a file with the name of `mt_MappingTask106.MTT.runtime.json` might contain the following data:

```
{
  "taskRun" : {
    "lastRuntime" : "2018-12-13T09:05:17.000Z"
```

```
        },
    "taskStateVariables" : [ {
      "category" : "TX_VARIABLE",
      "name" : "Sequence",
      "value" : "26908"
    } ]
  }
```

You can change the following attributes in an object state file if required:

- lastRuntime in the taskRun object
- value in the taskStateVariables object (for mapping tasks only)

Other changes to the files in the package can cause unexpected behavior or errors.

Starting a fetchState job

Use a POST request to start a fetchState job.

POST request

To start the job, use the following URI:

```
/public/core/v3/fetchState
```

Include the following fields in the request:

Field	Type	Required	Description
name	String		Name of the fetchState job. If blank, default is job-<currentTimeInMilliseconds>
objects	Collection <complex type>	Yes	Object IDs for the states to include in the object state package. **Note:** Informatica recommends that you include no more than 1000 objects in a package.
id	String	Yes	Included in the objects object. Global unique identifier for the object for which the state is requested. This can be a project, folder, or asset ID.
includeDependencies	Boolean		Included in the objects object. Determines whether to include the dependent objects' states. Default is True.

POST request example

You might use a request similar to the following example:

```
POST <baseApiUrl>/public/core/v3/fetchstate
Content-Type: application/json
Accept: application/json
INFA-SESSION-ID: <sessionId>

{
    "name" : "fetchStateJob1",
    "objects" : [
        {
            "id": "l7bgB85m5oGiXObDxwnvK9",
            "includeDependencies" : true
        },
        {
            "id": "1MW0GDAE1sFgnvWkvom7mK",
            "includeDependencies" : false
```

```
        },
        {
            "id": "iIVBNZSpUKFg4N6g2PKUox"
        }
    ]
}
```

POST response

If successful, returns the following information for the fetchState job:

Field	Type	Description
id	String	ID of the fetchState job.
createTime	String	Time object state package was created.
updateTime	String	Time object state package was last updated.
name	String	Name of the fetchState job.
startTime	String	Time the fetchState job was started.
endTime	String	Time the fetchState job ended.
status	Complex type	Status of the job.
state	String	Returned in the status object. Status of the fetchState job, such as In Progress, Success, or Failed.
message	String	Returned in the status object. Job status message.
objects	Collection	Collection of objects and object level status. Returns null if blank.

POST response examples

If successful, you might receive a response similar to the following example:

```
{
  "id": "7evG9CokA1whk8ehF3opKM",
  "createTime": "2018-10-26T08:15:48.502Z",
  "updateTime": "2018-10-26T08:15:48.502Z",
  "name": "fetchStateJob1",
  "startTime": "2018-10-26T08:15:48.501Z",
  "endTime": null,
  "status": {
    "state": "IN_PROGRESS",
    "message": "In Progress"
  },
  "objects": null
}
```

If you receive an error, you might see a response similar to the following example:

```
{
  "error": {
    "code": "MigrationSvc_034",
    "message": "User does not have required permissions.",
    "requestId": "2ataXVlgw3ydI1Yb2MA4sq"
  }
}
```

Getting the fetchState job status

Use a GET request to get the status of a fetchState job.

GET request

To get status of the fetchState job, use one of the following URIs:

- To receive status of the fetchState job, use the following URI, where <id> is the fetchState job ID:

  `/public/core/v3/fetchState/<job id>`

- To receive status for each object's state in the fetchState job, use the following URI:

  `/public/core/v3/fetchState/<job id>?expand=objects`

Continue polling the request until the state is SUCCESSFUL.

GET response

A request for status returns the following status information:

Field	Type	Description
id	String	ID of the fetchState job.
createTime	String	Time the fetchState job was created.
updateTime	String	Last time the fetchState job was updated.
name	String	Name of the fetchState job.
startTime	String	Start time of the fetchState job.
endTime	String	End time of the fetchState job.
status	Complex type	Status of the fetchState job.
state	String	Returned in the status object. State of the fetchState job, such as In Progress, Success, or Failed.
message	String	Returned in the status object. Job status message.
objects	Collection	Objects in the fetchState job. Returned when the URI includes `?expand=objects`
id	String	Returned in the objects object. Global unique identifier for the object requested.
name	String	Returned in the objects object. Name of the object..
path	String	Returned in the objects object. Complete path of the object.
description	String	Returned in the objects object. Description of the object.

Field	Type	Description
status	Complex type	Returned in the objects object. Status of the object.
state	String	Returned in the objects.status object. Status of the object, such as IN PROGRESS, SUCCESS, FAILED, or SKIPPED.
message	String	Returned in the objects.status object. Status message for the object.

GET response example

If your request for a fetchState job's status is successful, you might receive a response similar to the following example:

```
{
  "id": "7evG9CokA1whk8ehF3opKM",
  "createTime": "2018-10-26T08:15:48.502Z",
  "updateTime": "2018-10-26T08:15:48.502Z",
  "name": "fetchStateJob1",
  "startTime": "2018-10-26T08:15:48.501Z",
  "endTime": "2018-10-26T08:15:49.501Z",
  "status": {
    "state": "SUCCESSFUL",
    "message": "Export completed successfully."
  },
  "objects": null
}
```

If your request included status for individual objects, a successful response might be similar to the following example:

```
{
  "id": "7evG9CokA1whk8ehF3opKM",
  "createTime": "2017-10-26T08:15:49.000Z",
  "updateTime": "2017-10-26T08:15:50.000Z",
  "name": "fetchStateJob1",
  "startTime": "2018-10-26T08:15:49.000Z",
  "endTime": "2018-10-26T08:15:50.000Z",
  "status": {
    "state": "SUCCESSFUL",
    "message": "Export completed successfully."
  },
  "objects": [
    {
      "id": "1YmwRT083ZtfO04mUABaGF",
      "name": "Mapping1",
      "path": "/Mappings",
      "type": "DTEMPLATE",
      "description": "",
      "status": {
        "state": "SKIPPED",
        "message": null
      }
    },
    {
      "id": "46MhQv9oxrgbOD6qtosF8t",
      "name": "MappingTask1",
      "path": "/Tasks",
      "type": "MTT",
      "description": "",
      "status": {
        "state": "SUCCESSFUL",
        "message": null
```

```
        }
      },
      {
        "id": "7rM2ll1YjWYgHz4xiqRQO3",
        "name": "Default",
        "path": "/",
        "type": "Project",
        "description": "Auto-generated Default Project",
        "status": {
          "state": "SKIPPED",
          "message": null
        }
      },
      {
        "id": "8suj2pxCujqh5Vtmv0DsyP",
        "name": "Destination",
        "path": "null",
        "type": "Connection",
        "description": "Dst Connection",
        "status": {
          "state": "SKIPPED",
          "message": null
        }
      },
      {
            "id": "cpnxnIQMIYvkDOemLhFJ2q",
            "name": "03",
            "path": null,
            "type": "AgentGroup",
            "description": null,
            "status": {
                "state": "SKIPPED",
                "message": null
            }
        },
        {
            "id": "gJvuKZZuBifk9MfZFxtPAb",
            "name": "Source",
            "path": null,
            "type": "Connection",
            "description": "Src Connection",
            "status": {
                "state": "SKIPPED",
                "message": null
            }
        }
    ]
}
```

Downloading an object states package

Use a GET request to download an object states package.

GET request

To download the object states package, use the following URI:

```
/public/core/v3/fetchState/<id>/package
```

The <id> is the fetchState job ID.

GET response

If successful, you receive the ZIP stream in the response body and the response type is application/zip.

If unsuccessful, you might receive a response similar to the following example:

```
{
    "error": {
        "code": "MigrationSvc_017",
```

```
        "message": "Export request with identifier [6GnKs0tkLHdE6Hpd5nsWD] doesnt
exist.",
        "debugMessage": "Export request with identifier [6GnKs0tkLHdE6Hpd5nsWD] doesnt
exist.",
        "requestId": "0FrZZzXiEoafqCZUPqJsYd"
    }
}
```

loadState

Use this resource with the fetchState resource to synchronize object states across multiple organizations.

Loading object states includes a series of requests and responses. To load states into an organization, you perform the following tasks:

1. Send a loadState POST request to upload the ZIP file.
    Informatica Intelligent Cloud Services returns the job ID for the loadState job.

    See "Uploading an object states package" on page 157.

2. Send a loadState POST request to load the object states. The request includes the loadState job ID and a list of object IDs associated with the states you want to load.
    Informatica Intelligent Cloud Services returns the status of the job such as IN PROGRESS or SUCCESS, or returns an error message. The response also includes the source organization ID for the organization that created the object states package.

    See "Starting a loadState job" on page 158.

3. Send a loadState GET request to get the status of the job. You can also request status at the object level.
    Informatica Intelligent Cloud Services returns the status of the job and if requested, status of each object in the package.

    See "Getting the loadState job status" on page 160.

Uploading an object states package

Use a POST request to upload an object states package.

POST request

To upload the object states package, use the following URI:

    /public/core/v3/loadState/package

For Content-Type, use

    multipart/form-data

In the request body, include a part with the name of `package`. For its content, use the object states ZIP file that you want to upload.

POST response

If successful, returns the following information for the loadState job:

Field	Type	Description
jobId	String	ID of the loadState job.
jobStatus	Collection	Status of the package upload.

Field	Type	Description
state	String	Returned in the status object. Status of the loadState job, such as IN PROGRESS, SUCCESS, or FAILED.
message	String	Returned in the status object. Job status message.
checksumValid	Boolean	Indicates whether the object states package has valid checksum.

POST response example

You might receive a response similar to the following example:

```
{
    "jobId": "hUV9Uq1cKYtf8niqF09CWC",
    "jobStatus": {
        "state": "NOT_STARTED",
        "message": null
    },
    "checksumValid": true
}
```

Starting a loadState job

Use a POST request to specify and start a loadState job.

POST request

To specify the objects and start the loadState job, use the following URI:

```
/public/core/v3/loadState/<id>
```

The <id> is the loadState job ID received in the POST response for the object states package upload.

Include the following fields in the request:

Field	Type	Required	Description
name	String		Name of the loadState job. Default name is job-<currentTimeInMilliseconds>
importSpecification	Complex type		Used to specify the objects to include.
includeObjects	Collection<String>		Include in the importSpecification object. Objects to load. If not specified, the load includes all states in the object states package. If the specified object is a project, the load includes state of all objects that belong to the project and all dependent objects.
objectSpecification	Collection <complex type>		Include in the importSpecification object. Specifies the object properties.

Field	Type	Required	Description
sourceObjectId	String	Yes	Include in the objectSpecification object. The container ID in the object states package file. Required if objectSpecification is present.
targetObjectId	String	Yes	Include in the objectSpecification object. Used for Container to Container mapping. Required if objectSpecification is present.

POST request example

You might use a request similar to the following example:

```
POST <baseApiUrl>/public/core/v3/loadState/2oZb7vFI2QQg4ncd4AyCGn HTTP/1.0
Content-Type: application/json
Accept: application/json
INFA-SESSION-ID: <sessionId>

{
    "name" : "stateImportJob",
    "importSpecification" : {
        "includeObjects" : ["iIVBNZSpUKFg4N6g2PKUox","ejZY66c19YUccBdbGwKG4P"],
        "objectSpecification" : [{
            "sourceObjectId" : "iIVBNZSpUKFg4N6g2PKUox"
        },
        {
            "sourceObjectId" : "5FA0DnMzeuDbYZnn3hdto9",
            "targetObjectId" : "5KgUiEkW95NkjLRRefWKiG"
        }]
    }
}
```

POST response

If successful, returns the following information for the loadState job:

Field	Type	Description
Id	String	ID of the loadState job.
createTime	String	Time the loadState job was created.
updateTime	String	Last time the loadState job was updated.
name	String	Name of the loadState job.
startTime	String	Start time of the loadState job.
endTime	String	End time of the loadState job.
status	Collection	Status of the package upload.
state	String	Returned in the status object. Load state for each individual object, such as IN PROGRESS, SUCCESS, FAILED, or SKIPPED.

Field	Type	Description
message	String	Returned in the status object. Job status message.
objects	Collection	Objects included in the loadState job and object level status.
sourceOrgId	String	Organization ID of the organization that created the object states package.
checksumValid	Boolean	Indicates whether the import package has valid checksum.

POST response examples

If successful, you might receive a response similar to the following example:

```
{
    "id": "a7oaBNCyc8DdhxQD4mY4ul",
    "createTime": "2019-01-10T01:35:45.000Z",
    "updateTime": "2019-01-10T21:08:41.398Z",
    "name": "job-1547154520680",
    "startTime": "2019-01-10T21:08:41.389Z",
    "endTime": null,
    "status": {
        "state": "IN_PROGRESS",
        "message": "In Progress"
    },
    "objects": null,
    "sourceOrgId": "2wy21a5fkUphzTVNKaPowg",
    "checksumValid": true
}
```

If you receive an error, you might see a response similar to the following example:

```
{
    "error": {
        "message": "Import request with identifier [a7oaBNCyc8DdhxQD4mY4u] doesnt
exist.",
        "requestId": "9MopwrDFAOGbuMM9utiTqJ"
    }
}
```

Getting the loadState job status

Use a GET request to get the status of a loadState job.

GET request

To get status of the loadState job, use one of the following URIs, where <id> is the loadState job ID:

- To receive status of the loadState job, use the following URI:

  `/public/core/v3/loadState/<id>`

- To receive status for each object in the loadState job, use the following URI:

  `/public/core/v3/loadState/<id>?expand=objects`

GET response

A request for status returns the following import status information:

Field	Type	Description
id	String	ID of the loadState job.
createTime	String	Time the object states package was created.
updateTime	String	Last time the object states package was updated.
name	String	Name of the loadState job.
startTime	String	Start time of the loadState job.
endTime	String	End time of the loadState job.
status	Complex type	Status of the package upload.
state	String	Returned in the status object. Status of the loadState job, such as IN PROGRESS, SUCCESS, or FAILED.
message	String	Returned in the status object. Job status message.
sourceOrgId	String	ID of the organization that created the object states package.
objects	Collection	Objects included in the import.
sourceObject	Collection	Returned in the objects object. Object included in the import.
id	String	Returned in the objects.sourceObject object. Global unique identifier for the object included in the import.
name	String	Returned in the objects.sourceObject object. Name of the object included in the import.
path	String	Returned in the objects.sourceObject object. Complete path of the object included in the import.
type	String	Returned in the objects.sourceObject object. Type of object included in the import.
description	String	Returned in the objects.sourceObject object. Description of the object included in the import.
targetObject	Collection	Returned in the objects object. Target object.
id	String	Returned in the objects.targetObject object. Global unique identifier for the target object.

Field	Type	Description
name	String	Returned in the objects.targetObject object. Name of the target object.
path	String	Returned in the objects.targetObject object. Complete path of the target object.
type	String	Returned in the objects.targetObject object. Type of target object.
description	String	Returned in the objects.targetObject object. Description of target object.
status	String	Returned in the objects.targetObject object. Status of the target object.
status	Complex type	Returned in the objects object. Load status of the object.
state	String	Returned in the objects.status object. Load state of the object, such as IN PROGRESS, SUCCESS, or FAILED.
message	String	Returned in the objects.status object. Status message for the object.
checksumValid	Boolean	Returned in the objects.status object. Whether the checksum of the object was valid or not.

GET response example

If your request for a loadState job's status is successful, you might receive a response similar to the following example:

```
{
  "id": "2oZb7vFI2QQg4ncd4AyCGn",
  "createTime": "2017-10-26T08:40:09.000Z",
  "updateTime": "2017-10-26T08:55:56.000Z",
  "name": "stateImportJob",
  "startTime": "2017-10-26T08:55:53.000Z",
  "endTime": "2017-10-26T08:55:56.000Z",
  "status": {
    "state": "SUCCESSFUL",
    "message": "Import completed successfully."
  },
  "objects": null,
  "sourceOrgId": "2wy21a5fkUphzTVNKaPowg",
  "checksumValid": true
}
```

If your request included load status for individual objects, a successful response might be similar to the following example:

```
{
    "id": "3OpbyDU36UgkUhXXtvGsYA",
    "createTime": "2019-01-10T21:35:33.000Z",
    "updateTime": "2019-01-10T21:35:39.000Z",
    "name": "job-1547156138681",
    "startTime": "2019-01-10T21:35:39.000Z",
    "endTime": "2019-01-10T21:35:39.000Z",
```

```
                    "status": {
                        "state": "SUCCESSFUL",
                        "message": "Import completed successfully."
                    },
                    "objects": [
                        {
                            "sourceObject": {
                                "id": "46MhQv9oxrgbOD6qtosF8t",
                                "name": "MappingTask",
                                "path": "/Default",
                                "type": "MTT",
                                "description": ""
                            },
                            "targetObject": {
                                "id": null,
                                "name": "MappingTask",
                                "path": "/Default",
                                "type": "MTT",
                                "description": null,
                                "status": null
                            },
                            "status": {
                                "state": "SUCCESSFUL",
                                "message": "Overwrite existing."
                            }
                        }
                    ],
                    "sourceOrgId": "2wy21a5fkUphzTVNKaPowg",
                    "checksumValid": true
                }
```

Objects

Use the objects resource to get a list of an organization's assets. You might use this resource to find assets to export.

You can also use this resource to find object dependencies for an asset.

Finding an asset

Use the objects resource to find assets in an organization using query parameters.

Query parameters include filters for asset type, tag, folder location, last update time, the user who last updated the asset, and source control metadata. Query parameters also include the maximum number of assets to return and the number of elements to skip.

The response can include up to 200 assets. If a request uses query parameters that result in more than 200 assets, you can send an additional request using the skip parameter to skip the assets that you received in the previous response.

The response does not include assets that you do not have privileges to read.

GET request

To request a list of assets, use the following URI:

```
/public/core/v3/objects?<query parameters>
```

You can use the following query parameters in the URI:

Parameter	Type	Description
q	String	Query filter.
limit	Int	Maximum number of assets to return, up to 200.
skip	Int	Number of elements to skip. For example, a value of 4 excludes the first four assets in the list.

You can use the following fields to define the query filter:

Field	Type	Operators	Description
type	String	== !=	Asset type. Type can be project, folder, or one of the following asset types: For Data Integration, the object can be one of the following types: - DTEMPLATE. Mapping. - MTT. Mapping task. - DSS. Synchronization task. - DMASK. Masking task. - DRS. Replication task. - DMAPPLET. Mapplet created in Data Integration. - MAPPLET. PowerCenter mapplet. - BSERVICE. Business service definition. - HSCHEMA. Hierarchical schema. - PCS. PowerCenter task. - FWCONFIG. Fixed width configuration. - CUSTOMSOURCE. Saved query. - MI_TASK. Mass ingestion task. - WORKFLOW. Linear taskflow. - VISIOTEMPLATE - TASKFLOW - PROJECT - FOLDER For Application Integration, the object can be one of the following types: - PROCESS - GUIDE - AI_CONNECTION - AI_SERVICE_CONNECTOR - PROCESS_OBJECT For B2B Gateway, the object can be one of the following types: - B2BGW_MONITOR - B2BGW_CUSTOMER - B2BGW_SUPPLIER Object types are not case sensitive.
location	String	==	The project and folder path where the assets are located, such as Default/Sales.

Field	Type	Operators	Description
updateTime	Date	 < <= == => > !=	The last time the assets were updated.
updatedBy	String	== !=	The user who last updated the assets. Use the userName value for the user.
tag	String	==	The tag associated with the assets.
sourceControl.checkedOutBy	String	==, !=	User who checked out the asset.
sourceControl.checkedOutTime	Date	<,<=,==,=>,>, !=	Time the asset was checked out.
sourceControl.hash	String	==, !=	Source control hash. Supports partial hash using a wildcard (*).
sourceControl.lastCheckinBy	String	==, !=	User who last checked in the asset.
sourceControl.lastCheckinTime	Date	<,<=,==,=>,>, !=	The last time the asset was checked in.
sourceControl.lastPullTime	Date	<,<=,==,=>,>, !=	The last time the asset was pulled.
sourceControl.sourceControlled	Boolean	==, !=	Whether the asset is source controlled.
customAttributes.publishedBy	String	==, !=	User who published the asset. Applicable to Application Integration.
customAttributes.publicationDate	Date	<,<=,==,=>,>, !=	Date the asset was published. Applicable to Application Integration.

GET request examples

The following examples show how you can use query parameters to request a list of assets:

- To request a list of Data Integration mapping tasks that were last updated November 21, 2018 or later, you might use the following URI:

```
/public/core/v3/objects?q=type=='MTT' and updateTime>=2018-11-21T12:00:00Z
```

- To request a list of Data Integration mapping tasks that were last updated November 21, 2018 or later, you might use the following URI:

```
/public/core/v3/objects?q=type=='MTT' and updateTime>=2018-11-21T12:00:00Z
```

- To request a list of assets located in the Default/SalesOpps folder that were last updated before March 27, 2018, you might use the following URI:

```
/public/core/v3/objects?q=location=='Default/SalesOpps' and
updateTime<2018-03-27T12:00:00Z
```

- To request a list of assets associated with the UpsellOpps tag that were last updated January 10, 2018 or later, you might use the following URI:

```
/public/core/v3/objects?q=tag=='UpsellOpps' and updateTime>=2018-01-10T12:00:00Z
```

- To request a list of up to 150 assets that were last updated December 30, 2017, excluding Data Integration mappings, you might use the following URI:

  ```
  /public/core/v3/objects?q=type!='MAPPING' and updateTime=2017-12-30T12:00:00Z&max=150
  ```

- To request a list of assets over the limit of 200, you might use the following URI in an additional request:

  ```
  /public/core/v3/objects?limit=200&skip=200
  ```

GET response

If successful, returns a list of assets and a count of the assets based on the query results. If errors occur, returns an error object.

Note: If the result size is large, for example, over 1000 objects, the count returned in the response might not be precise for up to 90 seconds. The fetch list returns all the results using the top and limit parameters included in the request, even though the returned count in the API might not have changed. Also, the count field might not be the first field in the response.

A successful response includes the following information:

Field	Type	Description
count	String	Number of assets matching the filter query.
objects	-	Contains information for each of the assets.
id	String	Global unique identifier for the asset. Use the value of this field as the value for taskFederatedId when you submit a job request.
path	String	Full path of the asset including project, folder, and object name.
type	String	Type of asset.
description	String	Description of the asset.
updatedBy	String	User name of the user who last updated the asset. If the asset is a system-created object such as the Default project and the Add-On Bundles folder, the value for this field is `Informatica`.
updateTime	String	Last time the asset was modified.
tag	String	Tags associated with the asset.
sourceControl	-	Contains source control metadata for the asset.
sourceControl.checkedOutBy	String	User who checked out the object.
sourceControl.checkedOutTime	Date	Time the asset was checked out.
sourceControl.hash	String	Source control hash for the asset.
sourceControl.lastCheckinBy	String	User who checked out the object.
sourceControl.lastCheckinTime	Date	Time that the asset was checked in last.
sourceControl.lastPullTime	Date	Time that the asset was pulled last.
sourceControl.sourceControlled	Boolean	Whether the asset is source controlled.

Field	Type	Description
customAttributes	-	Contains publishing metadata for the Application Integration asset.
customAttributes.publishedBy	String	Included in the customAttributes object. User who published the asset. Applicable to Application Integration.
customAttributes.publicationDate	Date	Included in the customAttributes object. Date that the asset was published. Applicable to Application Integration.

GET example

The following example shows a request to receive a list of assets that are in the P1 folder and limit the response to two assets:

```
GET /saas/public/core/v3/objects?q=location=='P1'&limit=2
Content-Type: application/json
Accept: application/json
INFA-SESSION-ID: 3H05q5PicfolyDXnp3N06c
```

The response includes information for the first two assets as shown in the following example:

```
{
    "count": 4,
    "objects": [
        {
            "id": "1a3TnUrT2cfiwQGtkWQEUy",
            "path": "P1/F1",
            "type": "Folder",
            "description": "",
            "updatedBy": "mma@infa.com",
            "updateTime": "2018-12-17T00:29:29Z"
            "tags": [
                "tag3",
                "tag4"
            ],
            "sourceControl": {
                "checkedOutBy": "mma@infa.com",
                "checkedOutTime": "2020-05-05T17:37:13Z",
                "hash": "3e082fb9bcb2349e9f0a4fb516c739610c869391",
                "lastCheckinTime": "2020-05-05T04:51:09Z",
                "lastCheckinBy": "mma@infa.com",
                "lastPullTime": null,
                "sourceControlled": true
            },
            "customAttributes": {
                "publishedBy": "mma@infa.com"
            }
        },
        {
            "id": "0dGB1jBDWcuhrTxG9Gy1Kh",
            "path": "P1/Mapping1",
            "type": "DTEMPLATE",
            "description": "",
            "updatedBy": "mma@infa.com",
            "updateTime": "2018-12-10T02:25:14Z"
            "tags": [
                "tag3",
                "tag4"
            ],
            "sourceControl": {
                "checkedOutBy": null,
```

```
                    "checkedOutTime": null,
                    "hash": "a98327e09883bb30583574b48113bf1d3ab9d494",
                    "lastCheckinTime": "2020-05-27T20:43:05Z",
                    "lastCheckinBy": "mma@infa.com",
                    "lastPullTime": null,
                    "sourceControlled": true
                },
                "customAttributes": {
                    "publishedBy": "mma@infa.com",
                    "publicationDate": "2020-05-25T11:43:12Z"
                }
            }
        ]
    }
```

Finding asset dependencies

Use the objects resource to get a list of dependencies for an asset. You can receive a list of objects that the asset uses or a list of objects that use the asset.

GET request

To request a list of dependencies for an asset, use the following URI:

```
/public/core/v3/objects/<objectId>/references?<parameters>
```

You can use the following parameters in the URI:

Parameter	Type	Required	Description
refType	Enum	Yes	Whether to list objects that the asset uses or objects that use the asset. Use one of the following values: - uses. Objects that the asset uses. - usedBy. Objects that use the asset. One reference type can be included in a request.
limit	Int	-	Maximum number of objects to return, up to 50. Default is 25.
skip	Int	-	Number of elements to skip from the beginning.. Default is 0.

GET response

If successful, returns a list of dependent objects and a count of the of the dependent objects based on the query results. If errors occur, returns an error object.

Note: If the result size is large, for example, over 1000 objects, the count returned in the response might not be precise for up to 90 seconds. The fetch list returns all the results using the top and limit parameters

included in the request, even though the returned count in the API might not have changed. Also, the count field might not be the first field in the response.

A successful response returns the following information:

Field	Type	Description
id		Global unique identifier for the asset.
count		Number of dependent objects.
references	Collection <complex type>	Includes information for each object that uses or is used by the asset.
id	String	Included in the references object. Global unique identifier for the object.
appContextId	String	Included in the references object. ID of the object in context. To get details or make changes to the object, you can use the appContextId value as the object or task ID in a service-specific REST API call. Applicable only to Mass Ingestion and Data Integration.
path	String	Included in the references object. Full path of the object including project, folder, and object name.
type	String	Included in the references object. Type of object.
description	String	Included in the references object. Description of the object.
updateTime	String	Included in the references object. Last time the object was modified.

GET example

The following example is a request to receive a list of objects that an asset uses with a limit of 25 objects in the response:

```
GET /saas/public/core/v3/objects/1a3TnUrT2cfiwQGtkWQEUy/references?
refType=Uses&skip=0&limit=25
Content-Type: application/json
Accept: application/json
INFA-SESSION-ID: 3H05q5PicfolyDXnp3N06c
```

The response includes a list of objects that the asset uses, as shown in the following example:

```
{
  "id": "1a3TnUrT2cfiwQGtkWQEUy",
  "count": 2,
  "references": [
    {
      "id": "2iXOKghGpySlgv6ifQImyl",
      "appContextId": "N0A1700000000001J",
      "path": "Default/Mapping1",
      "type": "DTEMPLATE",
      "description": "My Mapping 1",
      "updateTime": "2018-04-12T21:34:11Z"
    }
```

```
      {
        "id": "1fOqrwpFvLkimAkFFvIiwl",
        "appContextId": "N0A1700000000001K",
        "path": "FF_Conn_1",
        "type": "Connection",
        "description": null,
        "updateTime": "2018-04-12T21:33:11Z"
      }
    ]
  }
```

Object permissions

Use the objects resource to configure user and user group permissions to objects such as assets, folders, and projects.

To configure permissions for an object, create an access control list (ACL) for each user or user group that might interact with the object. Each ACL specifies a user or user group and their access rights for the object.

For example, you want the NorthDevTeam and WestDevTeam user groups to have read/write access to the NorthwestAccounts project. You create an ACL to specify the NorthDevTeam's access rights to the NorthwestAccounts project, and then create an ACL to specify the WestDevTeam's access rights to the NorthwestAccounts project.

You can use the objects resource to perform the following tasks to manage permissions:

- Create ACLs for an object
- Update ACLs for an object
- Delete ACLs configured for an object
- Get permission details for a user or user group and an object
- Get your permission details for an object

Creating permissions

Create an access control list (ACL) for each user or user group and object pair.

POST request

To create an ACL, send a POST request using the following URI:

```
/public/core/v3/objects/<object ID/permissions
```

Include the following information:

Field	Type	Required	Description
principal	Object	Yes	Whether the ACL is defined for a user or user group and the name of the user or user group.
type	String	Yes	Include in principal object. Use one of the following values: - USER - GROUP

Field	Type	Required	Description
name	String	Yes	Include in principal object. Name of the object.
permissions	Object	Yes	Object permissions to assign to the user or group.
read	Boolean	Yes	Include in permissions object. Whether to allow the user or group to view the object.
delete	Boolean	Yes	Include in permissions object. Whether to allow the user or group to delete the object.
execute	Boolean	Yes	Include in permissions object. Whether to allow the user to run the task.
changePermission	Boolean	Yes	Include in permissions object. Whether to allow the user or group to change the permissions for the object.

POST response

If successful, returns the object with the details you included in the POST request plus the ACL ID.

POST example

To create user permissions for an object, you might send a request similar to the following example:

```
POST <baseApiUrl>/public/core/v3/objects/9EcgvBYZ9GGflOYr98GzOH/permissions
Content-Type: application/json
Accept: application/json
INFA-SESSION-ID: <sessionId>
{
    "principal": {
        "type":"USER",
        "name":"larry@infa.com"
    },
    "permissions" : {
        "read" : true,
        "update" : true,
        "delete" : true,
        "execute" : true,
        "changePermission" : true
    }
}
```

You might receive a response similar to the following example:

```
[
    {
        "id": "0dXigiEiWRbb5rKLgPfFCe",
        "principal": {
            "type": "USER",
            "name": "larry@infa.com"
        },
        "permissions": {
            "read": true,
            "update": true,
            "delete": true,
            "execute": true,
            "changePermission": true
        }
    }
]
```

Updating permissions

You can update an access control list (ACL) to change the access rights that a user or user group has for an object.

PUT request

To update an ACL, send a PUT request using the following URI:

```
/public/core/v3/objects/<object ID>/permissions/<ACL ID>
```

Include the following information:

Field	Type	Required	Description
principal	Object	Yes	Whether the permission is defined for a user or user group and the name of the user or user group.
type	String	Yes	Include in principal object. Use one of the following values: - USER - GROUP
name	String	Yes	Include in principal object. Name of the object.
permissions	Object	Yes	Object permissions to assign to the user or group.
read	Boolean	Yes	Include in permissions object. Whether to allow the user or group to view the object.
delete	Boolean	Yes	Include in permissions object. Whether to allow the user or group to delete the object.
execute	Boolean	Yes	Include in permissions object. Whether to allow the user to run the task.
changePermission	Boolean	Yes	Include in permissions object. Whether to allow the user or group to change the permissions for the object.

PUT response

Returns a success code if successful or an error object if errors occur.

PUT example

To update user permissions for an object, you might send a request similar to the following example:

```
PUT <baseApiUrl>/public/core/v3/objects/9EcgvBYZ9GGflOYr98GzOH/permissions/
4eMlUawPcahhK4eKBmDLsI
Content-Type: application/json
Accept: application/json
INFA-SESSION-ID: <sessionId>
{
    "principal": {
        "type":"USER",
        "name":"scotty@infa.com"
    },
    "permissions" : {
        "read" : true,
```

```
            "update" : true,
            "delete" : true,
            "execute" : false,
            "changePermission" : true
        }
    }
```

Deleting permissions

You can delete the permissions that a user or user group has for an object or delete all of the permissions configured for an object.

To delete the permissions that a user or user group has for an object, send a DELETE request using the following URI:

```
/public/core/v3/objects/<object ID>/permissions/<ACL ID>
```

To delete all of the permissions configured for an object, send a DELETE request using the following URI:

```
/public/core/v3/objects/<object ID>/permissions
```

For example, to delete all of the permissions for an object, you might send a request similar to the following request:

```
DELETE <baseApiUrl>/public/core/v3/objects/9EcgvBYZ9GGflOYr98GzOH/permissions
```

Getting permission details

You can get permission details for particular ACL ID or get the details for all permissions on an object.

GET request

To get the details for a particular ACL ID, send a GET request using the following URI:

```
/public/core/v3/objects/<object ID>/permissions/<ACL ID>
```

To get the details for all permissions on an object, send a GET request using the following URI:

```
/public/core/v3/objects/<object ID>/permissions
```

GET response

If successful, returns the following information for an ACL ID:

Field	Type	Description
id	String	ACL ID
principal	Object	Information about the user or group.
type	String	Whether the principal is a user or group.
name	String	User name or name of the user group.
permissions	Object	Object permissions for the user or group.
read	Boolean	Whether the user or group can view the object.
update	Boolean	Whether the user or group can update the object.

Field	Type	Description
delete	Boolean	Whether the user or group can delete the object.
execute	Boolean	Whether the user or group can run the task.
changePermission	Boolean	Whether the user or group can change the permissions for the object.

Get example

To get the details for all permissions on a object, you might send a request that's similiar to the following example:

```
/public/core/v3/objects/2ymkhUZCl7XbUmN7dsq6Wc/permissions
```

You might receive a response similar to the following example:

```
[
    {
        "id": "4D6ER3yic8cjjE1GmxEKEi",
        "principal": {
            "type": "USER",
            "name": "saki@infa.com"
        },
        "permissions": {
            "read": true,
            "update": true,
            "delete": true,
            "execute": true,
            "changePermission": true
        }
    },
    {
        "id": "6g51tCtEX08bldfh8k6cgz",
        "principal": {
            "type": "GROUP",
            "name": "Everyone"
        },
        "permissions": {
            "read": false,
            "update": true,
            "delete": true,
            "execute": false,
            "changePermission": true
        }
    }
]
```

Checking permissions

You can send a request to check your access rights for a particular object or asset type. The response is based on the permissions and privileges for the user who started the current session.

You might want to check your access rights before you attempt to create an asset in a project or folder. If you don't have permissions for the project or folder, the value of "false" will show for each permission type.

GET request

To get your access rights for an object, send a GET request using the following URI:

```
/public/core/v3/objects/<object ID>/permissions/checkAccess
```

The response includes your access rights for the given object ID.

To find out if you can create a particular type of asset in a project or folder, send a GET request that includes the project or folder ID and the asset type. Use the following URI:

```
/public/core/v3/objects/<object ID>/permissions/checkAccess?type=<asset type>
```

For a list of asset types, see .

GET examples

To check your access rights for an object, you might send a request similar to the following example:

```
GET <baseApiUrl>/public/core/v3/objects/2ymkhUZCl7XbUmN7dsq6Wc/permissions/checkAccess
```

You might receive a response similar to the following example:

```
{
    "permissions": {
        "create": true,
        "read": true,
        "update": true,
        "delete": true,
        "execute": false,
        "changePermission": true
    }
}
```

To see if you have permissions to create a Data Integration mapping in a project, you might send a request similar to the following example:

```
GET <baseApiUrl>/public/core/v3/objects/2ymkhUZCl7XbUmN7dsq6Wc/permissions/checkAccess?
type=DTEMPLATE
```

You might receive a response similar to the following example:

```
{
    "permissions": {
        "create": true,
        "read": true,
        "update": true,
        "delete": true,
        "execute": true,
        "changePermission": true
    }
}
```

Passwords

You can manage passwords using the Informatica Intelligent Cloud Services REST API.

You can perform the following tasks:

- Change your Informatica Intelligent Cloud Services password if your current password has not expired. If you have administrator privileges, you can change other users' passwords.

- Reset a password if it has expired or if you forgot your password.

Changing a password

You can change your Informatica Intelligent Cloud Services password if your current password has not expired. If you have administrator privileges, you can change other users' passwords.

POST request

To change your password, use the following URI:

```
/public/core/v3/Users/ChangePassword
```

Include the following information in the request:

Field	Type	Required	Description
newPassword	String	Yes	New password.
oldPassword	String	Required if you are changing your own password.	Current password.
userId	String	Required if an administrator is changing the password for another user.	Informatica Intelligent Cloud Services user ID.

POST example

To change your password, you might send a request similar to the following example:

```
POST <baseApiUrl>/public/core/v3/Users/ChangePassword
Content-Type: application/json
Accept: application/json
INFA-SESSION-ID: <sessionId>
{
    "newPassword" : "<new password>",
    "oldPassword" : "<old password>"
}
```

A successful request will not return a response. An unsuccessful request will return an error.

Resetting a password

You can reset a password if it has expired or if you forgot your password. To reset a password, you must include the security answer in the request.

POST request

To reset your password, include the security answer in the request. Use the following URI:

```
/public/core/v3/Users/ResetPassword
```

Include the following information in the request:

Field	Type	Required	Description
userId	String	Yes	Informatica Intelligent Cloud Services user ID.
securityAnswer	String	Yes	Security answer to the user's security question.
newPassword	String	Yes	New password for the user.

POST example

To reset your password, you might send a request similar to the following example:

```
POST <baseApiUrl>/public/core/v3/Users/ResetPassword
Content-Type: application/json
Accept: application/json
INFA-SESSION-ID: <sessionId>
{
    "userId" : "5N9JGth6pRYfOGjGKv3Q2D",
    "securityAnswer" : "Simba",
    "newPassword" : "<password>"
}
```

A successful request will not return a response. An unsuccessful request will return an error.

Privileges

Use the privileges resource to obtain a list of privileges that you can use for custom roles.

For information on adding and updating privileges in custom roles, see

GET request

You can request a list of the privileges that you are currently licensed to use. Or, you can request a list of all of the privileges, including the privileges that are disabled due to expired licenses.

To request a list of enabled and default privileges, use the following URI:

```
/public/core/v3/privileges
```

To request a complete list of privileges that includes enabled, disabled, default, and unassigned privileges, include a query parameter in the URI as shown in this example:

```
/public/core/v3/privileges?q=status==All
```

GET response

If successful, returns the following information for each privilege:

Field	Type	Description
id	String	Privilege ID.
name	String	Name of the privilege.
description	String	Description of the privilege.
service	String	The Informatica Intelligent Cloud Services service that applies to the privilege.
status	String	Status of the privilege. Returns one of the following values: - Enabled. License to use the privilege is valid. - Disabled. License to use the privilege has expired. - Unassigned. No license to use this privilege. - Default. Privilege included by default.

GET response example

If you send a request to get all privileges, you might receive a response similar to the following example:

```
[
    {
        "id": "0aoGhY1lAG0iS0PUeLMwoz",
        "name": "changeperm.bservice",
        "description": "",
        "service": "DI",
        "status": "Enabled"
    },
    {
        "id": "0bsvE8I4soacaMt8RHx1yT",
        "name": "update.RULE_SPECIFICATION",
        "description": "update RULE_SPECIFICATION",
        "service": "DQ",
        "status": "Unassigned"
    },
    {
        "id": "0CFJVGBp7Cae8o9EVFakYz",
        "name": "view.RULE_SPECIFICATION",
        "description": "view RULE_SPECIFICATION",
        "service": "DQ",
        "status": "Disabled"
    },
    {
        "id": "11Ai6CJ2PU8jaJwiAwyR0I",
        "name": "changeperm.folder",
        "description": "Change Permission Folder",
        "service": "Admin",
        "status": "Default"
    },
    {
        "id": "11ertBcF3aUkT7vqmn23a3",
        "name": "view.auditLog",
        "description": "view audit log",
        "service": "Admin",
        "status": "Default"
    },
    ....
]
```

Projects and folders

You can create, update, and delete projects and folders through REST API calls.

Note: If a project or folder name contains special characters, space characters, or international characters, use URL encoding in the URI so that the server can properly decode the name. For example, for a project named "My Project," include the following text in the URI:

```
My%20Project
```

You can't use the following characters:

```
# ? ' | { } " ^ & [ ] / \
```

Creating a project

Create projects to organize your assets.

POST request

To create a project, send a POST request using the following URI:

```
/public/core/v3/projects
```

Note: You can create a maximum of 500 projects usingInformatica Intelligent Cloud Services REST API.

Include the following information:

Field	Type	Required	Description
name	String	Yes	Name of the project.
description	String	-	Description of the project.

POST response

If successful, returns the projects object with the following details:

Field	Type	Description
id	String	Project ID.
name	String	Project name.
description	String	Project description.
updatedBy	String	User who created the project.
updateTime	String	Date and time the project was created.

POST example

To create a project, you might send a request similar to the following example:

```
POST <baseApiUrl>/public/core/v3/projects
Content-Type: application/json
Accept: application/json
INFA-SESSION-ID: <sessionId>
{
    "name":"Orders",
    "description": "Orders from all regions"
}
```

You might receive a response similar to the following example:

```
{
    "id": "kTKQG4Kv3uhgQleaOoUbWu",
    "name": "Orders",
    "description": "Orders from all regions",
    "updatedBy": "scotty@infa.com",
    "updateTime": "2022-04-12T23:40:40.395Z"
}
```

Updating a project

You can change a project's name or description if you have update permission for the project.

PATCH request

To update a project, send a PATCH request using one of the following URIs:

    /public/core/v3/projects/<project ID>

    /public/core/v3/projects/name/<project name>

Include the following information:

Field	Type	Required	Description
name	String	-	Name of the project.
description	String	-	Description of the project.

PATCH response

Returns a success code if successful or an error object if errors occur.

PATCH example

To update a project, you might send a request similar to the following example:

```
PATCH <baseApiUrl>/public/core/v3/projects/kTKQG4Kv3uhgQleaOoUbWu
Content-Type: application/json
Accept: application/json
INFA-SESSION-ID: <sessionId>
{
    "name":"Requests",
    "description": "Orders from all regions"
}
```

Deleting a project

You can delete a project from your organization if it doesn't contain any folders or assets and if you have delete permission for the project.

DELETE request

To delete a project, use one of the following URIs:

    /public/core/v3/projects/<project ID>

    /public/core/v3/projects/name/<project name>

DELETE example

To delete a project named "All Orders," you might send the following request:

    DELETE <baseApiUrl>/public/core/v3/projects/name/All%20Orders

Creating a folder

You can create a folder in the Default project or specify a different project.

Note: You can create a maximum of 5000 folders usingInformatica Intelligent Cloud Services REST API.

POST request

To create a folder in the Default project, use the following URI:

```
/public/core/v3/folders
```

To create a folder in a project other than the Default project, use one of the following URIs:

```
/public/core/v3/projects/<project ID>/folders
```

```
/public/core/v3/projects/name/<project name>/folders
```

Include the following information:

Field	Type	Required	Description
name	String	Yes	Name of the folder.
description	String	-	Description of the folder .

POST response

If successful, returns the folder object with the following details:

Field	Type	Description
id	String	Folder ID.
name	String	Folder name.
description	String	Folder description.
updatedBy	String	User who created the folder.
updateTime	String	Date and time the folder was created.

POST example

To create a folder in the Orders project, you might send a request similar to the following example:

```
POST <baseApiUrl>/public/core/v3/projects/name/Orders/folders
Content-Type: application/json
Accept: application/json
INFA-SESSION-ID: <sessionId>
{
    "name":"Northwest Orders",
    "description": "Orders from northwest offices"
}
```

You might receive a response similar to the following example:

```
{
    "id": "kTrYZ8fInr1Uw3ugQbOuWe",
    "name": "Northwest Orders",
    "description": "Orders from northwest offices",
    "updatedBy": "scotty@infa.com",
    "updateTime": "2022-04-13T18:20:40.325Z"
}
```

Updating a folder

You can change a folder's name or description if you have update permission for the folder.

PATCH request

To update a folder that's in the Default project, send a PATCH request using the following URI:

```
/public/core/v3/folders/<folder ID>
```

To update a folder that's in a project other than the Default project, use one of the following URIs:

```
/public/core/v3/projects/<project ID>/folders/<folder ID>
```

```
/public/core/v3/projects/name/<project name>/folders/name/<folder name>
```

Note: You can include project and folder IDs or project and folder names in a URI, but you can't include an ID and a name in a URI. For example, you can't include the project ID and the folder name.

Include the following information:

Field	Type	Required	Description
name	String	-	Name of the folder.
description	String	-	Description of the folder.

PATCH response

Returns a success code if successful or an error object if errors occur.

PATCH example

To change the name and description of the Northeast Orders folder, you might send a request similar to the following example:

```
PATCH <baseApiUrl>/public/core/v3/projects/name/Orders/folders/name/Northeast%20Orders
Content-Type: application/json
Accept: application/json
INFA-SESSION-ID: <sessionId>
{
    "name":"Northeast Requests",
    "description": "Requests from the northeast offices"
}
```

Deleting a folder

You can delete a folder if it is empty and you have delete permission for the folder. If the folder contains assets, you must delete or move the assets before you can delete it.

DELETE request

To delete a folder that's in the Default project, use the following URI:

```
/public/core/v3/projects/<folder ID>
```

To delete a folder that's in a project other than the Default project, use one of the following URIs:

```
/public/core/v3/projects/<project ID>/folders/<folder ID>
```

```
/public/core/v3/projects/name/<project name>/folders/name/<folder name>
```

Note: You can include project and folder IDs or project and folder names in a URI, but you can't include an ID and a name in a URI. For example, you can't include the project ID and the folder name.

DELETE example

To delete a folder named "Orders:West Coast," you might send the following request:

```
DELETE <baseApiUrl>/public/core/v3/projects/name/Orders/folders/name/Orders%3AWest
%20Coast
```

Roles

A role is a collection of privileges that you can assign to users and groups. Use the roles resource to get the details for roles in your organization. You can also use this resource to create, update, and delete custom roles.

Getting role details

You can request the details for all of your organization's roles or request the details for a particular role.

GET request

To get role details, use the following URI:

```
/public/core/v3/roles
```

To get the details for a particular role, you can include the following query parameters in the URI:

Parameter	Type	Description
q	String	Query filter. You can filter using one of the following fields: - roleId. Unique identifier for the role. - roleName. Name of the role.
expand	String	Returns the privileges associated with the role specified in the query filter. Include the following phrase in the query: `expand=privileges`

For example, to get details for the Business Manager role including privileges, you might use the following request:

```
/public/core/v3/roles?q=roleName=="Business Manager"&expand=privileges
```

GET response

If successful, returns the following information for each role:

Field	Type	Description
id	String	Role ID.
orgId	String	ID of the organization the role belongs to.
createdBy	String	User who created the role.

Field	Type	Description
updatedBy	String	User who last updated the role.
createTime	String	Date and time the role was created.
updateTime	String	Date and time the role was last updated.
roleName	String	Name of the role.
description	String	Description of the role.
displayName	String	Role name displayed in the user interface.
displayDescription	String	Description displayed in the user interface.
systemRole	Boolean	Whether the role is a system-defined role. Returns one of the following values: - True. Role is a system-defined role. - False. Role is a custom role.
status	String	Whether the organization's license to use the role is valid or has expired. Returns one of the following values: - Enabled - Disabled
privileges	Array	Privileges assigned to the role. Returned only when the URI includes `?expand=privileges` in a query.
id	String	Included in the privileges object. Privilege ID.
name	String	Included in the privileges object. Privilege name.
description	String	Included in the privileges object. Description of the privilege.
service	String	Included in the privileges object. The Informatica Intelligent Cloud Services service that uses the privilege.
status	String	Included in the privileges object. Whether the organization's license to use the privilege is valid or has expired. Returns one of the following values: - Enabled - Disabled

GET response example

You might receive a response similar to the following example:

```
[
    {
        "id": "7EjAMAHsiOTcg8v29z0Gsl",
        "orgId": "52ZSTB0IDK6dXxaEQLUaQu",
        "createdBy": "ops-post-deploy-user",
        "updatedBy": "ops-post-deploy-user",
        "createTime": "2019-03-22T21:26:46.000Z",
        "updateTime": "2019-03-22T21:26:52.000Z",
```

```
                "roleName": "Business Manager",
                "description": "Role used for business managers",
                "displayName": "Application Integration Business Manager",
                "displayDescription": "Role used for business managers",
                "systemRole": true,
                "status": "Disabled",
                "privileges": [
                    {
                        "id": "5Cgp0GcsmRejyxIgV4eXy1",
                        "name": "view.ai.console",
                        "description": "View application integration console",
                        "service": "ApplicationIntegration",
                        "status": "Disabled"
                    },
                    {
                        "id": "aReU2uciLYglcq0Ntvc2Ob",
                        "name": "view.ai.assets",
                        "description": "View  application integration assets",
                        "service": "ApplicationIntegration",
                        "status": "Disabled"
                    },
                    {
                        "id": "8zDel5v89cKfeMtM2FHFEw",
                        "name": "view.ai.designer",
                        "description": "View application integration designer",
                        "service": "ApplicationIntegration",
                        "status": "Disabled"
                    }
                ]
            }
        ]
```

Creating a role

You can create custom roles for your organization.

POST request

To create a custom role, send a POST request using the following URI:

```
/public/core/v3/roles
```

Note: The number of users, user groups, and roles combined cannot exceed 1000 for an organization.

Include the following information:

Field	Type	Required	Description
name	String	Yes	Name of the role.
description	String	-	Description of the role.
privileges	Array	Yes	IDs of the privileges to assign to the role. A role must have at least one privilege assigned to it.

POST response

If successful, returns the roles object with the details you included in the POST request.

POST example

To create a custom role, you might send a request similar to the following example:

```
POST <baseApiUrl>/public/core/v3/roles
Content-Type: application/json
Accept: application/json
INFA-SESSION-ID: <sessionId>
{
    "name" : "CAIviewer",
    "description": "A role to view Application Integration designer and assets",
    "privileges" : ["aQwUdcM8RcQewA1yWphZ4F", "0nTOXl8dzEwlSFoM0cO8gI"]
}
```

You might receive a response similar to the following example:

```
{
    "id": "8j2MPlr8ubZgteIOwleSCk",
    "orgId": "cPYWk02I4aBeuLEvYRtaMS",
    "createdBy": "a@abc.com",
    "updatedBy": "a@abc.com",
    "createTime": "2019-03-20T18:33:33.361Z",
    "updateTime": "2019-03-20T18:33:33.428Z",
    "roleName": "CAIviewer",
    "description": "A role to view Application Integration designer and assets",
    "displayName": "CAIviewer",
    "displayDescription": "A role to view Application Integration designer and assets",
    "systemRole": false,
    "status": "Enabled",
    "privileges": [
        {
            "id": "0nTOXl8dzEwlSFoM0cO8gI",
            "name": "view.ai.designer",
            "description": "View application integration designer"
        },
        {
            "id": "aQwUdcM8RcQewA1yWphZ4F",
            "name": "view.ai.assets",
            "description": "View  application integration assets"
        }
    ]
}
```

Updating a role

You can add or remove privileges for custom roles.

Add privileges

To add privileges to a custom role, send a PUT request using one of the following URIs:

```
/public/core/v3/roles/<role ID>/addPrivileges
```

```
/public/core/v3/roles/name/<role name>/addPrivileges
```

Include an array of privilege names for the custom role.

For example, to give a role create and delete privileges for data transfer tasks, you might use the following request:

```
PUT <baseApiUrl>/public/core/v3/roles/cPYWk02I4aBeuLEvYRtaMS/addPrivileges
Content-Type: application/json
Accept: application/json
INFA-SESSION-ID: <sessionId>
{
    "privileges" : ["create.data.transfer.task", "delete.data.transfer.task"]
}
```

You can use the privileges resource to get a list of privilege names that you can use. For more information, see "Privileges" on page 177.

Remove privileges

To remove privileges from a custom role, send a PUT request using one of the following URIs:

```
/public/core/v3/roles/<role ID>/removePrivileges

/public/core/v3/roles/name/<role name>/removePrivileges
```

You can't remove all of the privileges from a role. A role must have at least one privilege assigned to it.

Include an array of privilege names to remove for the custom role.

For example, to remove create and delete privileges for API collection from a role, you might use the following request:

```
PUT <baseApiUrl>/public/core/v3/roles/cPYWk053KnrUerLvYRtaMS/removePrivileges
Content-Type: application/json
Accept: application/json
INFA-SESSION-ID: <sessionId>
{
    "privileges" : ["create.api.collection", "delete.api.collection"]
}
```

Deleting a role

You can delete custom roles from your organization.

To delete a role, send a DELETE request using the following URI:

```
/public/core/v3/roles/<role ID>
```

SAML group and role mapping

If your organization uses SAML single sign-on (SSO) for authentication and authorization, use the Orgs resource to map SAML roles and groups to Informatica Intelligent Cloud Services roles. You can also use the Orgs resource to get details about SAML role and group mappings for an organization.

You can map up to 200 SAML groups or roles to an Informatica Intelligent Cloud Services role. To find a list of Informatica Intelligent Cloud Services roles, you can use the "Roles" on page 183 resource.

Before you map SAML groups and roles, ensure that the **Map SAML Groups and Roles** option is enabled in Administrator.

After you map SAML groups and roles to Informatica Intelligent Cloud Services roles, you cannot configure user roles or groups for SAML SSO users individually in Administrator or through the Informatica Intelligent Cloud Services REST API.

For more information about using SAML SSO, see *User Administration* in the Administrator help.

Adding SAML group mappings

Add SAML group mappings that map SAML groups to Informatica Intelligent Cloud Services roles.

Note: You cannot map a SAML group or role that contains one of the following characters in the name:

- Pipe (|)

- Semicolon (;)
- Comma (,)

PUT request

To add a mapping between an Informatica Intelligent Cloud Services role and SAML groups, use the following URI:

```
/public/core/v3/Orgs/<organization ID>/addSamlGroupMappings
```

If you map a SAML group that doesn't exist in Informatica Intelligent Cloud Services, Informatica Intelligent Cloud Services creates the group. If a group with the same name exists in Informatica Intelligent Cloud Services, you can specify whether to reuse the group or create a new group. By default, Informatica Intelligent Cloud Services creates a new group with _EXT appended to the name.

Include the following information:

Field	Type	Required	Description
groupMappings	Array	yes	Object that defines the Informatica Intelligent Cloud Services role and SAML group mappings.
roleName	String	yes	Include in the groupMappings object. Name of the Informatica Intelligent Cloud Services role.
samlGroupNames	String	yes	Include in the groupMappings object Names of the SAML groups to map to the Informatica Intelligent Cloud Services role.
reuseGroup	Boolean	--	Whether to use the existing Informatica Intelligent Cloud Services group if the group name is the same as the SAML group name. If false, Informatica Intelligent Cloud Services creates a new group. Default is false.

Returns the 204 response code if successful. Returns an error object if errors occurs.

PUT examples

To map the Informatica Intelligent Cloud Services Admin role to the Security and Dev-Managers SAML groups, and map the Informatica Intelligent Cloud Services Designer role to the Developers SAML group, you might send a request similar to the following example:

```
PUT <baseApiUrl>/public/core/v3/Orgs/6MRgiMIfvdRfCCLITrLCcI/addSamlGroupMappings
Accept: application/json
INFA-SESSION-ID: 9KA11tLGqxVcGeul8SQBK3
{
    "groupMappings": [
        {
            "roleName": "Admin",
            "samlGroupNames": ["Security","Dev-Managers"]
        },
        {
            "roleName": "Designer",
            "samlGroupNames": "Developers"
        }
    ],
    "reuseGroup": "true"

}
```

To add another group mapping to map the Informatica Intelligent Cloud Services Admin role to the Administrators SAML group, you might send a request similar to the following example:

```
PUT <baseApiUrl>/public/core/v3/Orgs/6MRgiMIfvdRfCCLITrLCcI/addSamlGroupMappings
Accept: application/json
INFA-SESSION-ID: 9KA11tLGqxVcGeul8SQBK3
{
    "groupMappings": [
        {
            "roleName": "Admin",
            "samlGroupNames": "Administrators"
        }
    ],
     "reuseGroup": "true"
}
```

Now the Informatica Intelligent Cloud Services Admin role is mapped to the Security, Dev-Managers, and Administrators SAML groups.

Adding SAML role mappings

Add SAML role mappings that map SAML roles to Informatica Intelligent Cloud Services roles.

Note: You cannot map a SAML group or role that contains one of the following characters in the name:

- Pipe (|)
- Semicolon (;)
- Comma (,)

PUT request

To add a mapping between Informatica Intelligent Cloud Services roles and SAML roles, use the following URI:

```
/public/core/v3/Orgs/<organization ID>/addSamlRoleMappings
```

Include the following information:

Field	Type	Required	Description
roleMappings	Array	yes	Object that defines the Informatica Intelligent Cloud Services role and SAML role mappings.
roleName	String	yes	Include in the roleMappings object. Name of the Informatica Intelligent Cloud Services role.
samlRoleNames	List	yes	Include in the roleMappings object Names of the SAML roles to map to the Informatica Intelligent Cloud Services role.

Returns the 204 response code if successful. Returns an error object if errors occurs.

PUT example

To map the Informatica Intelligent Cloud Services Developer role to the Data Designer and MS Deployer SAML roles, and map the Informatica Intelligent Cloud Services Admin role to the Operator SAML role, you might send a request similar to the following example:

```
PUT <baseApiUrl>/public/core/v3/Orgs/6MRgiMIfvdRfCCLITrLCcI/addSamlRoleMappings
Accept: application/json
INFA-SESSION-ID: 9KA11tLGqxVcGeul8SQBK3
```

```
{
    "roleMappings": [
        {
            "roleName": "Developer",
            "samlRoleNames": ["Data Designer","MS Deployer"]
        },
        {
            "roleName": "Admin",
            "samlRoleNames": "Operator"
        }
    ]
}
```

Removing SAML group mappings

Remove SAML group mappings that map SAML groups to Informatica Intelligent Cloud Services roles.

PUT request

To remove a SAML group mapping, send a PUT request using the following URI:

```
/public/core/v3/Orgs/<organization ID>/removeGroupMappings
```

Include the following information:

Field	Type	Required	Description
groupMappings	Array	yes	Object that defines the Informatica Intelligent Cloud Services role and SAML group mappings.
roleName	String	yes	Include in the groupMappings object. Name of the Informatica Intelligent Cloud Services role.
samlGroupNames	String	yes	Include in the groupMappings object. Names of the SAML groups to remove.

PUT example

If you want to remove the mapping between the Informatica Intelligent Cloud Services Admin role and the Administrators SAML group, you might send a request similar to the following example:

```
PUT <baseApiUrl>/public/core/v3/Orgs/6MRgiMIfvdRfCCLITrLCcI/removeSamlGroupMappings
Accept: application/json
INFA-SESSION-ID: 9KA11tLGqxVcGeul8SQBK3
{
    "groupMappings": [
        {
            "roleName": "Admin",
            "samlGroupNames": "Administrators"
        }
    ]
}
```

Removing SAML role mappings

Remove SAML role mappings that map SAML roles to Informatica Intelligent Cloud Services roles.

PUT request

To remove a SAML role mapping, send a PUT request using the following URI:

    /public/core/v3/Orgs/<organization ID>/removeSamlRoleMappings

Include the following information:

Field	Type	Required	Description
roleMappings	Array	yes	Object that defines the Informatica Intelligent Cloud Services role and SAML role mapping.
roleName	String	yes	Include in the roleMappings object. Name of the Informatica Intelligent Cloud Services role.
samlRoleNames	String	yes	Include in the roleMappings object. Names of the SAML roles to remove.

PUT example

If you want to remove the mapping between the Informatica Intelligent Cloud Services Admin role and the Operator SAML role, you might send a request similar to the following example:

```
PUT <baseApiUrl>/public/core/v3/Orgs/6MRgiMIfvdRfCCLITrLCcI/removeSamlRoleMappings
Accept: application/json
INFA-SESSION-ID: 9KA11tLGqxVcGeul8SQBK3
{
    "roleMappings": [
        {
            "roleName": "Admin",
            "samlRoleNames": "Operator"
        }
    ]
}
```

Getting SAML group mapping details

You can get the details for all the group mappings for an organization or use a filter to get group mapping details for a particular Informatica Intelligent Cloud Services role or SAML group.

GET request

To get details for group mappings, use the following URI:

    public/core/v3/Orgs/<organization ID>/SAMLConfig/groupMappings?<query parameters>

To get all the group mapping details for the organization, omit the query parameters.

To get details for a particular role or SAML group, you can include the following query parameters in the URI:

Parameter	Type	Description
q	String	Query filter string. You can filter using one of the following fields: - roleId - roleName - samlGroupNames
limit	Int	Maximum number of mappings to return. Default is 200.
skip	Int	Number of mappings to skip in the results. For example, a value of 4 excludes the first four mappings. Default is 0.

You can use the following fields to define the query filter:

Field	Type	Operators	Description
roleId	String	==	ID of the Informatica Intelligent Cloud Services role.
roleName	String	==	Name of the Informatica Intelligent Cloud Services role.
samlGroupNames	String	==	Name of the SAML group.

GET request example

To get the group mapping details for a particular Informatica Intelligent Cloud Services role ID, you might use the following request:

```
GET <baseApiUrl>/public/core/v3/Orgs/<organization ID>/SAMLConfig/groupMappings?
q=roleId=="94207429"
```

GET response

Returns the group mapping details. Returns an error if errors occur.

If the **Map SAML Groups and Roles** option is enabled but no mappings exist, the response is empty.

If successful, returns a count of the available group mappings and the following information for each group mapping:

Field	Type	Description
count	String	Number of group mappings available.
groupMappings	-	Contains mapping information for each group mapping.
roleId	String	ID of the Informatica Intelligent Cloud Services role.
roleName	String	Name of the Informatica Intelligent Cloud Services role.
samlGroupNames	String	Names of the SAML groups.

GET response example

If successful, you might receive a response similar to the following example:

```
{
    "count": 2,
    "groupMappings": [
        {
            "roleId":"94207429",
            "roleName:"Admin",
            "samlGroupNames": [
                "Security", "Dev-Managers"
            ]
        },
        {
            "roleId":"149021",
            "roleName:"Monitor",
            "samlGroupNames": "Developers"
        }
    ]
}
```

Getting SAML role mapping details

You can get the details for all the role mappings for an organization or use a filter to get role mapping details for a particular Informatica Intelligent Cloud Services role or SAML role.

GET request

To get details for role mappings, use the following URI:

```
public/core/v3/Orgs/<organization ID>/SAMLConfig/roleMappings?<query parameters>
```

To get all the role mapping details for the organization, omit the query parameters.

To get details for a particular role or SAML role, you can include the following query parameters in the URI:

Parameter	Type	Description
q	String	Query filter string. You can filter using one of the following fields: - roleId - roleName - samlRoleNames
limit	Int	Maximum number of mappings to return. Default is 200.
skip	Int	Number of mappings to skip in the results. For example, a value of 4 excludes the first four mappings. Default is 0.

You can use the following fields to define the query filter:

Field	Type	Operators	Description
roleId	String	==	ID of the Informatica Intelligent Cloud Services role.
roleName	String	==	Name of the Informatica Intelligent Cloud Services role.
samlRoleNames	String	==	Name of the SAML role.

GET request example

To get the role mapping details for a particular Informatica Intelligent Cloud Services role ID, you might use the following request:

```
GET <baseApiUrl>/public/core/v3/Orgs/03kEYNeduIJkVv0Qvq8NgY/SAMLConfig/roleMappings?
q=roleId=="94247429"
```

GET response

Returns the role mapping details. Returns an error if errors occur.

If the **Map SAML Groups and Roles** option is enabled but no mappings exist, the response is empty.

If successful, returns a count of the available role mappings and the following information for each role mapping:

Field	Type	Description
count	String	Number of role mappings available.
roleMappings	-	Contains mapping information for each role mapping.
roleId	String	ID of the Informatica Intelligent Cloud Services role.
roleName	String	Name of the Informatica Intelligent Cloud Services role.
samlRoleNames	String	Names of the SAML roles.

GET response example

If successful, you might receive a response similar to the following example:

```
{
    "count": 2,
    "roleMappings": [
        {
            "roleId":"94247429",
            "roleName":"Developer",
            "samlRoleNames": [
                "Data Designer", "MS Deployer"
            ]
        },
        {
            "roleId":"142321",
            "roleName":"Admin",
            "samlRoleNames": "Operator"
        }
    ]
}
```

Schedules

Use the schedule resource to request details of the schedules in the organization. You can also use this resource to create, update, or delete schedules.

You can use the following request methods:

- To get schedule details, use a GET request.
- To create a schedule, use a POST request.
- To update a schedule, use a PATCH request.
- To delete a schedule, use a DELETE request.

Note: To leverage full scheduling capabilities, use this resource instead of the REST API version 2 schedule resource.

Getting schedule details

Use the schedule resource to request details about the schedules in the organization.

GET request

To get the details of all schedules in the organization, use the following URI:

```
/public/core/v3/schedule
```

To get the details of a schedule using the schedule ID, use the following URI:

```
/public/core/v3/schedule/<id>
```

You can use a query parameter to get the details for specific schedules. For example, to get the details of all disabled schedules created by the user jdoe, you might use the following URI:

```
/public/core/v3/schedule?q=status=='Disabled' and createdBy=='jdoe'
```

You can use the following query parameters in the URI:

Parameter	Type	Description
status	Boolean	Status of the schedule. You can use the following operators: - == - !=
id	String	Schedule ID. Use the == operator.
scheduleFederatedId	String	Global unique identifier for the schedule. Use the == operator.
name	String	Schedule name. Use the == operator. If the schedule name includes a space, replace the space with `%20`

Parameter	Type	Description
updateTime	Date	Last time the schedule was updated, in UTC format. You can use the following operators: - < - <= - == - => - > - !=
updatedBy	String	User who updated the schedule. Use the == operator.
createdBy	String	User who created the schedule. Use the == operator.
interval	String	Interval or repeat frequency at which the schedule runs. You can use the following values: - None - Minutely - Hourly - Daily - Weekly - Biweekly - Monthly You can use the following operators: - == - !=

GET response

If successful, returns the schedules object for the requested schedule. If you request the details for all schedules, the schedules object contains details for each schedule in the organization.

Returns the error object if errors occur.

The schedules object includes the following attributes:

Field	Type	Description
id	String	Schedule ID.
scheduleFederatedId	String	Global unique identifier for the schedule.
name	String	Schedule name.
status	String	Status of the schedule. Returns one of the following values: - enabled - disabled
description	String	Description of the schedule.
createTime	Date/time	Time the schedule was created.
updateTime	Date/time	Last time the schedule was updated.

Field	Type	Description
createdBy	String	User who created the schedule.
updatedBy	String	User who last updated the schedule.
startTime	Date/time	Date and time when the schedule starts running, in UTC format.
endTime	Date/time	Date and time when the schedule stops running.
interval	String	Interval or repeat frequency at which the schedule runs tasks. Returns one of the following codes: - None. The schedule does not repeat. - Minutely. Tasks run on an interval based on the specified number of minutes, days, and time range. - Hourly. Tasks run on an hourly interval based on the start time of the schedule. - Daily. Tasks run on a daily interval based on the start time of the schedule. - Weekly. Tasks run on a weekly interval based on the start time of the schedule. - Biweekly. Tasks run every two weeks based on the start time of the schedule. - Monthly. Tasks run on a monthly interval based on the start time of the schedule.
frequency	Int	Frequency that the schedule runs for the specified interval. For example, if the interval is Hourly, a frequency of 2 means the task runs every 2 hours. Returned for Minutely, Hourly, and Daily intervals only.
rangeStartTime	Date/time	The start of the time range within a day that tasks run. Returned for Minutely and Hourly intervals only.
rangeEndTime	Date/time	The end of the time range within a day that tasks run. Returned for Minutely and Hourly intervals only.
sun	Boolean	Tasks run on Sunday. Returns one of the following codes: - true - false Returned for Minutely, Hourly, Weekly, and Biweekly intervals only.
mon	Boolean	Tasks run on Monday. See description for sun.
tue	Boolean	Tasks run on Tuesday. See description for sun.
wed	Boolean	Tasks run on Wednesday. See description for sun.
thu	Boolean	Tasks run on Thursday. See description for sun.
fri	Boolean	Tasks run on Friday. See description for sun.
sat	Boolean	Tasks run on Saturday. See description for sun.

Field	Type	Description
weekDay	Boolean	Tasks run on weekdays only. Returns one of the following codes: - true - false Returned for the Daily interval only.
dayOfMonth	Int	Date of the month that tasks run. Returns a date between 1-28. Returned for the Monthly interval only.
weekOfMonth	String	Week of the month that tasks run. Returns one of the following codes: - First. The tasks run in the first week of the month. - Second. The tasks run in the second week of the month. - Third. The tasks run in the third week of the month. - Fourth. The tasks run in the fourth week of the month. - Last. The tasks run in the last week of the month. Returned for the Monthly interval only.
dayOfWeek	String	Day of the week that tasks run. Returns one of the following codes: - Day. Tasks run on the first day or last day of the month, based on the selected weekOfMonth option. - Sunday - Monday - Tuesday - Wednesday - Thursday - Friday - Saturday Returned for the Monthly interval only.
timeZoneId	String	Time zone used for the day of the week, day of the month, or week of the month that the tasks run.

GET example

To request information about a schedule using the schedule ID, you might use the following request:

```
GET <baseApiUrl>/public/core/v3/schedule/0An1v84VPL3k6kypOlxq06D0000000000003
Accept: application/json
INFA-SESSION-ID: <sessionId>
```

A successful response might look like the following example:

```
{

"id": "0An1v84VPL3k6kypOlxq06D0000000000003",
"scheduleFederatedId" : "24bDtKg6d9SbaNlqDolHSR",
"name": "MI_FILE_LISTENER_10107",
"status":"enabled",
"createTime": "2018-12-03T17:34:45.000Z",
"updateTime": "2019-05-09T12:13:34.000Z",
"createdBy": "clouddemo",
"updatedBy": "vnath",
"startTime": "2020-06-09T00:15:55.000Z",
"interval": "Minutely",
"frequency": 5,
"rangeStartTime" : "",
"rangeEndTime" : "",
"mon": true,
"tue": true,
"wed": true,
"thu": true,
"fri": true,
```

```
"sat": true,
"sun": true,
"weekDay": false,
"dayOfMonth": 0,
"weekOfMonth": null,
"dayOfWeek": null,
"timeZoneId": "America/Los Angeles"

}
```

Creating a schedule

Use the schedule resource to create a schedule for the organization.

POST request

To create a schedule, use the following URI:

```
/public/core/v3/schedule
```

You can use the following fields in a schedules object:

Field	Type	Required	Description
name	String	Yes	Schedule name.
description	String	-	Description of the schedule.
status	String	-	Status of the schedule. Use one of the following values: - enabled - disabled Default is enabled.
startTime	Date/time	Yes	Date and time when the schedule starts running, in UTC format.
endTime	Date/time	-	Date and time when the schedule stops running. If you do not use this parameter, the schedule runs indefinitely.

Field	Type	Required	Description

interval	String	Yes	Interval or repeat frequency at which the schedule runs tasks. Use one of the following options:
			- None. Tasks run at the schedule start time. The schedule does not repeat.
			- Minutely. Tasks run on an interval based on the specified number of minutes, days, and time range. You can use the following parameters:
			  - frequency. Frequency in minutes that tasks run.
			  - sun, mon, tue, wed, thu, fri, sat. The days of the week that tasks run.
			  - startTimeRange and endTimeRange. The time range within a day tasks should run. Do not use if you want tasks to run all day.
			  - endTime. When the schedule should stop running. Do not use if you want the schedule to run indefinitely.
			- Hourly. Tasks run on an hourly interval based on the start time of the schedule. You can use the following parameters:
			  - frequency. Frequency in hours that tasks run.
			  - sun, mon, tue, wed, thu, fri, sat. The days of the week that tasks run.
			  - startTimeRange and endTimeRange. The time range within a day tasks should run. Do not use if you want tasks to run all day.
			  - endTime. When the schedule should stop running. Do not use if you want the schedule to run indefinitely.
			- Daily. Tasks run on a daily interval based on the start time configured for the schedule. You can use the following parameters:
			  - frequency. Frequency in days that tasks run.
			  - weekDay. Runs the tasks every weekday. Do not use if you want the tasks to run every day.
			  - endTime. When the schedule should stop running. Do not use if you want the schedule to run indefinitely.
			- Weekly. Tasks run on a weekly interval based on the start time of the schedule. You can use the following parameters:
			  - sun, mon, tue, wed, thu, fri, sat. The days of the week that tasks run.
			  - endTime. When the schedule should stop running. Do not use if you want the schedule to run indefinitely.
			- Biweekly. Tasks run every two weeks based on the start time of the schedule. You can use the following parameters:
			  - sun, mon, tue, wed, thu, fri, sat. The days of the week that tasks run.
			  - endTime. When the schedule should stop running. Do not use if you want the schedule to run indefinitely.
			- Monthly. Tasks run on a monthly interval based on the start time of the schedule. You can use the following parameters:
			  - dayOfMonth. Day of the month when you want tasks to run, between 1-28.
			  - dayOfWeek. Day of the week when you want tasks to run.
			  - weekOfMonth. Week of the month when you want tasks to run.
			  - endTime. When the schedule should stop running. Do not use if you want the schedule to run indefinitely.
			
			To indicate when tasks should run, use dayOfWeek with weekOfMonth, such as the First Monday. Or use dayOfMonth, such as 1.
			
			Tip: To run tasks on the last day of the month, use the Last weekOfMonth parameter with the Day dayOfWeek parameter.
			
			Default is no interval.

Field	Type	Required	Description
frequency	Int	Yes	Repeat frequency for tasks. Use one of the following values: - For Minutely intervals, use one of the following options: 5, 10, 15, 20, 30, 45. Default is 5. - For Hourly intervals, use one of the following options: 1, 2, 3, 4, 6, 8, 12. Default is 1. - For Daily intervals, use number of days between 1 -30. Default is 1. Use with Minutely, Hourly, and Daily intervals only.
rangeStartTime	Date/time	-	The start of the time range within a day that you want tasks to run. Enter a date and time using standard date/time format. Only the time portion is used. Use with Minutely, Hourly, and Daily intervals only.
rangeEndTime	Date/time	-	The end of the time range within a day that you want tasks to run. Enter a date and time using standard date/time format. Only the time portion is used. Use with Minutely, Hourly, and Daily intervals only.
sun	Boolean	-	Runs tasks on Sunday at the configured time. You can use the sun - sat parameters to run tasks on several days of the week. Use with Minutely, Hourly, Weekly, and Biweekly intervals only.
mon	Boolean	-	Runs tasks on Monday at the configured time. See description for sun.
tue	Boolean	-	Runs tasks on Tuesday at the configured time. See description for sun.
wed	Boolean	-	Runs tasks on Wednesday at the configured time. See description for sun.
thu	Boolean	-	Runs tasks on Thursday at the configured time. See description for sun.
fri	Boolean	-	Runs tasks on Friday at the configured time. See description for sun.
sat	Boolean	-	Runs tasks on Saturday at the configured time. See description for sun.
weekDay	Boolean	-	Runs tasks on weekdays. Use one of the following options: - True. Run tasks on Monday through Friday. Does not run tasks on the weekend. - False. Run tasks every day. Use with the Daily interval only.

Field	Type	Required	Description
dayOfMonth	Int	-	Date of the month that tasks should run. Use a date between 1-28. Use with the Monthly interval only. Tip: To run tasks on the last day of the month, use the Last weekOfMonth parameter with the Day dayOfWeek parameter.
weekOfMonth	String	-	Week of the month that tasks should run. Use with dayOfWeek to specify the day and week of the month that tasks should run. For example, the First Day or the Last Wednesday of the month. Use one of the following options: - First - Second - Third - Fourth - Last Use with the Monthly interval only.
dayOfWeek	String	-	Day of the week that tasks should run. Use with weekOfMonth to specify the day and week of the month that tasks should run. For example, the First Day or the Last Wednesday of the month. Use one of the following options: - Day - Sunday - Monday - Tuesday - Wednesday - Thursday - Friday - Saturday Use with the Monthly interval only.
timeZoneId	String	-	Time zone that the schedule uses for the dayOfMonth, weekOfMonth, and dayOfWeek fields. Default is UTC. For more information, see Appendix A, "Time zone codes" on page 543

POST request example

To create a schedule, your request might look something like the following example:

```
POST <baseApiUrl>/public/core/v3/schedule
Accept: application/json
INFA-SESSION-ID: <sessionId>
{
  "name": "my_schedule_1",
  "startTime" : "2023-09-18T22:00:00.000Z",
  "interval" : "Minutely",
  "frequency" : 5,
  "mon" : true,
  "wed" : true,
  "status" : "enabled"
}
```

POST response

Returns the schedules response object for the schedule that you created or updated.

Returns an error object if errors occur.

A successful response might look like the following example:

```
{
    "id": "8oKIw0ib9qMg1lGIWNPzkdD000000000000H",
    "createTime": "2023-09-10T11:48:28.000Z",
    "updateTime": "2023-09-10T11:48:28.000Z",
    "createdBy": "dev_scott",
    "updatedBy": "dev_scott",
    "name": "my_schedule_1",
    "rangeStartTime": null,
    "rangeEndTime": null,
    "status": "enabled",
    "frequency": 5,
    "description": null,
    "mon": true,
    "tue": false,
    "wed": true,
    "thu": false,
    "fri": false,
    "sat": false,
    "sun": false,
    "weekDay": false,
    "dayOfMonth": 0,
    "weekOfMonth": null,
    "dayOfWeek": null,
    "scheduleFederatedId": "1BrVocfYMAzeQHwXaaMWe7",
    "startTime": "2020-12-25T12:00:00.000Z",
    "endTime": null,
    "interval": "Minutely",
    "timeZoneId": "America/Los Angeles"
}
```

Updating a schedule

Use the schedule resource to update, enable, or disable a schedule.

PATCH request

Use the following URI:

```
/public/core/v3/schedule/<id>
```

Include the following fields in the request body:

Field	Type	Required	Description
name	String	-	Schedule name.
id	String	Yes	Schedule ID. You cannot update the schedule ID.
scheduleFederatedId	String	Yes	Global unique identifier for the schedule. You cannot update the federated schedule ID.
description	String	-	Description of the schedule.
status	String	-	Status of the schedule. Use one of the following values: - enabled - disabled Default is enabled.

Field	Type	Required	Description
startTime	Date/time	-	Date and time when the schedule starts running the tasks, in UTC format.
endTime	Date/time	-	Date and time when the schedule stops running the tasks. If you do not use this parameter, the schedule runs indefinitely.

Field	Type	Required	Description
interval	String	-	Interval or repeat frequency at which the schedule runs tasks. Use one of the following options: - None. Tasks run at the schedule start time. The schedule does not repeat. - Minutely. Tasks run on an interval based on the specified number of minutes, days, and time range. You can use the following parameters: - frequency. Frequency in minutes that tasks run. - sun, mon, tue, wed, thu, fri, sat. The days of the week that tasks run. - startTimeRange and endTimeRange. The time range within a day tasks should run. Do not use if you want tasks to run all day. - endTime. When the schedule should stop running. Do not use if you want the schedule to run indefinitely. - Hourly. Tasks run on an hourly interval based on the start time of the schedule. You can use the following parameters: - frequency. Frequency in hours that tasks run. - sun, mon, tue, wed, thu, fri, sat. The days of the week that tasks run. - startTimeRange and endTimeRange. The time range within a day tasks should run. Do not use if you want tasks to run all day. - endTime. When the schedule should stop running. Do not use if you want the schedule to run indefinitely. - Daily. Tasks run on a daily interval on the start time configured for the schedule. You can use the following parameters: - frequency. Frequency in days that tasks run. - weekDay. Runs the tasks every weekday. Do not use if you want the tasks to run every day. - endTime. When the schedule should stop running. Do not use if you want the schedule to run indefinitely. - Weekly. Tasks run on a weekly interval based on the start time of the schedule. You can use the following parameters: - sun, mon, tue, wed, thu, fri, sat. The days of the week that tasks run. - endTime. When the schedule should stop running. Do not use if you want the schedule to run indefinitely. - Biweekly. Tasks run every two weeks based on the start time of the schedule. You can use the following parameters: - sun, mon, tue, wed, thu, fri, sat. The days of the week that tasks run. - endTime. When the schedule should stop running. Do not use if you want the schedule to run indefinitely. - Monthly. Tasks run on a monthly interval based on the start time of the schedule. You can use the following parameters: - dayOfMonth. Day of the month when you want tasks to run, between 1-28. - dayOfWeek. Day of the week when you want tasks to run. - weekOfMonth. Week of the month when you want tasks to run. - endTime. When the schedule should stop running. Do not use if you want the schedule to run indefinitely. To indicate when tasks should run, use dayOfWeek with weekOfMonth, such as the First Monday. Or use dayOfMonth, such as 1. Tip: To run tasks on the last day of the month, use the Last weekOfMonth parameter with the Day dayOfWeek parameter. Default is no interval.

Field	Type	Required	Description

frequency	Int	-	Repeat frequency for tasks. Use one of the following values:
			
			- For Minutely intervals, use one of the following options: 5, 10, 15, 20, 30, 45.
			  Default is 5.
			- For Hourly intervals, use one of the following options: 1, 2, 3, 4, 6, 8, 12.
			  Default is 1.
			- For Daily intervals, use number of days between 1 -30.
			  Default is 1.
			
			Use with Minutely and Hourly intervals only.

rangeStartTime	Date/time	-	The start of the time range within a day that you want tasks to run. Enter a date and time using standard date/time format. Only the time portion is used. Use with Minutely and Hourly intervals only.
rangeEndTime	Date/time	-	The end of the time range within a day that you want tasks to run. Enter a date and time using standard date/time format. Only the time portion is used. Use with Minutely and Hourly intervals only.
sun	Boolean	-	Runs tasks on Sunday at the configured time. You can use the sun - sat parameters to run tasks on several days of the week. Use with Minutely, Hourly, Weekly, and Biweekly intervals only.
mon	Boolean	-	Runs tasks on Monday at the configured time. See description for sun.
tue	Boolean	-	Runs tasks on Tuesday at the configured time. See description for sun.
wed	Boolean	-	Runs tasks on Wednesday at the configured time. See description for sun.
thu	Boolean	-	Runs tasks on Thursday at the configured time. See description for sun.
fri	Boolean	-	Runs tasks on Friday at the configured time. See description for sun.
sat	Boolean	-	Runs tasks on Saturday at the configured time. See description for sun.
weekDay	Boolean	-	Runs tasks on weekdays. Use one of the following options: - True. Run tasks on Monday through Friday. Does not run tasks on the weekend. - False. Run tasks every day. Use with the Daily interval only.

Field	Type	Required	Description
dayOfMonth	Int	-	Date of the month that tasks should run. Use a date between 1-28. Use with the Monthly interval only. Tip: To run tasks on the last day of the month, use the Last weekOfMonth parameter with the Day dayOfWeek parameter.
weekOfMonth	String	-	Week of the month that tasks should run. Use with dayOfWeek to specify the day and week of the month that tasks should run. For example, the First Day or the Last Wednesday of the month. Use one of the following options: - First - Second - Third - Fourth - Last Use with the Monthly interval only.
dayOfWeek	String	-	Day of the week that tasks should run. Use with weekOfMonth to specify the day and week of the month that tasks should run. For example, the First Day or the Last Wednesday of the month. Use one of the following options: - Day - Sunday - Monday - Tuesday - Wednesday - Thursday - Friday - Saturday Use with the Monthly interval only.
timeZoneId	String	-	Time zone that the schedule uses for the dayOfMonth, weekOfMonth, and dayOfWeek fields. Default is UTC. For more information, see Appendix A, "Time zone codes" on page 543

PATCH example

To update a schedule, your request might look something like the following example:

```
PATCH <baseApiUrl>/public/core/v3/schedule/0An1v84VPL3k6kypOlxq06D0000000000003
Accept: application/json
INFA-SESSION-ID: <sessionId>
{
    "schedules": [
        {
            "id": "0An1v84VPL3k6kypOlxq06D0000000000003",
            "scheduleFederatedId": "1KiAwzRVIOTlAtCjPtzV4H",
            "name": "V3_Test_CreateSchedule_1569944878",
            "status": "disabled",
            "description": "Update version 2",
            "sat": true
        }
    ]
}
```

A successful response might look like the following example:

```
{
    "id": "0An1v84VPL3k6kypOlxq06D0000000000003",
```

```
                "createTime": "2022-09-24T15:34:36.000Z",
                "updateTime": "2022-10-01T15:47:59.442Z",
                "createdBy": "dev_larry",
                "updatedBy": "dev_larry",
                "name": "V3_Test_CreateSchedule_1569944878",
                "rangeStartTime": null,
                "rangeEndTime": null,
                "status": "disabled",
                "frequency": 1,
                "description": "Update version 2",
                "mon": false,
                "tue": false,
                "wed": true,
                "thu": false,
                "fri": false,
                "sat": true,
                "sun": false,
                "weekDay": false,
                "dayOfMonth": 0,
                "weekOfMonth": null,
                "dayOfWeek": null,
                "scheduleFederatedId": "1KiAwzRVIOTlAtCjPtzV4H",
                "startTime": "2022-12-25T12:00:00.000Z",
                "endTime": null,
                "interval": "Hourly",
                "timeZoneId": "America/Los Angeles"
        }
```

Deleting a schedule

Use the schedule resource to delete a schedule.

To delete a schedule, send a DELETE request using the following URI:

```
/public/core/v3/schedule/<id>
```

A successful response returns the 204 response code. If errors occur, an error object is returned.

Secure Agent services

Use the agentservice resource to stop or start a Secure Agent service.

After you send a POST request to start or stop a Secure Agent service, you can check the status of the service using the REST API V2 agent resource.

POST request

To stop or start a Secure Agent service, use the following URI:

```
public/core/v3/agent/service
```

Include the following fields in the request:

Field	Type	Required	Description
serviceName	String	Yes	Display name of the Secure Agent service to start or stop.
serviceAction	String	Yes	Action to perform on the Secure Agent service. Include one of the following values: - start. Start the latest version of the Secure Agent service. - stop. Stop all versions of the Secure Agent service.
agentId	String	Yes	The ID of the agent on which the Secure Agent service is located. To find the ID, send a lookup POST request that includes the agent path.

POST response

If the request is successful, the response includes one of the following states for the service:

State	Description
NEED_RUNNING	The start process has been initiated.
NEED_STOP	The stop process has been initiated.
DEPLOYING	The service is being provisioned.
DEPLOYED	The service is deployed and will be running soon.
UNKNOWN	Status is unknown. Check the status using the REST API version 2 agent resource. To find the status, send a lookup POST request that includes the agent path.
ERROR	The service is in a error state.
STARTING	The service is starting up.
RUNNING	The service is running and ready to accept jobs.
RESTARTING	The service is restarting and will be running soon.
STOPPING	The service is shutting down.
USER_STOPPED	The service has been stopped by a user.

POST request example

To start a Secure Agent service, you might send a request similar to the following example:

```
POST <baseApiUrl>/public/core/v3/agent/service
Content-Type: application/json
Accept: application/json
INFA-SESSION-ID: <sessionId>
{
    "serviceName":"Data Integration",
    "serviceAction":"start",
    "agentId":"kiphQJoRWWJfaC3enJ1smP"
}
```

POST response example

If a request is successful, you might receive a response similar to the following example:

```
{
    "serviceState":"STARTING",
    "message":"Successfully initiated start action. Note that only the latest version of
the service will be started. Send a GET request to  /v2/agent/details API to check the
updated status of the service."
}
```

Security logs

Use the securityLog resource to receive security log entries. Security logs include information about events such as login actions and creating, updating, and deleting users, user groups, and roles. To use this resource, you must be logged in with an administrator role.

GET request

To request entries for the last 24 hours with a maximum of 200 entries, use the following URI.

```
/public/core/v3/securityLog
```

Alternatively, you can use query parameters to specify which entries to return. For example, the following URI returns entries created on July 26, 2019 between 8:00AM and 5:00PM:

```
/public/core/v3/securityLog?
q=entryTime>="2019-07-26T08:00:00.000Z";entryTime<="2019-07-26T17:00:00.000Z"
```

You can include the following query parameters in the URI:

Parameter	Type	Description
entryTime	String	Start time or end time of the entry in UTC format. Use one of the following formats: - yyyy-MM-dd'T'HH:mm:ss'Z' - yyyy-MM-dd'T'HH:mm:ssZ - yyyy-MM-dd'T'HH:mm:ss.SSS'Z' - yyyy-MM-dd'T'HH:mm:ss.SSSZ The maximum date range is 14 days. You can use the following operators: - <= - => - > - == - != Default is to return entries for the last 24 hours with a maximum of 200.
actionCategory	String	Category of the security log entry. You can use the following operators: - == - != To use this query parameter, you must also include a valid time range using the entryTime query parameter.

Parameter	Type	Description
actor	String	User name who performed the action. You can use the following operators: - == - != To use this query parameter, you must also include a valid time range using the entryTime query parameter.
objectName	String	Name of the object acted upon. You can use the following operators: - == - != To use this query parameter, you must also include a valid time range using the entryTime query parameter.
skip	Int	Number of records to skip. To use this query parameter, you must also include a valid time range using the entryTime query parameter. Default is 0.
limit	Int	Number of entries to include in the response. You can specify a minimum of 100 and maximum of 1000. Default is 200.

GET response

Returns a securityLogEntry object for each security log entry returned. Returns the error object if errors occur.

The securityLogEntry object includes the following attributes:

Field	Type	Description
id	String	Security log entry ID.
orgId	String	Organization ID.
actor	String	User who performed the action.
entryTime	Timestamp	Time the action occurred.
objectId	String	ID of the object used.
objectName	String	Name of the object used.

Field	Type	Description
actionCategory	String	Category of security log entry. Returns one of the following codes: - Authentication - Organization - Sub-organization - User - Group - Role - Privilege - Agent - Privilege-Category - Preference
actionEvent	String	Type of action performed. Returns one of the following codes: - CREATE - UPDATE - DELETE - DISABLE - AGENT_LOGIN - USER_LOGIN - LOGOUT - PASSWORD_RESET

GET example

To view entries for the actions that the user "admin" performed on July 26, 2019 between 8:00AM and 5:00PM, you might use the following URI:

```
GET <baseApiUrl>/public/core/v3/securityLog?
q=entryTime>="2019-07-26T08:00:00.000Z";entryTime<="2019-07-26T17:00:00.000Z";actor=='adm
in'
Content-Type: application/json
Accept: application/json
INFA-SESSION-ID: <sessionId>
```

The response might look similar to the following example:

```
{
    "entries": [
        {
        "id": "1AoqT9lYsrUhu7kl49kGsx",
        "orgId": "9l10ywsSnqadMx1NtEEbKT",
        "actor": "admin",
        "entryTime": "2019-07-23T22:28:07.000Z",
        "objectId": "9l10ywsSnqadMx1NtEEbKT",
        "objectName": "idsv3_org_1563920884151",
        "actionCategory": "Organization",
        "actionEvent": "CREATE"
        },
        {
        "id": "595EZai5YqFi6X8GIpVVu0",
        "orgId": "9l10ywsSnqadMx1NtEEbKT",
        "actor": "admin",
        "entryTime": "2019-07-23T22:28:13.000Z",
        "objectId": "9pieratUfEWkhFHnzY1r49",
        "objectName": "idsv3_user_1563920884151",
        "actionCategory": "User",
        "actionEvent": "CREATE"
        }
    ]
}
```

Source control

You can use a Git source control repository to manage and track changes made to Informatica Intelligent Cloud Services objects such as projects, folders, and assets.

You can use the global Git repository for your organization. If the organization administrator enables project-level repositories, you can use repositories that are configured at the project level or use the global repository.

You can use the following resources:

- pull. Load objects to the Informatica Intelligent Cloud Services organization.
- pullByCommitHash. Retrieve objects that were modified by a particular commit and load them into your organization.
- checkout. Check objects out of the repository.
- undoCheckout. Undo the checkout of specified objects or undo the checkout of all objects that were included in a checkout operation.
- checkin. Check updated objects in to the repository.
- commit. Retrieve details about a commit from your repository.
- commitHistory. Receive commit history for a specific object or all of the objects in the organization.
- repositoryConnection. Get the repository connection ID for a project-level repository.
- sourceControlAction. Get the status of a source control operation.

Pulling objects

Use the pull resource to retrieve objects such as assets and projects from your global repository and load them into your organization.

When you pull a project, all of the source-controlled assets in the project and the project's folders are included in the pull. If the assets already exist in the target organization, their contents are overwritten to match the assets' state at the specified commit hash. If the project and the project's folders already exist in the target organization, they're reused.

Dependent objects that are located in other projects are not included in the pull. If dependent objects are missing in the pull request, the response includes an error message that lists the missing dependent objects.

If an asset's connections, runtime environments, or schedule exist in the target organization, they're reused. If you want an asset to use a different connection or runtime environment in the target organization, you can map the connection or runtime environment to a connection or runtime environment in your organization using the objectSpecification object.

For example, an asset uses the SecureAgent_dev runtime environment in the dev organization and uses the SecureAgent_test runtime environment in the test organization. You can use the objectSpecification object to map the SecureAgent_dev runtime environment to the SecureAgent_test runtime environment.

Note: Informatica recommends that you include less than 1000 objects in a pull request.

POST request

To load the latest version of objects from your repository to your organization, use the following URI:

```
/public/core/v3/pull
```

Note: You might receive a response to the POST request before the pull operation completes.

You can include the following fields in the request:

Field	Type	Required	Description
commitHash	String	Yes	Unique commit hash. The commit hash is validated during the operation. If you use a GitHub repository, you can include a partial hash in the request by sending the first 7 characters of the hash instead of the full string.
relaxObjectSpecificationValidation	Boolean	No	Whether the sources specified in the objectSpecification object must exist in the assets being pulled. Use one of the following values: - true. The objectSpecification objects are ignored if the sources don't exist in the assets that are included in the pull. - false. An error occurs when an objectSpecification source doesn't exist in the assets that are included in the pull. Default is false.
objects	List<Object>	Yes	Contains a list of all the objects to be pulled.
path	List<String>	Yes, if ID is not included	Include in the objects object. Full path of the object to be pulled.
id	String	Yes, if path is not included	Include in the objects object. ID of the object.

Field	Type	Required	Description
type	String	-	Include in the objects object. Type of asset to be pulled. If not specified, default is project. Can be one of the following types: - DTEMPLATE. Mapping. - MTT. Mapping task. - DSS. Synchronization task. - DMASK. Masking task. - DRS. Replication task. - MAPPLET. - BSERVICE. Business service definition. - HSCHEMA. Hierarchical schema. - PCS. PowerCenter task. - FWCONFIG. Fixed width configuration. - CUSTOMSOURCE. Saved query. - MI_TASK. Mass ingestion task. - WORKFLOW. Linear taskflow. - VISIOTEMPLATE - TASKFLOW
objectSpecification	List<Object>	-	Object specification for connection and runtime environments.
source	Object	Yes, if objectSpecification object is included and relaxObjectSpecificationValidation value is not true	Include in the objectSpecification object. Contains information about the source object.
path	List<String>	Yes, if objectSpecification object is included	Include in the source object. Full path of the connection or runtime environment in the repository.
type	String	Yes, if objectSpecification object is included	Include in the source object. Asset type. Use one of the following values: - Connection - AgentGroup

Field	Type	Required	Description
target	Object	Yes, if objectSpecification object is included	Include in the objectSpecification object. Contains information about the target object. Include path and type or include ID. If path, type, and ID are included, ID takes precedence.
path	List<String>	Yes, if ID is not included	Include in the target object. Path of the connection or runtime environment that is being mapped. Use with type.
type	String	Yes, if ID is not included	Include in the target object. Asset type. Use with path. Use one of the following values: - Connection - AgentGroup
id	String	Yes, if path and type are not included	Include in the target object. ID of the target object.

POST response

If successful, a POST request returns the following information:

Field	Type	Description
pullActionId	String	ID for the pull operation.
status	Object	Status of the pull operation.
state	String	Returned in the status object. Initial state of the pull operation. For a successful request, value will always be NOT_STARTED. To see the status after the operation begins, use the "Getting the status of a source control operation" on page 239 resource.
message	String	Returned in the status object Descriptive status message for the pull operation.

POST request examples for projects

You can request a pull operation for one or more assets or projects in a single POST request. To request a pull operation for multiple projects using the path field to specify the projects to pull, you might send a request that's similar to the following example:

```
POST <baseApiUrl>/public/core/v3/pull
Content-Type: application/json
Accept: application/json
INFA-SESSION-ID: <sessionId>

{
    "commitHash": "7c525831c247cf792f595d1663396d1ae2c85033",
     "objects": [
        {
            "path": ["Project2"]
        },
        {
            "path": ["Default"]
        }
    ]
}
```

To request a pull operation for the projects using project IDs, you might send a request that's similar to the following example:

```
POST <baseApiUrl>/public/core/v3/pull
Content-Type: application/json
Accept: application/json
INFA-SESSION-ID: <sessionId>

{
    "commitHash": "7c525831c247cf792f595d1663396d1ae2c85033",
     "objects": [
        {
            "id": "4gmWUVziA1qe7zXbyN1l6E"
        },
        {
            "id": "4TjbmrAGrk2eal3DOwdIk8"
        }
    ]
}
```

POST request examples for assets

You can request a pull operation for one or more assets in a POST request. To request a pull operation for an asset using the path field to specify the asset to pull, you might send a request that's similar to the following example:

```
POST <baseApiUrl>/public/core/v3/pull
Content-Type: application/json
Accept: application/json
INFA-SESSION-ID: <sessionId>

{
    "commitHash": "1013f61bf318758cccec08f2165f59bbbb41e8f0",
     "objects": [
        {
            "path": ["Default","Test_Mapping"],
            "type": "DTEMPLATE"
        }
    ]
}
```

To request a pull operation using the asset ID, you might send a request that's similar to the following example:

```
POST <baseApiUrl>/public/core/v3/pull
Content-Type: application/json
```

```
      Accept: application/json
      INFA-SESSION-ID: <sessionId>

      {
          "commitHash": "1013f61bf318758cccec08f2165f59bbbb41e8f0",
           "objects": [
               {
                    "id": "6wLjSK4tS4rdjKq5uGuC0T"
               }
           ]
      }
```

To request a pull operation using the asset ID and include the connections, you might send a request that's similar to the following example:

```
      POST <baseApiUrl>/public/core/v3/pull
      Content-Type: application/json
      Accept: application/json
      INFA-SESSION-ID: <sessionId>

      {

          "commitHash": "1013f61bf318758cccec08f2165f59bbbb41e8f0",
           "objects": [
               {
                    "id": "6wLjSK4tS4rdjKq5uGuC0T"
               }
           ],
           "objectSpecification":[
            {
               "source":
               {
                    "path":["ff"],
                    "type":"Connection"
               },
               "target":
               {
                    "path":["target_connection"],
                    "type":"Connection"
               }
            }
          ]
      }
```

To request a pull operation using the asset ID, the source runtime environment, and the target ID, you might send a request that's similar to the following example:

```
      POST <baseApiUrl>/public/core/v3/pull
      Content-Type: application/json
      Accept: application/json
      INFA-SESSION-ID: <sessionId>

      {

          "commitHash": "1013f61bf318758cccec08f2165f59bbbb41e8f0",
           "objects": [
               {
                    "id": "6wLjSK4tS4rdjKq5uGuC0T"
               }
           ],
           "objectSpecification":[
            {
               "source":
               {
                    "path":["USW1MJ02YNFB"],
                    "type":"AgentGroup"
               },
               "target":
               {
                    "id":"7UPtJVbrESTj0VkCBYAcUv"
```

```
            }
        }
      ]
    }
```

POST response example

For a successful POST request, the POST response might look like the following example:

```
{
    "pullActionId": "awRrziMMWXol7i42aTm1ih",
    "status": {
        "state": "NOT_STARTED",
        "message": "Initialized"
    }
}
```

Pulling objects in a commit

Use the pullByCommitHash resource to retrieve objects that were modified by a particular commit and load them into your organization.

When you use the pullByCommitHash resource, you include a commit hash in the request. The objects that changed in the commit are included in the pull. If you want to pull objects by path or object ID, use the pull resource instead.

If you use project-level repositories, include the repository connection ID in a pullByCommitHash request. You can find the repository connection ID for projects by sending a repositoryConnection request.

For more information about finding repository connection IDs, see "Getting repository connection details" on page 238.

If you want an asset to use a different connection or runtime environment, you can map the connection or runtime environment to a connection or runtime environment in your organization using the objectSpecification object.

Note: Informatica recommends that you include less than 1000 objects in a pull request.

POST request

To load the objects that changed in a particular commit from your repository to your organization, use the following URI:

```
/public/core/v3/pullByCommitHash
```

Note: You might receive a response to the POST request before the pull operation completes.

You can include the following fields in the request:

Field	Type	Required	Description
commitHash	String	Yes	Unique commit hash. The commit hash is validated during the operation. If you use a GitHub repository, you can include a partial hash by sending the first 7 characters of the commit hash. For other repositories, include the full 40-character commit hash.
searchCustomRepositories	Boolean	-	Whether to search project-level repositories if the commit hash wasn't found for the global repository.
repoConnectionId	String	-	Connection ID of the project-level repository to search.
relaxObjectSpecificationValidation	Boolean	-	Whether the sources specified in the objectSpecification object must exist in the assets being pulled. Use one of the following values: - true. The objectSpecification objects are ignored if the sources don't exist in the assets that are included in the pull. - false. An error occurs when an objectSpecification source doesn't exist in the assets that are included in the pull. Default is false.
objectSpecification	List<Object>	-	Object specification for connection and runtime environments.
source	Object	Yes, if objectSpecification object is included	Include in the objectSpecification object. Contains information about the source object.
path	List<String>	Yes, if objectSpecification object is included	Include in the source object. Name of the connection or runtime environment in the repository.
type	String	Yes, if objectSpecification object is included	Include in the source object. Asset type. Use one of the following values: - Connection - AgentGroup

Field	Type	Required	Description
target	Object	Yes, if objectSpecification object is included	Include in the objectSpecification object. Contains information about the target object. Include path and type or include ID. If path, type, and ID are included, ID takes precedence.
path	List<String>	Yes, if ID is not included	Include in the target object. Name of the connection or runtime environment. Use with type.
type	String	Yes, if ID is not included	Include in the target object. Asset type. Use with path. Use one of the following values: - Connection - AgentGroup
id	String	Yes, if path and type are not included	Include in the target object. ID of the target object.

POST response

If successful, a POST request returns the following information:

Field	Type	Description
pullActionId	String	ID for the pull operation.
status	Object	Status of the pull operation.
state	String	Returned in the status object. Initial state of the pull operation. For a successful request, value will always be NOT_STARTED. To see the status after the operation begins, use the "Getting the status of a source control operation" on page 239 resource.
message	String	Returned in the status object. Descriptive status message for the pull operation.

POST request examples

To request a pull operation for all changed assets in a commit, you might send a request that's similar to the following example:

```
POST <baseApiUrl>/public/core/v3/pullByCommitHash
Content-Type: application/json
Accept: application/json
INFA-SESSION-ID: <sessionId>

{
```

```
        "commitHash": "7c525831c247cf792f595d1663396d1ae2c85033"
    }
```

To request a pull operation for all changed assets in a commit made to a project-level repository, you might send a request that's similar to the following example:

```
POST <baseApiUrl>/public/core/v3/pullByCommitHash
Content-Type: application/json
Accept: application/json
INFA-SESSION-ID: <sessionId>

{
    "commitHash": "7c525831c247cf792f595d1663396d1ae2c85033",
    "searchCustomRepositories": true
}
```

To request a pull operation for all changed assets in a commit and include connections and runtime environments, you might send a request that's similar to the following example:

```
POST <baseApiUrl>/public/core/v3/pullByCommitHash
Content-Type: application/json
Accept: application/json
INFA-SESSION-ID: <sessionId>

{

    "commitHash": "1013f61bf318758cccec08f2165f59bbbb41e8f0",
    "objectSpecification":[
     {
        "source":
        {
            "path":["ff"],
            "type":"Connection"
        },
        "target":
        {
            "path":["target_connection"],
            "type":"Connection"
        }
     },
     {
        "source": {
            "path": ["USW1MJ02YNFB"],
            "type": "AgentGroup"
        },
        "target": {
            "id": "2ga6h3hRNZCf9Br0ZWB7EF"
        }
     }
    ]
}
```

POST response example

For a successful POST request, the response might look like the following example:

```
{
    "pullActionId": "iW5TmGqUjmUcdZKk4c4VQH",
    "status": {
        "state": "NOT_STARTED",
        "message": "Initialized"
    }
}
```

Checking out objects

Use the checkout resource to check out a source-controlled object so that you can make changes to it. When you check out an object, the object is locked so that other users can't make changes to it.

You can check out multiple projects, folders, or assets in one request.

If multiple objects are included in the checkout and the checkout fails for any of them, none of the objects will be checked out. The objects that would have been successful will have a status of CANCELLED.

For more information about the checkout status, see "Getting the status of a source control operation" on page 239.

POST request

To check objects out of the repository, use the following URI:

```
/public/core/v3/checkout
```

In the request, you must provide either the object ID or the full path and object type.

You can include the following fields in the request:

Field	Type	Required	Description
objects	List<Object>	Yes	Contains a list of all the objects to be checked out.
id	String	Yes, if path and type are not included	Include in the objects object. ID of the object.
path	List<String>	Yes, if ID is not included	Include in the objects object. Full path of the object to be checked out.

Field	Type	Required	Description
type	String	Yes, if ID is not included	Include in the objects object. Type of asset to be checked out. Can be one of the following types: - DTEMPLATE. Mapping. - MTT. Mapping task. - DSS. Synchronization task. - DMASK. Masking task. - DRS. Replication task. - MAPPLET. - BSERVICE. Business service definition. - HSCHEMA. Hierarchical schema. - PCS. PowerCenter task. - FWCONFIG. Fixed width configuration. - CUSTOMSOURCE. Saved query. - MI_TASK. Mass ingestion task. - WORKFLOW. Linear taskflow. - VISIOTEMPLATE - TASKFLOW - PROCESS. Application Integration process. - AI_CONNECTION. Application Integration app connection. - AI_SERVICE_CONNECTOR. Application Integration service connector. - GUIDE. Application Integration guide. - PROCESS_OBJECT. Application Integration process object. - HUMAN_TASK. Application Integration human task.
includeContainerAssets	Boolean	-	Include in the objects object. Whether all objects in a project or folder are included in the check-in. Use one of the following values: - true. Include all objects in the project or folder. - false. Do not include objects in the project or folder. Default is false.

POST response

If successful, a POST request returns the following information:

Field	Type	Description
Id	String	ID for the checkout operation.
status	Object	Status of the checkout operation.
state	String	Returned in the status object. Initial state of the checkout operation. For a successful request, value will always be NOT_STARTED. To see the status after the operation begins, use the "Getting the status of a source control operation" on page 239 resource.
message	String	Returned in the status object Descriptive status message for the checkout operation.

POST request examples

To request a checkout operation for a project and include all of the assets in the project, you might send a request that's similar to the following example:

```
POST <baseApiUrl>/public/core/v3/checkout
Content-Type: application/json
Accept: application/json
INFA-SESSION-ID: <sessionId>

{
  "objects": [
    {
      "id": "ejZY66c19YUccBdbGwKG4P",
      "includeContainerAssets": true
    }
  ]
}
```

To request a checkout operation for a project and include two of the assets in the project, you might send a request that's similar to the following example:

```
POST <baseApiUrl>/public/core/v3/checkout
Content-Type: application/json
Accept: application/json
INFA-SESSION-ID: <sessionId>

{
  "objects": [
    {
      "id": "iIVBNZSpUKFg4N6g2PKUox",
      "includeContainerAssets": false
    },
    {
      "id": "l7bgB85m5oGiXObDxwnvK9"
    },
    {
      "id": "1MW0GDAE1sFgnvWkvom7mK"
    }
  ]
}
```

To request a checkout operation for an asset named Test_Mapping that's in the Default project, you might send a request that's similar to the following example:

```
POST <baseApiUrl>/public/core/v3/checkout
Content-Type: application/json
Accept: application/json
INFA-SESSION-ID: <sessionId>

{
  "objects": [
    {
      "path": [
        "Default",
        "Test_Mapping"
      ],
      "type": "DTEMPLATE"
    }
  ]
}
```

To request a checkout operation using the asset ID, you might send a request that's similar to the following example:

```
POST <baseApiUrl>/public/core/v3/checkout
Content-Type: application/json
Accept: application/json
INFA-SESSION-ID: <sessionId>
```

```
{
  "objects": [
    {
      "id": "3iWWHkLbM2giVppBmJmZgV"
    }
  ]
}
```

POST response example

You might receive a response similar to the following example:

```
{
    "id": "awRrziMMWXol7i42aTm1ih",
    "status": {
        "state": "NOT_STARTED",
        "message": "Initialized"
    }
}
```

Undoing a checkout

Use the undoCheckout resource to undo a checkout. The object will revert to the last version that was pulled.

You can undo the checkout of an asset, project, or folder. You can also undo a checkout for all objects that were included in a checkout. If you are logged in as an administrator, you can undo a checkout performed by another user.

POST request

To undo a checkout, use the following URI:

```
/public/core/v3/undoCheckout
```

In the request, include the ID or the path and type for each object that you want to perform the undo checkout operation. Or, include a checkout operation ID to undo the checkout for all objects that were included in a checkout operation.

Note: Source control logs are retained for seven days. If the checkout occurred more than seven days ago, the checkout operation ID no longer exists so it can't be used in an undo checkout operation.

If you want an asset to use a different connection or runtime environment, you can map the connection or runtime environment to a connection or runtime environment in your organization using the objectSpecification object.

You can include the following fields in the request:

Field	Type	Required	Description
checkoutOperationId	String	-	Checkout operation ID. If specified, all assets that were included in the checkout operation will revert to their original state.
objects	List<Object>	-	Contains a list of all the objects to be included in the undo checkout operation.
id	String	Yes, if the object's path and type are not included and checkoutOperation ID is not included	Include in the objects object. ID of the object.

Field	Type	Required	Description
path	List<String>	Yes, if the object's ID is not included and checkoutOperation ID is not included	Include in the objects object. Full path of the object.
type	String	Yes, if the object's ID is not included and checkoutOperation ID is not included	Include in the objects object. Type of asset. Can be one of the following types: - DTEMPLATE. Mapping. - MTT. Mapping task. - DSS. Synchronization task. - DMASK. Masking task. - DRS. Replication task. - MAPPLET. - BSERVICE. Business service definition. - HSCHEMA. Hierarchical schema. - PCS. PowerCenter task. - FWCONFIG. Fixed width configuration. - CUSTOMSOURCE. Saved query. - MI_TASK. Mass ingestion task. - WORKFLOW. Linear taskflow. - VISIOTEMPLATE - TASKFLOW - PROCESS. Application Integration process. - AI_CONNECTION. Application Integration app connection. - AI_SERVICE_CONNECTOR. Application Integration service connector. - GUIDE. Application Integration guide. - PROCESS_OBJECT. Application Integration process object. - HUMAN_TASK. Application Integration human task.
includeContainerAssets	Boolean	-	Include in the objects object. Applicable to projects and folders. Whether all objects in a project or folder are included in the undo checkout operation. Use one of the following values: - true. Include all objects in the project or folder. - false. Do not include objects in the project or folder. Default is false.
objectSpecification	List<Object>	-	Object specification for connection and runtime environments.
source	Object	Yes, if objectSpecification object is included	Include in the objectSpecification object. Contains information about the source object.
path	List<String>	Yes, if objectSpecification object is included	Include in the source object. Full path of the connection or runtime environment in the repository.

Field	Type	Required	Description
type	String	Yes, if objectSpecification object is included	Include in the source object. Asset type. Use one of the following values: - Connection - AgentGroup
target	Object	Yes, if objectSpecification object is included	Include in the objectSpecification object. Contains information about the target object. Include path and type or include ID. If path, type, and ID are included, ID takes precedence.
path	List<String>	Yes, if ID is not included	Include in the target object. Path of the connection or runtime environment. Use with type.
type	String	Yes, if ID is not included	Include in the target object. Asset type. Use with path. Use one of the following values: - Connection - AgentGroup
id	String	Yes, if path and type are not included	Include in the target object. ID of the target object.

POST response

If successful, a POST request returns the following information:

Field	Type	Description
id	String	ID for the undo checkout operation.
status	Object	Status of the undo checkout operation.
state	String	Returned in the status object. Initial state of the undo checkout operation. For a successful request, value will always be NOT_STARTED. To see the status after the operation begins, use the "Getting the status of a source control operation" on page 239 resource.
message	String	Returned in the status object. Descriptive status message for the undo checkout operation.

POST request examples

To request an undoCheckout operation using asset IDs, you might send a request that's similar to the following example:

```
POST <baseApiUrl>/public/core/v3/undoCheckout
Content-Type: application/json
Accept: application/json
INFA-SESSION-ID: <sessionId>
```

```
{
     "objects": [
         {
             "id": "4gmWUVziA1qe7zXbyN1l6E"
         },
         {
             "id": "4TjbmrAGrk2eal3DOwdIk8"
         }
     ]
}
```

To request an undo checkout operation for assets by path, you might send a request that's similar to the following example:

```
POST <baseApiUrl>/public/core/v3/undoCheckout
Content-Type: application/json
Accept: application/json
INFA-SESSION-ID: <sessionId>

{
     "objects": [
         {
             "path": ["Default","Test_Mapping1"],
             "type": "DTEMPLATE"
         },
         {
             "path": ["Default","Test_Mapping2"],
             "type": "DTEMPLATE"
         }
     ]
}
```

To request an undoCheckout operation for an asset and map the source connection and runtime environment to the target connection and runtime environment, you might send a request that's similar to the following example:

```
POST <baseApiUrl>/public/core/v3/undoCheckout
Content-Type: application/json
Accept: application/json
INFA-SESSION-ID: <sessionId>

{
     "objects": [
         {
             "id": "6wLjSK4tS4rdjKq5uGuC0T"
         }
     ],
     "objectSpecification":[
      {
         "source":
         {
             "path":["ff"],
             "type":"Connection"
         },
         "target":
         {
             "path":["target_connection"],
             "type":"Connection"
         }
      },
      {
         "source": {
             "path": ["USW1MJ02YNFB"],
             "type": "AgentGroup"
         },
         "target": {
             "id": "2ga6h3hRNZCf9Br0ZWB7EF"
         }
```

```
        }
      ]
    }
```

To request an undoCheckout operation using the checkout operation ID, you might send a request that's similar to the following example:

```
POST <baseApiUrl>/public/core/v3/undoCheckout
Content-Type: application/json
Accept: application/json
INFA-SESSION-ID: <sessionId>

{
    "checkoutOperationId": "<checkout operation ID>"
}
```

POST response example

For any of the request examples, you might receive a response similar to the following example:

```
{
    "id": "awRrziMMWXol7i42aTm1ih",
    "status": {
        "state": "NOT_STARTED",
        "message": "Initialized"
    }
}
```

Checking in objects

Use the checkin resource to check updated objects in to the repository.

POST request

To check objects in to the repository, use the following URI:

```
/public/core/v3/checkin
```

In the request, you must provide either the object ID or the full path and object type.

You can include the following fields in the request:

Field	Type	Required	Description
objects	List<Object>	Yes	Contains a list of all the objects to be checked in. You can check in a single asset or check in any number of projects or folders.
id	String	Yes, if path and type are not included	Include in the objects object. ID of the object.
path	List<String>	Yes, if ID is not included	Include in the objects object. Full path of the object to be checked in.

Field	Type	Required	Description
type	String	Yes, if ID is not included	Include in the objects object. Type of asset to be checked in. Can be one of the following types: - DTEMPLATE. Mapping. - MTT. Mapping task. - DSS. Synchronization task. - DMASK. Masking task. - DRS. Replication task. - MAPPLET. - BSERVICE. Business service definition. - HSCHEMA. Hierarchical schema. - PCS. PowerCenter task. - FWCONFIG. Fixed width configuration. - CUSTOMSOURCE. Saved query. - MI_TASK. Mass ingestion task. - WORKFLOW. Linear taskflow. - VISIOTEMPLATE - TASKFLOW - PROCESS. Application Integration process. - AI_CONNECTION. Application Integration app connection. - AI_SERVICE_CONNECTOR. Application Integration service connector. - GUIDE. Application Integration guide. - PROCESS_OBJECT. Application Integration process object. - HUMAN_TASK. Application Integration human task.
includeContainerAssets	Boolean	-	Include in the objects object. Whether all objects in a project or folder are included in the check-in. Use one of the following values: - true. Include all objects in the project or folder. - false. Do not include objects in the project or folder. Default is false.
summary	String	Yes	Summary of the check-in. Maximum length is 255 characters.
description	String	-	Description of the check-in.

POST response

If successful, a POST request returns the following information:

Field	Type	Description
Id	String	ID for the check-in operation.
status	Object	Status of the check-in operation.

Field	Type	Description
state	String	Returned in the status object. Initial state of the check-in operation. For a successful request, value will always be NOT_STARTED. To see the status after the operation begins, use the sourceControlAction resource.
message	String	Returned in the status object Descriptive status message for the check-in operation.

POST request examples for projects

You can request a check-in operation for one or more projects in a single POST request. To request a check-in operation for a project and include all of the assets in the project, you might send a request that's similar to the following example:

```
POST <baseApiUrl>/public/core/v3/checkin
Content-Type: application/json
Accept: application/json
INFA-SESSION-ID: <sessionId>

{
  "objects": [
    {
      "id": "3iWWHkLbM2giVppBmJmZgV",
      "includeContainerAssets": true
    }
  ],
  "summary": "Revised mappings",
  "description": "Revised m_custArch and m_custNew"
}
```

To request a check-in operation for an asset using the path and object type, you might send a request that's similar to the following example:

```
POST <baseApiUrl>/public/core/v3/checkin
Content-Type: application/json
Accept: application/json
INFA-SESSION-ID: <sessionId>

{
  "objects": [
    {
      "path": [
        "Default",
        "Test_Mapping"
      ],
      "type": "DTEMPLATE"
    }
  ]
}
```

POST request examples for assets

You can request a check-in operation for one asset in a POST request. To request a check-in operation for an asset named Test_Mapping that's in the Default project, you might send a request that's similar to the following example:

```
POST <baseApiUrl>/public/core/v3/checkin
Content-Type: application/json
Accept: application/json
INFA-SESSION-ID: <sessionId>

{
```

```
      "objects": [
        {
          "path": [
            "Default",
            "Test_Mapping"
          ],
          "type": "DTEMPLATE"
        }
      ]
  }
```

To request a check-in operation using the asset ID, you might send a request that's similar to the following example:

```
POST <baseApiUrl>/public/core/v3/checkin
Content-Type: application/json
Accept: application/json
INFA-SESSION-ID: <sessionId>

{
  "objects": [
    {
      "id": "3iWWHkLbM2giVppBmJmZgV"
    }
  ],
  "summary": "Revised Revised m_custArch"
}
```

POST response example

A POST response might look like the following example:

```
{
    "Id": "awRrziMMWXol7i42aTm1ih",
    "status": {
        "state": "NOT_STARTED",
        "message": "Initialized"
    }
}
```

Getting commit details

Use the commit resource to retrieve details about a commit from your repository.

GET request

To get the details for a commit, use the following URI:

```
/public/core/v3/commit
```

You can include the following fields in the request:

Field	Type	Required	Description
commitHash	String	Yes	The commit hash that you're requesting details for.
searchCustomRepositories	Boolean	-	Whether to search project-level repositories if the commit hash wasn't found for the global repository.
repoConnectionId	String	-	Connection ID of the project-level repository to search. To find the repository connection ID, you can send a repositoryConnection request. For more information, see <link to the new Getting repository connection details topic>

GET response

If successful, a GET request returns the following information:

Field	Type	Description
hash	String	Unique identifier for the commit. You can use this value in a pullByCommitHash request.
summary	String	Summary associated with the commit hash.
description	String	Detailed description associated with the commit hash.
username	String	Name of the user who performed the commit.
email	String	Email address of the user who performed the commit.
date	Date	The timestamp when the commit was submitted in the following format: `yyyy-MM-dd'T'HH:mm:ss.SSSZ`
committer	String	The Git user who filed the commit.
changes	List <Object>	Contains change details for each object included in the commit.
id	String	Included in the changes object. Global unique identifier for the object. Included if the object currently exists in your organization.
appContextId	String	Included in the changes object. ID of the object in context, used in REST API version 2 calls. Included if the object currently exists in your organization.
name	String	Included in the changes object. Name of the asset, project, or folder.
type	String	Included in the changes object. Type of object.
path	List <String>	Included in the changes object. Full path of the object.
oldPath	List <String>	Included in the changes object. Remote repository path for the object before the commit. Included when the object was moved or renamed.
action	String	Included in the changes object. Indicates the type of changes made to the object during the commit. Includes one of the following values: - ADDED - DELETED - MODIFIED - MOVED

GET response example

A GET response might look like the following example:

```
{
    "hash": "aca30f6c44de9bef23db59ed16967653481c5e23",
    "summary": "Update. ",
    "description": "Restructured and updated.",
    "username": "testuser",
    "email": "user@gmail.com",
    "date": "2019-05-13T17:50:46.000Z"
    "committer": "Alexander Freeman",
    "changes": [
        {
            "id": "5G15DJ19Fw3j20XGL6oooL",
            "appContextId": "N0A1700000000001J",
            "name": "Mapping1",
            "type": "DTEMPLATE",
            "path": ["Versioned_Project2","Mapping1"],
            "action": "MODIFIED"
        },
        {
            "id": "2jFxl7stzKgi3lKAtVpWi3",
            "appContextId": "N0A17000000003A02",
            "name": "Versioned_Project",
            "type": "Project",
            "path": ["Versioned_Project"],
            "action": "ADDED"
        },
        {
            "id": "k3SYVglqhAafSWHuQuZdm1",
            "appContextId": "N0F17000000000291",
            "name": "Versioned_Folder",
            "type": "Folder",
            "path": ["Versioned_Project", "Versioned_Folder"],
            "oldPath": ["Versioned_Project", "Test_Folder"],
            "action": "MOVED"
        },
        {
            "id": null,
            "appContextId": null,
            "name": "Mapping2",
            "type": "DTEMPLATE",
            "path": ["Versioned_Project2","Mapping2"],
            "action": "DELETED"
        }
    ]
}
```

Getting commit history

Use the commitHistory resource to get commit history for source-controlled objects in your organization.

You can request commit history for all source-controlled objects, specific projects or folders, or specific assets. You can also use query parameters to request commit history for all objects on a particular Explore page or a specified number of pages.

GET request

You can request the commit history for all of your organization's projects and assets or request the history for a particular project or asset.

To get commit history, use the following URI:

```
/public/core/v3/commitHistory?<query parameters>
```

To get the commit history for all objects in the organization, omit the query parameters.

To get the commit history for a particular project or asset, you can include the following query parameters in the URI:

Parameter	Type	Description
q	String	Query filter string. Include an object ID, project or folder name, or asset type.
perPage	String	Number of entries to include per page. Maximum of entries is 100. Default is 100.
page	String	Show a specific page of results. Default is page 1.

You can use the following fields to define the query filter:

Field	Type	Operators	Description
id	String	==	ID of the project, folder, or asset.
path	String	==	Project or path where the assets are located.
type	String	==	Asset type. Required to receive commit history for an asset. Can be one of the following types: - DTEMPLATE. Mapping. - MTT. Mapping task. - DSS. Synchronization task. - DMASK. Masking task. - DRS. Replication task. - MAPPLET. - BSERVICE. Business service definition. - HSCHEMA. Hierarchical schema. - PCS. PowerCenter task. - FWCONFIG. Fixed width configuration. - CUSTOMSOURCE. Saved query. - MI_TASK. Mass ingestion task. - WORKFLOW. Linear taskflow. - VISIOTEMPLATE - TASKFLOW - PROCESS. Application Integration process. - AI_CONNECTION. Application Integration app connection. - AI_SERVICE_CONNECTOR. Application Integration service connector. - GUIDE. Application Integration guide. - PROCESS_OBJECT. Application Integration process object. - HUMAN_TASK. Application Integration human task.

Query examples

The following examples show how you can use query parameters to get commit history for certain objects.

- Commit history for a single asset or folder using IDs:

      GET /public/core/v3/commitHistory?q=id=='23546'

- Commit history for two projects using IDs:

      GET /public/core/v3/commitHistory?q=id=='23423' and id=='5645esf'

- Commit history for two projects using paths:

      GET /public/core/v3/commitHistory?q=path=='project name 1' and path=='project name 2'

- Commit history for an asset using the path:

```
GET /public/core/v3/commitHistory?q=path=='ProjectName/FolderName/AssetName1' and
type=='DTEMPLATE'
```

GET response

Returns a list of commits with the latest commit listed first. Returns an error if errors occur.

If successful, returns the following information for each commit in the commits object:

Field	Type	Description
commits	List<Object>	List of all of the commits.
hash	String	Unique identifier for the commit.
summary	String	Summary associated with the commit hash.
description	String	Detailed description associated with the commit hash.
username	String	Name of the user who performed the commit.
email	String	Email address of the user who performed the commit.
date	Date	The timestamp when the commit was submitted in the following format: `yyyy-MM-dd'T'HH:mm:ss.SSSZ`
committer	String	The Git user who filed the commit.
pagination	Object	Page information.
pageSize	Int	Number of elements on the page.
currentPage	Int	Current page number.
hasNext	Boolean	Whether there is another page after the current page.

Get response example

If successful, you might receive a response similar to the following example:

```
{
    "commits": [
        {
            "hash": "b0bdc63a7fb9047db6c3bc29ad67d5ecbf7d1d47",
            "summary": "Default Project v1",
            "description": "Default Project v1",
            "username": "testuser@gmail.com",
            "email": "testuser@gmail.com",
            "date": "2020-03-30T22:30:19.000Z",
            "committer": "testuserinfa"
        },
        {
            "hash": "fc6fcc318ad1b4aec17017d053bc2f0d1f605096",
            "summary": "Synchronization Task1 - Copy 1 v1",
            "description": "Synchronization Task1 - Copy 1 v1",
            "username": "testuser@gmail.com",
            "email": "testuser@gmail.com",
            "date": "2020-03-30T22:22:02.000Z",
            "committer": "testuserinfa"
        },
        {
```

```
                    "hash": "74d776c574dad3bc5cf7a44b22195cf423560fe9",
                    "summary": "Project2 Folder1 v1",
                    "description": null,
                    "username": "testuser@gmail.com",
                    "email": "testuser@gmail.com",
                    "date": "2020-03-30T22:17:48.000Z",
                    "committer": "testuserinfa"
                },
                {
                    "hash": "02c8b5950df4ef2110288ba7f77a220bc6f05b0a",
                    "summary": "v1 of project2",
                    "description": "v1 of project2",
                    "username": "testuser@gmail.com",
                    "email": "testuser@gmail.com",
                    "date": "2020-03-24T23:41:52.000Z",
                    "committer": "testuserinfa"
                },
            ],
            "pagination": null
        }
```

Getting repository connection details

Use the repositoryConnection resource to get repository details such as the repository connection ID for a given project ID or project name. You can get the repository details for one or more projects in one request.

Use the repository connection ID that's included in the response to send a pullByCommitHash request for objects in project-level repositories.

GET request

To get details about the repository connection for a given project, use the following URL:

```
/public/core/v3/repositoryConnection
```

You can include the following fields in the request:

Field	Type	Required	Description
projectIds	List of String	-	Project ID that you want to find the repository connection details for.
projectName	List of String	-	Project name that you want to find the repository connection details for.

GET response

If successful, a GET request returns the following information:

Field	Type	Description
pullActionId	String	ID for the pull operation.
projectRepoConnections	List <Object>	Object that contains repository details for each project ID or project name included in the request.
projectId	String	Returned in the projectRepoConnections object. ID of the project. Included when the project ID was included in the request.

Field	Type	Description
projectName	String	Returned in the projectRepoConnections object. Name of the project. Included when the project name was included in the request.
gitRepositoryURL	String	Git repository URL.
gitRepositoryBranch	String	Git repository branch.
globalRepository	Boolean	Whether project is configured with a global Git repository. If true, repoConnectionId is null.
repoConnectionId	String	Repository connection ID for the project. Null if the project isn't configured with a project-level Git repository.

GET request example

To request repository connection details for two project using project ID, the request might look like the following example:

```
https://{{pod_link}}/saas/public/core/v3/repositoryConnection?
projectIds=6ub6oUKHJ8Dkr3QXbGG9Sm,5FmdSKPASroei2vyeXwFeT
```

GET response example

For a successful GET request, the response might look like the following example:

```
{
    "projectRepoConnections": [
        {
            "projectId": "6ub6oUKHJ8Dkr3QXbGG9Sm",
            "projectName": "customProject2",
            "globalRepository": false,
            "gitRepositoryURL": "https://priraj@dev.azure.com/priraj/infa-cloud-vcs-
azure/_git/infa-cloud-vcs-azure",
            "gitRepositoryBranch": "main",
            "repoConnectionId": "1rMeeN2te0TetGYVtRS3GS"
        },
        {
            "projectId": "5FmdSKPASroei2vyeXwFeT",
            "projectName": "TestProject",
            "globalRepository": true,
            "gitRepositoryURL": "https://priraj@dev.azure.com/priraj/infa-cloud-vcs/_git/
infa-cloud-vcs",
            "gitRepositoryBranch": "main",
            "repoConnectionId": null
        }
    ]
}
```

Getting the status of a source control operation

Use the sourceControlAction resource to get the status of a source control operation.

You can request the status of a source control operation.

GET request

To receive the status of a source control operation, include the action ID in the following URI::

```
/public/core/v3/sourceControlAction/<action ID>
```

To receive the status for each object in the source control operation, use the following URI:

```
/public/core/v3/sourceControlAction/<action ID>?expand=objects
```

GET response

Returns the following information for the source control operation:

Field	Type	Description
id	String	ID of the source control operation.
action	String	Type of operation. Returns one of the following values: - CHECKIN - CHECKOUT - PULL - UNLINK - UNDO_CHECKOUT
commitHash	String	Unique commit hash. Included when the request is for checkin and pull operations.
startTime	TimeStamp	Start time of the operation.
endTime	TimeStamp	End time of the operation.
status	Object	Includes status information for the operation.
state	String	Included in the status object. Status of the operation. Returns one of the following values: - NOT_STARTED - IN_PROGRESS - SUCCESSFUL - FAILED - WARNING
message	String	Included in the status object. Descriptive status message for the operation.
objects	List<Object>	Lists each object included in the operation. Returned when expand=objects is included in the URI.
target	Object	Included in the objects object. Target object
id	String	Included in the target object. ID of the target object.
path	List<String>	Included in the target object. Complete path of the target object. For example, "Default" , "mt_MappingTask1".
type	String	Included in the target object. Asset type of the target object.
status	Object	Included in the target object. Status information for the target object.

Field	Type	Description
state	String	Included in the status object. Status of the operation. Returns one of the following values: - NOT_STARTED - IN_PROGRESS - SUCCSSFUL - FAILED - SKIPPED - CANCELLED - WARNING
message	String	Included in the status object. Descriptive status message for the operation.

Get response example

If successful, you might receive a response similar to the following example:

```
{
  "id": "drLV4N8PFiuhAbcprrur2W",
  "action": "CHECKIN"
  "commitHash": "1234567abcdefg"
  "startTime": "2020-03-24T22:07:44Z",
  "endTime": "2020-03-24T22:08:14Z",
  "status": {
    "state": "SUCCESSFUL",
    "message": "Checkin Successful"
  },
  "objects": [
    {
      "target": {
        "path": [
          "Versioned_Project",
          "Versioned_Folder",
          "Versioned Mapping - Rename"
        ],
        "id": "2CefbUuBsYxhG6eeKXvGmh",
        "type": "MAPPING"
      },
      "status": {
        "state": "SUCCESSFUL",
        "message": "Checkin Successful"
      }
    },
    {
      "target": {
        "path": [
          "Versioned_Project",
          "Versioned_Folder",
          "Versioned Mapping - Edit"
        ],
        "id": "2CefbUuBsYxhG6eeKXvGmh",
        "type": "MAPPING"
      },
      "status": {
        "state": "FAILED",
        "message": "Checkin Failed."
      }
    }
  ]
}
```

Tags

A tag is an asset property that you can use to group assets.

You can use the following resources:

- TagObjects. Use this resource to assign tags to an asset.
- UntagObjects. Use this resource to remove tags from an asset.

Assigning tags

Use the TagObjects resource to assign tags to an asset.

POST request

To assign a tag to an asset, use the following URI:

```
/public/core/v3/TagObjects
```

You can assign tags to a maximum of 100 assets in a request.

Include the following information for each asset:

Field	Type	Required	Description
id	String	Yes	Global unique identifier for the object.
tags	List	Yes	List of tags to assign to the object.

POST example

To assign tags to two assets, you might use a POST request similar to the following example:

```
POST <baseApiUrl>/public/core/v3/TagObjects
Content-Type: application/json
Accept: application/json
INFA-SESSION-ID: <sessionId>

[{
    "id":"5kuZuAC3Os0dycZuqGpqmM",
    "tags": ["R12 Tag", "DevQA"]

}, {
    "id":"7feHjtC50mLb44CTW4Xmon",
    "tags": ["Prod", "DevQA", "R12 Tag"]

}]
```

Returns the 204 response code if the request is successful. Returns errors if the request is unsuccessful. If the request is partially successful, returns information for the successful and unsuccessful transactions, as shown in the following example:

```
[{
    "id": "9WfGCcHsygueFigGhAdWqh",
    "status": "FAILED",
    "msg": "Object: 9WfGCcHsygueFigGhAdWqh skipped, missing READ/UPDATE permissions."
}, {
    "id": "0cLD48xB4TOgm8cNjP2kmJ",
    "status": "SUCCESS",
    "msg": "Object: 0cLD48xB4TOgm8cNjP2kmJ Operation Message: [Tag assignment succeeded
for artifact 0cLD48xB4TOgm8cNjP2kmJ.]"
}]
```

Removing tags

Use the UntagObjects to remove tags from an asset.

POST request

To remove a tag from an asset, use the following URI:

```
/public/core/v3/UntagObjects
```

You can remove tags from a maximum of 100 assets in a request.

Include the following information for each asset:

Field	Type	Required	Description
id	String	Yes	Global unique identifier for the object.
tags	List	Yes	List of tags to remove from the object.

POST example

To remove tags from two assets, you might use a POST request similar to the following example:

```
POST <baseApiUrl>/public/core/v3/UntagObjects
Content-Type: application/json
Accept: application/json
INFA-SESSION-ID: <sessionId>

[{
    "id":"5kuZuAC3Os0dycZuqGpqmM",
    "tags": ["R12 Tag", "DevQA"]

}, {
    "id":"7feHjtC50mLb44CTW4Xmon",
    "tags": ["DevQA", "R12 Tag"]

}]
```

Returns the 204 response code if the request is successful. Returns errors if the request is unsuccessful. If the request is partially successful, returns information for the successful and unsuccessful transactions, as shown in the following example:

```
[{
    "id": "9WfGCcHsygueFigGhAdWqh",
    "status": "FAILED",
    "msg": "Object: 9WfGCcHsygueFigGhAdWqh skipped, missing READ/UPDATE permissions."
}, {
    "id": "0cLD48xB4TOgm8cNjP2kmJ",
    "status": "SUCCESS",
    "msg": "Object: 0cLD48xB4TOgm8cNjP2kmJ Operation Message: [Tag assignment succeeded
for artifact 0cLD48xB4TOgm8cNjP2kmJ.]"
}]
```

Users

Use the users resource to request Informatica Intelligent Cloud Services user details, create users, update role and user group assignments, and delete users.

Use the users resource along with the userGroups and roles resources to manage user privileges for Informatica Intelligent Cloud Services tasks and assets. Users and groups can perform tasks and access assets based on the roles that you assign to them.

For information about using the userGroups and roles REST API resources, see the following topics:

- "Roles" on page 183
- "User groups" on page 251

For general information about users, user groups, and roles, see the Administrator help.

Getting user details

Use the users resource to request Informatica Intelligent Cloud Services user details. You can request the details for all users in the organization or request the details for a particular user.

GET request

To get user details, use the following URI:

```
/public/core/v3/users
```

To get the details for a particular user, you can include the following query parameters in the URI:

Parameter	Type	Description
q	String	Query filter. You can filter using one of the following fields: - userId. Unique identifier for the user. - userName. Informatica Intelligent Cloud Services user name.
limit	Int	Maximum number of users to return. Maximum of 200. Default is 100.
skip	Int	Amount to offset the list of results. Default is 0.

For example, to get details for a particular user based on the user's ID, you might use the following request:

```
/public/core/v3/users?q=userId==5N9JGth6pRYfOGjGKv3Q2D &limit=1 &skip=0
```

GET response

If successful, returns the following information for each user:

Field	Type	Description
id	String	User ID.
orgId	String	ID of the organization the user belongs to.
createdBy	String	User who created the user account.

Field	Type	Description
updatedBy	String	User who last updated the user account.
createTime	String	Date and time the user was created.
updateTime	String	Date and time the user was last updated.
userName	String	User name for the user account.
firstName	String	First name for the user account.
lastName	String	Last name for the user account.
description	String	User description.
title	String	Job title of the user.
phone	String	Phone number for the user.
email	String	Email address for the user.
state	String	State of the user account. Returns one of the following values: - Active. User account exists and user has activated the account. - Provisioned. User account exists but the user has not activated the account. - Disabled. User account is disabled because the user exceeded the maximum number of login attempts. **Note:** If the user's password is expired, the value is null.
timeZoneId	String	Time zone of the user. For more information, see "Time zone codes" on page 543.
maxLoginAttempts	Int	Number of times a user can attempt to log in before the account is locked.
authentication	String	Whether the user accesses Informatica Intelligent Cloud Services through single sign-in (SAML). Returns one of the following values: - 0. Native. - 1. SAML.
forcePasswordChange	Boolean	Whether the user must reset the password after the user logs in for the first time.
lastLoginTime	String	The date and time that the user last logged in.
lastLoginMode	String	Whether the user logged in through a REST API call or through the UI.
roles	Array	Roles assigned to the user account.
id	String	Included in the roles object. Role ID.
roleName	String	Included in the roles object. Role name.

Field	Type	Description
description	String	Included in the roles object. Description of the role.
displayName	String	Included in the roles object. Role name that is displayed in the user interface.
displayDescription	String	Included in the roles object. Role description that is displayed in the user interface.
groups	Array	Group in which the user account is a member.
id	String	Included in the groups object. User group ID.
userGroupName	String	Included in the groups object. User group name.
description	String	Included in the groups object. Description of the user group.

GET response example

If successful, you might receive a response similar to the following example:

```
[
    {
        "id": "5N9JGth6pRYfOGjGKv3Q2D",
        "orgId": "cPYWk02I4aBeuLEvYRtaMS",
        "createdBy": "ma",
        "updatedBy": "a@abc.com",
        "createTime": "2019-03-06T22:04:00.000Z",
        "updateTime": "2019-03-18T22:34:53.000Z",
        "userName": "a@abc.com",
        "firstName": "a",
        "lastName": "jones",
        "description": "",
        "title": "dev",
        "phone": "1112221111",
        "email": "a@abc.com",
        "state": "Enabled",
        "timeZoneId": "America/Los_Angeles",
        "maxLoginAttempts": "10",
        "authentication": "Native",
        "forcePasswordChange": false,
        "lastLoginTime": "2020-07-31T21:50:10Z",
        "lastLoginMode": "API",
        "roles": [
            {
                "id": "9c2XrdpAz80hg29yXDBPEN",
                "roleName": "Data Preview",
                "description": "Role to preview data"
                "displayName": "Data Preview",
                "displayDescription": "Role to preview data"
            },
            {
                "id": "1VfnsgZiCT1fi25VAupQg1",
                "roleName": "Designer",
                "description": "Role for creating assets, ... and runtime environments.
    Has access to the Application Integration Console."
                "displayName": "Designer",
                "displayDescription": "Role for creating assets, ... and runtime
```

```
                environments. Has access to the Application Integration Console."
                    }
              ],
              "groups": [
                    {
                        "id": "a6x85hoMvH2kWUIlcIRBEh",
                        "userGroupName": "group_a",
                        "description": ""
                    }
              ]
        },
        {
              "id": "aNJWtppg613c1YbXvRRHcV",
              "orgId": "cPYWk02I4aBeuLEvYRtaMS",
              "createdBy": "a@abc.com",
              "updatedBy": "a@abc.com",
              "createTime": "2019-03-13T20:15:58.000Z",
              "updateTime": "2019-03-13T20:15:58.000Z",
              "userName": "b@abc.com",
              "firstName": "b",
              "lastName": "smith",
              "description": "",
              "title": "cs",
              "phone": "1112223333",
              "email": "b@abc.com",
              "state": "Provisioned",
              "timeZoneId": "America/Los_Angeles",
              "maxLoginAttempts": "10",
              "authentication": "Native",
              "forcePasswordChange": false,
              "lastLoginTime": "2020-07-31T21:50:10Z",
              "lastLoginMode": "API",
               "roles": [
                    {
                        "id": "9c2XrdpAz80hg29yXDBPEN",
                        "roleName": "Data Preview",
                        "description": "Role to preview data"
                        "displayName": "Data Preview",
                        "displayDescription": "Role to preview data"
                    }
              ],
              "groups": [
                    {
                        "id": "a6x85hoMvH2kWUIlcIRBEh",
                        "userGroupName": "group_a",
                        "description": ""
                    }
              ]
        }
    ]
```

Creating a user

If you have administrator privileges, you can use the users resource to create a user.

POST request

To create a user, send a POST request using the following URI:

```
/public/core/v3/users
```

Note: The number of users, user groups, and roles combined cannot exceed 1000 for an organization.

Include the following information:

Field	Type	Required	Description
name	String	Yes	Informatica Intelligent Cloud Services user name. Maximum length is 255 characters.
firstName	String	Yes	First name for the user account.
lastName	String	Yes	Last name for the user account.
password	String	-	Informatica Intelligent Cloud Services password. If password is empty, the user receives an activation email. Maximum length is 255 characters.
description	String	-	Description of the user.
email	String	Yes	Email address for the user.
title	String	-	Job title of the user.
phone	String	-	Phone number for the user.
forcePasswordChange	Boolean	-	Determines whether the user must reset the password after the user logs in for the first time.
maxLoginAttempts	Int	-	Number of times a user can attempt to log in before the account is locked.
authentication	Int	-	Determines whether the user accesses Informatica Intelligent Cloud Services through single sign-in (SAML). Use one of the following values: - 0. Native. - 1. SAML.
aliasName	String	Required when authentication is not 0.	The user identifier or user name in the 3rd party system.
roles	Array	Required when no group IDs are included.	IDs of the roles to assign to the user.
groups	Array	Required when no role IDs are included.	IDs of the user groups to assign to the user.

POST response

If successful, returns the users object with the details you included in the POST request.

POST example

To create a user, you might send a request similar to the following example:

```
POST <baseApiUrl>/public/core/v3/users
Content-Type: application/json
Accept: application/json
INFA-SESSION-ID: <sessionId>
{
```

```
            "name" : "c@abc.com",
            "firstName" : "c",
            "lastName" : "smith",
            "email" : "c@abc.com",
            "authentication" : 0,
            "roles" : ["5IPgtye09EbiWqz5XXuzwC", "9gedBDoYQoQibNMohf5KCh"],
            "groups" : ["a6x85hoMvH2kWUIlcIRBEh"]
    }
```

You might receive a response similar to the following example:

```
    {
        "id": "9EcgvBYZ9GGflOYr98GzOH",
        "orgId": "cPYWk02I4aBeuLEvYRtaMS",
        "createdBy": "a@abc.com",
        "updatedBy": "a@abc.com",
        "createTime": "2020-08-20T18:29:19.987Z",
        "updateTime": "2020-08-20T18:29:20.653Z",
        "userName": "c@abc.com",
        "firstName": "c",
        "lastName": "smith",
        "description": null,
        "title": "dev",
        "phone": null,
        "email": "c@abc.com",
        "state": "Provisioned",
        "timeZoneId": "America/Los_Angeles",
        "maxLoginAttempts": "10",
        "authentication": "Native",
        "forcePasswordChange": false,
        "lastLoginTime": null,
        "lastLoginMode": "None",
        "roles": [
            {
                "id": "5IPgtye09EbiWqz5XXuzwC",
                "roleName": "test",
                "description": ""
            },
            {
                "id": "9gedBDoYQoQibNMohf5KCh",
                "roleName": "Admin",
                "description": "Role for performing administrative tasks for an
    organization. Has full access to all licensed services."
            }
        ],
        "groups": [
            {
                "id": "a6x85hoMvH2kWUIlcIRBEh",
                "userGroupName": "group_a",
                "description": ""
            }
        ]
    }
```

Updating role assignments

You can add or remove role assignments for a user.

You cannot update a user's role assigments if your organization enabled the Map SAML Groups and Roles option for SAML single sign-on. For more information, see *User Administration* in the Administrator help.

Assign roles

To assign a role to a user, send a PUT request using one of the following URIs:

```
/public/core/v3/users/<user ID>/addRoles
```

```
/public/core/v3/users/name/<user name>/addRoles
```

Include an array of the roles that you want to assign to the user.

For example, to assign the Admin and Business Manager roles to a user, you might use the following request:

```
PUT <baseApiUrl>/public/core/v3/users/cPYWk02I4aBeuLEvYRtaMS/addRoles
Content-Type: application/json
Accept: application/json
INFA-SESSION-ID: <sessionId>
{
    "roles" : ["Admin", "Business Manager"]
}
```

You can use the roles resource to get a list of roles that you can assign. For more information, see "Roles" on page 183.

The response includes a success code if successful or an error object if errors occur.

Remove roles

To remove a role assignment from a user, send a PUT request using one of the following URIs:

```
/public/core/v3/users/<user ID>/removeRoles
```

```
/public/core/v3/users/name/<user name>/removeRoles
```

Include an array of roles to unassign.

For example, to remove the Designer role assignment from a user, you might use the following request:

```
PUT <baseApiUrl>/public/core/v3/users/cPYWk02I4aBeuLEvYRtaMS/removeRoles
Content-Type: application/json
Accept: application/json
INFA-SESSION-ID: <sessionId>
{
    "roles" : "Designer"
}
```

Updating user group assignments

You can add or remove user group assignments for a user.

You cannot update a user's user group assignments if your organization enabled the Map SAML Groups and Roles option for SAML single sign-on. For more information, see *User Administration* in the Administrator help.

Assign user groups

To assign a user group to a user, send a PUT request using one of the following URIs:

```
/public/core/v3/users/<user ID>/addGroups
```

```
/public/core/v3/users/name/<user name>/addGroups
```

Include an array of the user groups that you want to assign to the user.

For example, to assign the Workflow Manager and MDM Admin user groups to a user, you might use the following request:

```
PUT <baseApiUrl>/public/core/v3/users/cPYWk02I4aBeuLEvYRtaMS/addGroups
Content-Type: application/json
Accept: application/json
INFA-SESSION-ID: <sessionId>
{
    "groups" : ["Workflow Manager", "MDM Admin"]
}
```

You can use the userGroups resource to get a list of user groups that you can assign. For more information, see "User groups" on page 251.

The response includes a success code if successful or an error object if errors occur.

Remove user group assignments

To remove a user group assignment, send a PUT request using one of the following URIs:

```
/public/core/v3/users/<user ID>/removeGroups
```

```
/public/core/v3/users/name/<user name>/removeGroups
```

Include an array of user group assignments to remove from the user.

For example, to remove the MDM Admin user group assignment, you might use the following request:

```
PUT <baseApiUrl>/public/core/v3/users/name/LarryR/removeGroups
Content-Type: application/json
Accept: application/json
INFA-SESSION-ID: <sessionId>
{
    "groups" : "MDM Admin"
}
```

Deleting a user

If you have administrator privileges, you can use the users resource to delete a user.

DELETE request

To delete a user account, use the following URI:

```
/public/core/v3/users/<userId>
```

For example, you might send a request similar to the following example:

```
DELETE <baseApiUrl>/public/core/v3/users/5N9JGth6pRYfOGdGKv3Q2D
```

User groups

Use this resource to create, update, and delete user groups.

Use the userGroups resource along with the users and roles resources to manage user privileges. Users and groups can perform tasks and access assets based on the roles that you assign to them.

You can send the following requests:

- To get details about all of the user groups or a particular user group, send a GET request.
- To create a user group, use a POST request.
- To add roles or users to a user group, use a PUT request.
- To delete a user group, use a DELETE request.

For information about using the users and roles REST API resources, see the following topics:

-
-

For general information about users, user groups, and roles, see the Administrator help.

Note: The number of users, user groups, and roles combined cannot exceed 1000 for an organization.

Getting user group details

You can request the details for all user groups in the organization or request the details for a particular user group.

GET request

You can request the details for all user groups in the organization or request the details for a particular user group.

To get user group details, use the following URI:

```
/public/core/v3/userGroups
```

To get the details for a particular user group, you can include the following query parameters in the URI:

Parameter	Type	Description
q	String	Query filter. You can filter using one of the following fields: - userGroupId. Unique identifier for the user group. - userGroupName. Name of the user group.
limit	Int	Maximum number of user groups to return. Default is 100.
skip	Int	Amount to offset the list of results. Default is 0.

For example, to get details for a particular user group using the user group's name, you might use the following request:

```
public/core/v3/userGroups?q=userGroupName=="group_a"
```

GET response

If successful, returns the following information for each user group:

Field	Type	Description
id	String	User group ID.
orgId	String	ID of the organization the user group belongs to.
createdBy	String	User who created the user account.
updatedBy	String	User who last updated the user account.
createTime	String	Date and time the user group was created.
updateTime	String	Date and time the user group was last updated.
userGroupName	String	Name of the user group.
description	String	Description of the user group.
roles	Array	Roles assigned to the user group.

Field	Type	Description
id	String	Included in the roles object. Role ID.
roleName	String	Included in the roles object. Role name.
description	String	Included in the roles object. Description of the role.
displayName	String	Included in the roles object. Role name that is displayed in the user interface.
displayDescription	String	Included in the roles object. Role description that is displayed in the user interface.
users	Array	Users assigned to the user group.
id	String	Included in the users object. User ID.
userName	String	Included in the users object. User name.
description	String	Included in the users object. Description of the user.

GET response example

If successful, you might receive a response similar to the following example:

```
[
    {
        "id": "a6x85hoMvH2kWUIlcIRBEh",
        "orgId": "cPYWk02I4aBeuLEvYRtaMS",
        "createdBy": "a@abc.com",
        "updatedBy": "a@abc.com",
        "createTime": "2019-03-19T17:27:09.000Z",
        "updateTime": "2019-03-19T17:27:09.000Z",
        "userGroupName": "group_a",
        "description": "",
        "roles": [
            {
                "id": "9gedBDoYQoQibNMohf5KCh",
                "roleName": "Admin",
                "description": "Role for performing administrative tasks for an
organization. Has full access to all licensed services."
                "displayName": "Admin",
                "displayDescription": "Role for performing administrative tasks for an
organization. Has full access to all licensed services."
            }
        ],
        "users": [
            {
                "id": "5N9JGth6pRYfOGjGKv3Q2D",
                "userName": "a@abc.com",
                "description": ""
            }
        ]
```

```
        }
    ]
```

Creating a user group

If you have administrator privileges, you can use the userGroups resource to create a user group.

Note: The number of users, user groups, and roles combined cannot exceed 1000 for an organization.

POST request

To create a user group, send a POST request using the following URI:

```
/public/core/v3/userGroups
```

Include the following information:

Field	Type	Required	Description
name	String	Yes	Name of the user group.
description	String	-	Description of the user group.
roles	Array	Yes	IDs of the roles to assign to the user group.
users	Array	-	IDs of the users to assign to the user group.

POST response

If successful, returns the userGroups object with the details you included in the POST request.

POST example

To create a user group, you might send a request similar to the following example:

```
POST <baseApiUrl>/public/core/v3/userGroups
Content-Type: application/json
Accept: application/json
INFA-SESSION-ID: <sessionId>
{
    "name" : "user_group_1",
    "roles" : ["5IPgtye09EbiWqz5XXuzwC", "9gedBDoYQoQibNMohf5KCh"],
    "users" : ["9EcgvBYZ9GGflOYr98GzOH"]
}
```

You might receive a response similar to the following example:

```
{
    "id": "0TLmCMwX0jNdJ5SzlQC2CW",
    "orgId": "cPYWk02I4aBeuLEvYRtaMS",
    "createdBy": "a@abc.com",
    "updatedBy": "a@abc.com",
    "createTime": "2019-03-20T18:30:32.457Z",
    "updateTime": "2019-03-20T18:30:32.472Z",
    "userGroupName": "user_group_1",
    "description": null,
    "roles": [
        {
            "id": "9gedBDoYQoQibNMohf5KCh",
            "roleName": "Admin",
            "description": "Role for performing administrative tasks for an
organization. Has full access to all licensed services."
            "displayName": "Admin",
            "displayDescription": "Role for performing administrative tasks for an
organization. Has full access to all licensed services."
        },
```

```
        {
            "id": "5IPgtye09EbiWqz5XXuzwC",
            "roleName": "test_user_1",
            "description": ""
            "roleName": "test_user_1",
            "description": ""
        }
    ],
    "users": [
        {
            "id": "9EcgvBYZ9GGflOYr98GzOH",
            "userName": "test_user_2",
            "description": null
        }
    ]
}
```

Updating a user group

You can add users and roles to user groups and remove users and roles from user groups.

Add roles

To add a role to a user group, send a PUT request using one of the following URIs:

```
/public/core/v3/userGroups/<user group ID>/addRoles
```

```
/public/core/v3/userGroups/name/<user group name>/addRoles
```

Include an array of roles to add to the user group.

To add roles to a user group, you might send a request similar to the following example:

```
PUT <baseApiUrl>/public/core/v3/userGroups/cPYWk02I4aBeuLEvYRtaMS/addRoles
Content-Type: application/json
Accept: application/json
INFA-SESSION-ID: <sessionId>
{
    "roles" : ["Admin", "Business Manager"]
}
```

Remove roles

To remove a role from a user group, send a PUT request using one of the following URIs:

```
/public/core/v3/userGroups/<user group ID>/removeRoles
```

```
/public/core/v3/userGroups/name/<user group name>/removeRoles
```

Include an array of roles to remove from the user group.

To remove roles from a user group, you might send a request similar to the following example:

```
PUT <baseApiUrl>/public/core/v3/userGroups/cPYWk02I4aBeuLEvYRtaMS/removeRoles
Content-Type: application/json
Accept: application/json
INFA-SESSION-ID: <sessionId>
{
    "roles" : ["Admin", "Business Manager"]
}
```

Add users

To add a user to a user group, send a PUT request using one of the following URIs:

```
/public/core/v3/userGroups/<user group ID>/addUsers
```

```
/public/core/v3/userGroups/name/<user group name>/addUsers
```

Include an array of user names to add to the user group.

To add users to a user group, you might send a request similar to the following example:

```
PUT <baseApiUrl>/public/core/v3/userGroups/cPYWk02I4aBeuLEvYRtaMS/addUsers
Content-Type: application/json
Accept: application/json
INFA-SESSION-ID: <sessionId>
{
    "users" : ["LarryR", "ScottY"]
}
```

Remove users

To remove a user from a user group, send a PUT request using one of the following URIs:

```
/public/core/v3/userGroups/<user group ID>/removeUsers
```

```
/public/core/v3/userGroups/name/<user group name>/removeUsers
```

Include an array of user names to remove from the user group.

To remove users from a user group, you might send a request similar to the following example:

```
PUT <baseApiUrl>/public/core/v3/userGroups/cPYWk02I4aBeuLEvYRtaMS/removeUsers
Content-Type: application/json
Accept: application/json
INFA-SESSION-ID: <sessionId>
{
    "users" : ["LarryR", "ScottY"]
}
```

PUT response

Returns a success code if successful or an error object if errors occur.

Deleting a user group

You can delete user groups from your organization.

To delete a user group, use the following URI:

```
/public/core/v3/userGroups/<user group Id>
```

CHAPTER 4

Data Integration REST API

The REST API resources in this section apply specifically to the Data Integration service.

For most calls, use Data Integration REST API version 2 resources. When you use Data Integration REST API version 2 resources, note the following rules:

- Use JSON or XML format.
- Use the serverUrl value from the login response as the base URL. For example:

    ```
    https://na4.dm-us.informaticacloud.com/saas
    ```

- Use the following URI:

    ```
    /api/v2/<API name>
    ```

- Use the following request header format:

    ```
    <METHOD> <serverUrl>/<URI> HTTP/<HTTP version>
    Content-Type: application/json
    Accept: application/json
    icSessionId: <SessionId>
    ```

In the following example, the serverUrl is `https://na4.dm-us.informaticacloud.com/saas` and the URI is `/api/v2/agent`:

```
<METHOD> https://na4.dm-us.informaticacloud.com/saas/api/v2/agent HTTP/1.1
Content-Type: application/json
Accept: application/json
icSessionId: IV4wOrJmd6YUtmKa8t
```

Format rules for resources other than the version 2 resources are included in the appropriate resource topics.

Note: If you use a tool such as Postman that automatically includes the HTTP version, do not enter the HTTP version in the URL. If the HTTP version appears twice in the URL, the request fails.

CodeTask

Use the code task API to submit Spark code written in Scala to an advanced cluster. You can use the CodeTask resource to create, start, and cancel a code task job. You can also access session logs, view job details and job status of a code task.

Consider the following guidelines when you use the code task resource:

- Write your code task in Scala.
- Submit your code in a JAR file using the code task APIs.
- Use an AWS serverless or non-serverless environment.

- Use the following base URL:

  ```
  <server URL>/disnext/api/v1/<API name>
  ```

- Use the following request header format:

  ```
  <METHOD> <server URL>/<URI> HTTP/<HTTP version>
  Content-Type: application/json
  Accept: application/json
  IDS-SESSION-ID: <IDS_SESSION_ID>
  ```

- Ensure that you have the permission to create, execute, and view the code task APIs.

- Use the following persisted variables as needed for the code task APIs:

  - IDS_SESSION_ID

  - ORG_ID

  - CODE_TASK_ID

  - CODE_TASK_JOB_ID

Note: If you use a tool such as Postman that automatically includes the HTTP version, do not enter the HTTP version in the URL. If the HTTP version appears twice in the URL, the request fails.

Complete the following tasks to submit Scala code in a JAR file and manage and monitor code task jobs:

- Send login information to get the session ID using .

- Create the code task and get the code task ID using .

- Start the code task and get the job ID using .

- View code task details using .

- Check on the job status for the code task using .

- Cancel the code task job using .

- Access the session logs for the code task using .

- Access the Spark task results for the code task using .

Create a code task

Use the CodeTask resource to create a code task. The response includes the code task ID that you can use in subsequent API calls.

POST request

Use the following URI to create a code task:

```
POST <server URL>/disnext/api/v1/CodeTask
```

Use the following fields in the POST request:

Field	Type	Required / Optional	Description
codeTaskName	String	Required	Name of the code task.
runtimeEnvironmentName	String	Optional	Runtime environment used for the code task. Either the runtimeEnvironmentName or the agentGroupId is required. If both are provided, then the agentGroupId is used.

Field	Type	Required / Optional	Description
codeExecutionParameters			Parameters in the code task.
agentGroupId	String	Required	Runtime environment that contains the Secure Agent used to run the code task. Either the runtimeEnvironmentName or the agentGroupId is required. If both are provided, then the agentGroupId is used.
overrideTaskTimeout	Long	Optional	Overrides the code task timeout value for this execution. A value of -1 signifies no timeout.
logLevel	String	Optional	Log level for session logs, agent job log, Spark driver, and executor logs. Valid values are: none, terse, normal, verboseInitialization, or verboseData. The default value is normal.
sparkMainClass	String	Required	Entry point of the Spark application. For example: org.apache.spark.examples.company.SparkExampleApp
sparkMainClassArgs	List <String>	Optional	Ordered arguments sent to the Spark application main class. For example: --appTypeSPARK_PI_FILES_JARS-- classesToLoadcom.company.test.SparkTest1Class
sparkPrimaryResource	String	Required	Scala JAR file that contains the code task.
sparkJars	List <String>	Optional	The directory and file name of the JAR file that is uploaded to the cluster and added to the Spark driver and executor classpaths.
sparkFiles	List <String>	Optional	The directory and file name of the Spark file that is uploaded to the cluster and available under the current working directory.
advancedCustomProperties	String	Optional	Spark properties or other custom properties that Data Integration uses. For example: "{\"spark.driver.memory\": \"2G\", \"spark.executor.instances\": \"4\"}"

POST response

The following variable is set from the response attributes:

Name	Response Value	Note
CODE_TASK_ID	codeTaskId	Used in the start and view code task resources.

POST request example

Use the following sample as a reference to create a code task:

```
POST <server URL>/disnext/api/v1/CodeTask
Content-Type: application/json
```

```
Accept: application/json
IDS-SESSION-ID:{{IDS_SESSION_ID}}
{
    "codeTaskName" : "CODETASK_API",
    "runtimeEnvironmentName" : "{{RTE_NAME}}",
    "codeExecutionParameters" : {
        "agentGroupId": "{{AGENT_GROUP_ID}}",
        "logLevel": "normal",
        "sparkMainClass":
"org.apache.spark.examples.infa.sparkdirect.SparkDirectExampleApp",
        "sparkMainClassArgs": ["6"],
        "sparkPrimaryResource": "spark-examples_2.12-3.0.0.jar",
        "sparkJars": [],
        "sparkFiles": [],
        "advancedCustomProperties": "{\"spark.driver.memory\": \"1G\",
\"spark.executor.memory\": \"1G\", \"spark.kubernetes.driverEnv.SPARK_DIRECT_TASK_SLEEP
\": \"600\", \"spark.kubernetes.driverEnv.SPARK_DIRECT_APP_TYPE\": \"SPARK_PI\",
\"spark.kubernetes.driverEnv.SPARK_DIRECT_KMS_ENCRYPTED_PROPS\":
\"spark.sparkdirect.kms.prop\", \"spark.sparkdirect.kms.prop\":
\"5pkOjS0HILDwSaW6eyxtiwB3g2TBYayjKLRFSSyxn5M=0p6v3eCvrtFkw6K78Buwal\",
\"advanced.custom.property\": \"infa.spark.local=false\"}"
    }
}
```

POST response example

A successful POST response returns a summary, the code task ID, and the code task name similar to the following example:

```
{
    "summary": "Code Task created successfully",
    "codeTaskId": 3,
    "codeTaskName": "CODETASK_API"
}
```

Start a code task

Use this POST request to start the code task job. The response includes the code task job ID that you can use in subsequent API calls.

POST request

To start a code task, use the task ID. The code task ID is included in the response when you create a code task. For more information, see .

Use the following URI to start a code task:

```
POST <server URL>/disnext/api/v1/CodeTask/Start
```

Use the following field in the POST request:

Field	Type	Required / Optional	Description
codeTaskId	String	Required	The code task ID from the create resource.

POST response

The following variable is set from the response attributes:

Name	Response Value	Note
CODE_TASK_JOB_ID	jobId	Used to cancel the code task job, or to get job status, session logs, or Spark task status.

POST request example

Use this sample as a reference to start a code task.

```
POST <server URL>/disnext/api/v1/CodeTask/Start
Content-Type: application/json
Accept: application/json
IDS-SESSION-ID:{{IDS_SESSION_ID}}
{
    "codeTaskId" : {{CODE_TASK_ID}}
}
```

POST response example

A successful POST response returns the code task job ID and other information, similar to the following example:

```
{
    "summary": "Code Task started successfully",
    "jobId": "8zcuMdImeshidZ4XVExs20",
    "codeTaskInstanceName": "Demo-2",
    "sparkCodeTaskResponseBody": {
        "agentGroupId": "01000025000000000003",
        "clusterConfigId": "Default",
        "logLevel": "normal",
        "startRunTime": "2022-04-04T20:23:57.154+00:00",
        "submitTime": "2022-04-04T20:23:57.095+00:00"
    }
}
```

View a code task

Use this GET request to view the code task configuration parameters.

GET request

To request the parameters of a code task, use the task ID. The code task ID is included in the response when you create a code task. For more information, see .

Use the following URI to get the code task parameters:

```
GET <server URL>/disnext/api/v1/CodeTask/<Code task ID>
```

GET response

Returns the code task object for the requested task ID.

Returns the error object if errors occurred.

The following table describes the parameters in the response:

Field	Type	Description
codeTaskName	String	Name of the code task.
codeTaskId	Numeric	The code task identifier.
agentGroupId	String	Runtime environment that contains the Secure Agent used to run the code task.
overrideTaskTimeout	String	Overrides the code task timeout value for this execution. A value of -1 signifies no timeout.
logLevel	String	Log level for session logs, agent job log, Spark driver, and executor logs. Valid values are: none, terse, normal, verboseInitialization, or verboseData. The default value is normal.
sparkMainClass	String	Entry point of the Spark application. For example: org.apache.spark.examples.company.SparkExampleApp
sparkMainClassArgs	List<String>	Ordered arguments sent to the Spark application main class. For example: --appTypeSPARK_PI_FILES_JARS-- classesToLoadcom.company.test.SparkTest1Class
sparkPrimaryResource	String	Scala JAR file that contains the code task.
sparkJars	List<String>	The directory and file name of the JAR file that is uploaded to the cluster and added to the Spark driver and executor classpaths.
sparkFiles	List<String>	The directory and file name of the Spark file that is uploaded to the cluster and available under the current working directory.
advancedCustomProperties	String	Spark properties or other custom properties that Data Integration uses.

GET request example

Use this sample as a reference to get the code task parameters.

```
GET <server URL>/disnext/api/v1/CodeTask/<Code task ID>
Content-Type: application/json
Accept: application/json
IDS-SESSION-ID:{{IDS_SESSION_ID}}
```

GET response example

A successful GET response returns the code task parameters, similar to the following example:

```
{
    "codeTaskName": "CODETASK_API",
    "codeTaskId": 3,
    "agentGroupId": "01000025000000000003",
    "overrideTaskTimeout": null,
    "logLevel": "normal",
    "sparkMainClass": "org.apache.spark.examples.infa.sparkdirect.SparkDirectExampleApp",
    "sparkMainClassArgs": ["6"],
    "sparkPrimaryResource": "spark-examples_2.12-3.0.0.jar",
```

```
    "sparkJars": [],
    "sparkFiles": [],
    "advancedCustomProperties": "{\"spark.driver.memory\": \"1G\",
\"spark.executor.memory\": \"1G\", \"spark.kubernetes.driverEnv.SPARK_DIRECT_TASK_SLEEP
\": \"600\", \"spark.kubernetes.driverEnv.SPARK_DIRECT_APP_TYPE\": \"SPARK_PI\",
\"spark.kubernetes.driverEnv.SPARK_DIRECT_KMS_ENCRYPTED_PROPS\":
\"spark.sparkdirect.kms.prop\", \"spark.sparkdirect.kms.prop\":
\"5pkOjS0HILDwSaW6eyxtiwB3g2TBYayjKLRFSSyxn5M=0p6v3eCvrtFkw6K78Buwal\",
\"advanced.custom.property\": \"infa.spark.local=false\"}"
}
```

Status of a code task

Use this GET request to check the code task job status.

GET request

To request the status of a code task job, use the code task job ID. The code task job ID is included in the response when you start a code task. For more information, see .

Use the following URI to get the code task job status parameters:

```
GET <server URL>/disnext/api/v1/CodeTask/JobStatus/<Code task job ID>
```

GET response

Returns the code task object for the requested job ID.

Returns the error object if errors occurred.

The following table describes the parameters in the response:

Field	Type	Description
codeTaskName	String	Name of the code task.
codeTaskId	Numeric	The code task identifier.
agentGroupId	String	Runtime environment that contains the Secure Agent used to run the code task.
executionState	String	The state of the job: QUEUED, RUNNING, SUCCEEDED, FAILED, CANCELLED Job status. A job can have one of the following statuses: - Starting. The job is starting. - Running. The job is either queued or running. - Success. The job completed successfully. - Failed. The job did not complete because it encountered errors.
sessionLogUrl	String	URL of the session log file.
assignedServerlessComputeUnits	Double	Number of serverless compute units per hour that the task requested. You can view the number of requested compute units if the task runs in a serverless runtime environment.

Field	Type	Description
consumedServerlessComputeUnits	Double	Total number of serverless compute units that the task consumed. You can view the number of consumed compute units if the task runs in a serverless runtime environment.
firstErrorMessage	String	Error message, if any, that is associated with the job.

GET request example

Use this sample as a reference to get the code task job status.

```
GET <server URL>/disnext/api/v1/CodeTask/JobStatus/<Code task job ID>
Content-Type: application/json
Accept: application/json
IDS-SESSION-ID:{{IDS_SESSION_ID}}
```

GET response example

A successful GET response returns the code task job status, similar to the following example:

```
{
    "status": "RUNNING",
    "startTime": "2022-04-04T20:23:57.000",
    "updateTime": "2022-04-04T20:23:58.000",
    "endTime": null,
    "instanceName": "Demo-2",
    "assetName": "Demo",
    "runId": 2,
    "orgId": "kuJVH54wm6gfhVj4QEdboY",
    "startedBy": "coder@examplecompany.com",
    "runtimeEnvId": "01000025000000000003",
    "codeTaskId": 2,
    "errorMessage": "",
    "sessionLogUrl": "logservice/api/v1/jobs/8zcuMdImeshidZ4XVExs20/logs",
    "agentJobLogUrl": null,
    "advancedLogLocation": null,
    "advancedLogDownloadUrl": null
}
```

An error status GET response returns the code task job status, similar to the following example:

```
{
    "status": "FAILED",
    "startTime": "2022-03-10T17:23:29.000",
    "updateTime": "2022-03-10T17:23:34.000",
    "endTime": "2022-03-10T17:23:34.000",
    "instanceName": "Demo-2",
    "assetName": "Demo",
    "runId": 2,
    "orgId": "4nuRA6NIsI6lvUBYamL76P",
    "startedBy": "coder@examplecompany.com",
    "runtimeEnvId": "01000025000000000003",
    "codeTaskId": 477,
    "errorMessage": "WES_internal_error_Failed to start cluster for
[01000025000000000003]. Error reported while starting cluster [404 {\"code
\":\"CONFIG.NOT_FOUND_id\",\"message\":\"Internal error. Cannot find an advanced
configuration with ID 01000025000000000003. Contact Informatica Global Customer
Support.\",\"debugMessage\":\"Internal error. Cannot find an advanced
configurati[truncated]. For more information about the failure, check the application
log.If the problem persists, contact Informatica Global Customer Support.",
    "sessionLogUrl": "logservice/api/v1//jobs/6pqqt9KTgi3l9jyVnAajei/logs",
    "agentJobLogUrl": null,
    "advancedLogLocation": null,
    "advancedLogDownloadUrl": null
}
```

Cancel a code task

Use this PUT request to cancel the code task job.

PUT request

To cancel a code task job, use the code task job ID. The code task job ID is included in the response when you start a code task. For more information, see

Use the following URI to cancel a code task job:

```
PUT <server URL>/disnext/api/v1/CodeTask/Cancel
```

Use the following field in the PUT request:

Field	Type	Required / Optional	Description
jobId	String	Required	The code task job ID from the start resource.

PUT response

Returns the code task cancellation state for the requested job ID.

Returns the error object if errors occurred.

PUT request example

Use this sample as a reference to cancel a code task job.

```
PUT <server URL>/disnext/api/v1/CodeTask/Cancel
Content-Type: application/json
Accept: application/json
IDS-SESSION-ID:{{IDS_SESSION_ID}}
{
    "jobId" : {{CODE_TASK_JOB_ID}}
}
```

PUT response example

A successful PUT response returns the code task job cancellation information, similar to the following example:

```
{
    "state": "Code Task marked for cancellation.",
    "jobId": "6pqqt9KTgi3l9jyVnAajei"
}
```

Session logs for a code task

Use the jobs resource to access the code task session logs.

GET request

To access the session logs of a code task, use the code task job ID. The code task job ID is included in the response when you start a code task. For more information, see

Use the following URI to get code task session logs:

```
GET <server URL>/logservice/api/v1/jobs/<Code task job ID>/logs
```

You can use the Postman Send and Download option.

GET response

Returns a link to the code task ZIP file for the requested job ID.

Returns the error object if errors occurred.

GET request example

Use this sample as a reference to access code task session logs.

```
GET <server URL>/logservice/api/v1/jobs/<Code task job ID>/logs
Content-Type: application/json
Accept: application/json
IDS-SESSION-ID:{{IDS_SESSION_ID}}
```

Spark task results for a code task

Use the jobs resource to access the Spark task results for the code task.

GET request

To access the Spark task results of a code task, use the code task job ID. The code task job ID is included in the response when you start a code task. For more information, see .

Use the following URI to access code task Spark task results:

```
GET <server URL>/logservice/api/v1/jobs/<Code task job ID>/callback
```

GET response

Returns the code task Spark task details for the requested job ID.

Returns the error object if errors occurred.

GET request example

Use this sample as a reference to access code task Spark task results.

```
GET <server URL>/logservice/api/v1/jobs/<Code task job ID>/callback
Content-Type: application/json
Accept: application/json
IDS-SESSION-ID:{{IDS_SESSION_ID}}
```

GET response example

A successful GET response returns the code task Spark task results, similar to the following example:

```
{

    "jobId": "73LJ52V0tiNjC2HFb2YggK",

    "callbackBody": "{\"taskId\": \"73LJ52V0tiNjC2HFb2YggK\", \"executionId\":
\"73LJ52V0tiNjC2HFb2YggK\", \"sessionLogUrl\": \"logservice/api/v1/jobs/
73LJ52V0tiNjC2HFb2YggK/logs\", \"executionState\": \"RUNNING\"}"

}
```

connection

Use this resource to request connection details for an organization. You can also use this resource to create, update, test, and delete a connection.

Details GET Request

You can request the following details using a connection GET request:

- Details of all connections in the organization.
- Details for a particular connection in the organization.
- List of objects that you can use as a source or target with a particular connection.
- List of connections of a specified type associated with a Secure Agent or runtime environment.
- Metadata details for a specified connection.

Note: Do not use connections with names that begin with "DI Data Preview_".

Details of all connections in the organization

To request the details of all connections in the organization, use the following URI:

```
/api/v2/connection
```

Details of a particular connection

To request the details of a particular connection, include the connection ID or name in the URI. Use one of the following URIs:

```
/api/v2/connection/<id>
/api/v2/connection/name/<name>
```

If you use the connection name in the URI and the connection name includes a space, replace the space with `%20`. For example:

```
/api/v2/connection/name/my%20connection
```

List of objects that you can use as a source or target

You can request the objects that you can use as a source or target. To request source or target objects, you can include either the connection ID or connection name in the URI. Use one of the following URIs:

```
/api/v2/connection/source/<id>
/api/v2/connection/target/<id>
/api/v2/connection/source/name/<name>
/api/v2/connection/target/name/<name>
```

If you use the connection name in the URI and the connection name includes a space, replace the space with `%20`. For example:

```
/api/v2/connection/target/name/my%20connection
```

If you expect to receive a large number of objects, you might want to include one of the following parameters:

- searchPattern. Use the searchPattern parameter to filter the results so that only the objects with the specified string in the object name are included in the response. To use the searchPattern parameter, use the following URI:

```
/api/v2/connection/<source or target>/<id>?searchPattern=<pattern>
```

For example, the following request returns source objects that include "abc" in the object name:

```
/api/v2/connection/source/002D420000000J?searchPattern=abc
```

- maxRecordsCount. Use the maxRecordsCount parameter to set the maximum number of objects to return. To use the maxRecordsCount, use the following URI:

```
/api/v2/connection/<source or target>/<id>?maxRecordsCount=<max number of objects>
```

For example, the following request returns a maximum of 5000 source objects:

```
/api/v2/connection/source/002D420000000J?maxRecordsCount=5000
```

If you don't include the maxRecordsCount parameter, a maximum of 200 objects can be returned for the request.

List of connections of a specified type associated with a Secure Agent or runtime environment

To request a list of connections by Secure Agent ID and connection type, use the following URI:

```
/api/v2/connection/search?agentId=<agentId>&uiType=<uiType>
```

To request a list of connections by runtime environment ID and connection type, use the following URI:

```
/api/v2/connection/search?runtimeEnvironmentId=<runtimeEnvironmentId>&uiType=<uiType>
```

If you pass both agentId and runtimeEnvironmentId, the service uses runtimeEnvironmentId and ignores agentId. If you pass only agentId, the service translates agentId into its corresponding runtimeEnvironmentId before it saves the resource to the repository.

Metadata details for a specified connection

To request metadata details for a specified connection, use the following URI:

```
/api/v2/connection/source/<connection ID>/metadata
```
```
/api/v2/connection/target/<connection ID>/metadata
```

The metadata is returned in the runtimeAttribute object which contains the following attributes:

- name
- dataType
- defaultValue
- label
- mandatory
- maxLength
- sessionVarAllowed
- possibleValues

Use the following connection request URI attributes:

Field	Type	Required	Description
agentId	String	-	Secure Agent ID
runtimeEnvironmentId	String	-	Runtime environment ID.
uiType	String	Yes	Connection type. Use one of the following options: - CSVFile. CSV flat file. - FTP. - MS_ACCESS. - MSD. Microsoft Dynamics CRM. - MySQL. - ODBC. - Oracle. - OCOD. Oracle CRM On Demand. - Salesforce. - SFTP. Secure FTP. - SAP_ALE_IDoc_Reader. SAP IDoc Reader. - SAP_ALE_IDoc_Writer. SAP IDoc Writer. - SqlServer. Microsoft SQL Server 2000. - SqlServer2005. Microsoft SQL Server 2005. - SqlServer2008. Microsoft SQL Server 2008. - SqlServer2012. Microsoft SQL Server 2012. - SqlServer2014. Microsoft SQL Server 2014. - SqlServer2016. Microsoft SQL Server 2016. - SqlServer2017. Microsoft SQL Server 2017. - TOOLKIT. Informatica Cloud Connector. Also use for NetSuite connections. - WebServicesConsumer. Web Service.

Details GET Response

Returns the connection object for the requested connection ID. If you request information for all connections in the organization, returns a connection object for each connection in the organization.

If you request a list of connections based on the runtime environment ID and connection type, returns a connection object for each connection that matches the requirements.

If you request a list of source or target objects available for the requested connection ID, returns the connListItem object for each available object.

Returns the error object if errors occur.

The connection object includes different information based on connection type.

The following table describes attributes included in a connection object:

Field	Type	Description
id	String	Connection ID.
orgId	String	Organization ID.
name	String	Connection name.
description	String	Description of the connection.

Field	Type	Description
createTime	Date/time	Time the connection was created.
updateTime	Date/time	Last time the connection was updated.
createdBy	String	User who created the connection.
updatedBy	String	User who last updated the connection.
agentId	String	Secure Agent ID for Flat File, FTP/SFTP, Microsoft SQL Server, MS Access, MySQL, ODBC, Oracle, and Web Service connections.
runtimeEnvironmentId	String	Runtime environment used by the connection. This is the Runtime Environment field in the user interface. In the response returned to the user interface, this attribute is named agentGroupId.
instanceName	String	Microsoft SQL Server instance name.
host	String	Host name for FTP/SFTP, Microsoft SQL Server, MySQL, and Oracle connections.
domain	String	Domain name for Microsoft Dynamics CRM connections that use IFD or Active Directory authentication, and Web Service connections.
dateFormat		Date format for Flat File, FTP, and SFTP connections.
database	String	Returns the following information: - For Microsoft SQL Server and MySQL connections, returns the database name. - For Flat File connections, returns the directory. - For FTP and SFTP connections, returns the local directory. - For MS Access and ODBC connections, returns the data source name. - For Oracle connections, returns the service name. - For SAP IDoc Writer and Reader connections, returns the destination entry. - For Web Service connections, returns the service URL.
codepage		Code page for Flat File, FTP, SFTP, Microsoft SQL Server, MySQL, MS Access, ODBC, Oracle, and SAP.
clientCode	String	Client code for SAP IDoc Writer connections.
authenticationType	String	Authentication type for Microsoft Dynamics CRM, Microsoft SQL Server, and Web Service connections.
adjustedJdbcHostName	String	Host name. Or host and instance name for Microsoft SQL Server connections.
accountNumber	String	Account ID for NetSuite connections.
languageCode	String	Language code for SAP IDoc Writer connections.
remoteDirectory	String	Remote directory for FTP/SFTP connections.
schema	String	Schema name for Microsoft SQL Server, ODBC, Oracle, and Web Service connections.

Field	Type	Description
serviceUrl	String	Service URL for Microsoft Dynamics CRM, Oracle CRM On Demand, and Salesforce connections.
shortDescription	String	The first 50 letters of the description.
type	String	Connection type returns one of the following responses: - CSVFile. CSV flat file. - FTP. - MS_ACCESS. - MSD. Microsoft Dynamics CRM. - MySQL. - ODBC. - Oracle. - OCOD. Oracle CRM On Demand. - Salesforce. - SFTP. Secure FTP. - SAP_ALE_IDoc_Reader. SAP IDoc Reader. - SAP_ALE_IDoc_Writer. SAP IDoc Writer. - SqlServer. Microsoft SQL Server 2000. - SqlServer2005. Microsoft SQL Server 2005. - SqlServer2008. Microsoft SQL Server 2008. - SqlServer2012. Microsoft SQL Server 2012. - SqlServer2014. Microsoft SQL Server 2014. - SqlServer2016. Microsoft SQL Server 2016. - TOOLKIT. Informatica Cloud Connector. - WebServicesConsumer. Web Service. **Note:** The user interface field name on the Connections page varies depending on the connection. For example, for SQL Server, the user interface field name is SQL Server Version. Also note that for SQL Server, the REST API attribute that populates the value in the user interface is named subType.
port	Int	Port number for FTP/SFTP, Microsoft SQL Server, MySQL, and Oracle connections.
password	String	Password for the connection.
username	String	User name for the connection.
securityToken	String	Security token for a Salesforce connection.
stsUrl	String	Security token service URL for Microsoft Dynamics CRM connections that use Active Directory authentication.
organizationName	String	Organization name for Microsoft Dynamics CRM connections.
timeout	Int	Timeout for Web Service connections.
trustCertificatesFile	String	Trust certificates file name for Web Service connections.
certificateFile	String	Certificates file name for Web Service connections.
certificateFilePassword	String	Certificates file password for Web Service connections.
certificateFileType	String	Certificates file type for Web Service connections.
privateKeyFile	String	Private key file name for Web Service connections.

Field	Type	Description
privateKeyPassword	String	Private key password for Web Service connections.
privateKeyFileType	String	Private key file type for Web Service connections.
connParams	String	Parameters used in the connection. Includes connection attributes in the connParam object for SAP, NetSuite, Oracle CRM On Demand, ODBC, or other Informatica Cloud connectors.
federatedId	String	Global unique identifier for the connection.
internal	Boolean	Indicates whether the connection was created internally for data preview purposes only.
retryNetworkError	Boolean	Used for internal purposes only.
supportsCCIMultiGroup	Boolean	Used for internal purposes only.
connListItem		List of connections included in the connListItem object.
id	String	Included in the connListItem object. Source or target ID.
name	String	Included in the connListItem object. Source or target name.

Test GET Request

To test a connection, use the connection ID in the following URI:

```
/api/v2/connection/test/<id>
```

Test GET Response

Returns the success object if the test succeeds.

Returns the error object if errors occur.

POST Request

You can create or update connections. To update a connection, use the connection ID with the following URI. To create a connection, omit the optional connection ID.

```
/api/v2/connection/<id>
```

You can submit a partial update using partial mode. To submit a request using partial mode, use a JSON request and include the following line in the header:

```
Update-Mode=PARTIAL
```

In a connection POST request, use the additional attributes in the connection object. The attributes used by Informatica Cloud Connector connections vary by connection type.

To create or update an Informatica Cloud Connector connection, consult the Informatica Cloud application for the attributes used by the connection. Enclose any attributes that are not listed in the following tables in a connParam object.

To get a list of connectors that are available to the organization and attribute information for a specific connector type, see <u>"connector" on page 289</u>.

For more information about attributes and data types used for creating connections through the REST API, see "Connection user interface fields to REST API attributes mapping" on page 435 and "Connector data types" on page 434.

POST Response

If successful, returns the connection object for the connection that was created or updated.

Returns the error object if errors occur.

DELETE Request

To delete a connection, use the connection ID in the following URI.

```
/api/v2/connection/<id>
```

DELETE Response

Returns the 200 response code if the request is successful.

Returns the error object if errors occur.

POST Example

To update an SAP Table connection, you might use the following request, enclosing SAP attributes in the connParam object:

```
POST <serverUrl>/api/v2/connection/0002D420000000J
Content-Type: application/xml
Accept: application/xml
icSessionId: <icSessionId>

<connection>
  <id>0002D420000000J</id>
  <orgId>00342000</orgId>
  <name>test dir</name>
  <type>TOOLKIT</type>
  <agentId>00001Y08000000000002</agentId>
  <username>username</username>
  <password>password</password>
  <instanceName>SAPTableConnector</instanceName>
  <connParams>
     <agentId>00001Y08000000000002</agentId>
     <username>username</username>
     <password>password</password>
     <client>800</client>
     <language>EN</language>
     <Saprfc Ini Path>C:\\Windows\\SysWOW64</Saprfc Ini Path>
     <Destination>GE6</Destination>
  </connParams>
  <runtimeEnvironmentId>00000C25000000000002</runtimeEnvironmentId>
</connection>
```

A successful request returns the connection object that you updated.

CSV Flat File Connections

When you create or update a CSV flat file connection, you can configure additional attributes, such as the connection ID and the connection name.

The following table describes attributes that you can use for CSV flat file connections:

Attribute	Description
id	Connection ID.
orgId	Organization ID.
name	Connection name.
description	Optional connection description.
type	Connection type. Use CSVFile.
database	Directory where flat files are stored. In the user interface, this attribute is the Directory field. In the REST API response that populates the value in the user interface, the name of this attribute is dirName.

Attribute	Description
dateFormat	Date format for date fields in the flat file. Use one of the following formats: - MM/dd/yyyy - MM-dd-yyyy - MM.dd.yyyy - dd/MM/yyyy - dd-MM-yyyy - dd.MM.yyyy - MM/dd/yyyy HH:mm - MM-dd-yyyy HH:mm - MM.dd.yyyy HH:mm - dd/MM/yyyy HH:mm - dd-MM-yyyy HH:mm - dd.MM.yyyy HH:mm - MM/dd/yyyy HH:mm:ss - MM-dd-yyyy HH:mm:ss - MM.dd.yyyy HH:mm:ss - dd/MM/yyyy HH:mm:ss - dd-MM-yyyy HH:mm:ss - dd.MM.yyyy HH:mm:ss - yyyy-MM-dd - yyyy-MM-dd HH:mm - yyyy-MM-dd HH:mm:ss - yyyy-MM-ddTHH:mm:ss.SSSZ - yyyy/MM/dd - yyyy/MM/dd HH:mm - yyyy/MM/dd HH:mm:ss - yyyyMMdd - MM/dd/yyyy HH24:mm:ss - MM/dd/yyyy HH24:mm:ss.ms - MM/dd/yyyy HH24:mm:ss.us - MM/dd/yyyy HH24:mm:ss.ns Default is MM/dd/yyyy HH:mm:ss.
codepage	The code page of the system that hosts the flat file. Use one of the following options: - UTF-8. Unicode Transformation Format, multibyte. - MS1252. MS Windows Latin 1 (ANSI), superset of Latin 1. - ISO-8859-15. Latin 9, Western European. - ISO-8859-2. Eastern European. - ISO-8859-3. Southeast European. - ISO-8859-5. Cyrillic. - ISO-8859-9. Latin 5, Turkish. - IBM500. IBM EBCDIC International Latin-1.

FTP and SFTP Connections

When you create or update an FTP or SFTP connection, you can configure additional attributes, such as the connection ID and the connection name.

The following table describes attributes that you can use for FTP or SFTP file connections:

Attribute	Description
id	Connection ID.
orgId	Organization ID.

Attribute	Description
name	Connection name.
description	Optional connection description.
type	Connection type. Use FTP or SFTP.
username	User name.
password	Password.
host	Name of the machine hosting the database server or FTP/SFTP host. For a FTP/SFTP connection, enter the host name of IP address.
port	Network port number used to connect to FTP/SFTP connection. Default port is 21 for FTP and 22 for SFTP.
database	Directory on a local machine that stores the local file. In the user interface, this attribute is the Directory field. In the REST API response that populates the value in the user interface, the name of this attribute is dirName. The local machine must also run the Secure Agent used to run the corresponding task. Enter a local directory or use the Browse button to select a local directory.
remoteDirectory	Directory on the FTP/SFTP host that stores the remote flat file. Depending on the FTP/SFTP server, you may have limited options to enter directions. For more information, see the FTP/SFTP server documentation.
dateFormat	Date format for date fields in the flat file. Use one of the following formats: - MM/dd/yyyy - MM-dd-yyyy - MM.dd.yyyy - dd/MM/yyyy - dd-MM-yyyy - dd.MM.yyyy - MM/dd/yyyy HH:mm - MM-dd-yyyy HH:mm - MM.dd.yyyy HH:mm - dd/MM/yyyy HH:mm - dd-MM-yyyy HH:mm - dd.MM.yyyy HH:mm - MM/dd/yyyy HH:mm:ss - MM-dd-yyyy HH:mm:ss - MM.dd.yyyy HH:mm:ss - dd/MM/yyyy HH:mm:ss - dd-MM-yyyy HH:mm:ss - dd.MM.yyyy HH:mm:ss - yyyy-MM-dd - yyyy-MM-dd HH:mm - yyyy-MM-dd HH:mm:ss - yyyy-MM-ddTHH:mm:ss.SSSZ

Attribute	Description
codepage	The code page of the system that hosts the flat file. Use one of the following options: - UTF-8. Unicode Transformation Format, multibyte. - MS1252. MS Windows Latin 1 (ANSI), superset of Latin 1. - ISO-8859-15. Latin 9, Western European. - ISO-8859-2. Eastern European. - ISO-8859-3. Southeast European. - ISO-8859-5. Cyrillic. - ISO-8859-9. Latin 5, Turkish. - IBM500. IBM EBCDIC International Latin-1.
agentId	Secure Agent ID.

Microsoft Access Connections

When you create or update a Microsoft Access connection, you can configure additional attributes, such as the connection ID and the connection name.

The following table describes attributes that you can use for Microsoft Access connections:

Attribute	Description
id	Connection ID.
orgId	Organization ID.
name	Connection name.
description	Optional connection description.
type	Connection type. Use MS_ACCESS.
database	Data source name. In the user interface, this is the Data Source Name field.
codepage	The code page compatible with the MS Access database. Use one of the following options: - UTF-8. Unicode Transformation Format, multibyte. - MS1252. MS Windows Latin 1 (ANSI), superset of Latin 1. - ISO-8859-15. Latin 9, Western European. - ISO-8859-2. Eastern European. - ISO-8859-3. Southeast European. - ISO-8859-5. Cyrillic. - ISO-8859-9. Latin 5, Turkish. - IBM500. IBM EBCDIC International Latin-1.
agentId	Secure Agent ID.

Microsoft Dynamics CRM Connections

When you create or update a Microsoft Dynamics CRM connection, you can configure additional attributes, such as the connection ID and the connection name.

The following table describes attributes that you can use for Microsoft Dynamics CRM connections:

Attribute	Description
id	Connection ID.
orgId	Organization ID.
name	Connection name.
description	Optional connection description.
type	Connection type. Use MSD.
authenticationType	Authentication type for the connection. Select a valid authentication type. Use one of the following authentication types: - LIVE. Microsoft Live. Use for synchronization tasks or PowerCenter tasks. - IFD. Internet Facing Development (IFD). Use for synchronization tasks or PowerCenter tasks. - AD. Active Directory. Use for PowerCenter tasks only.
username	Microsoft Dynamics CRM user name.
password	Microsoft Dynamics CRM password.
organizationName	Microsoft Dynamics CRM organization name.
domain	Microsoft Dynamics CRM domain name. Required for IFD and Active Directory authetication.
serviceURL	URL of the Microsoft Dynamics CRM service. For Microsoft Live authentication, use the following format: https:// <orgname>.crm.dynamics.com For IFD authentication, use the following format: https://<server.company.com>:<port> For Active Directory, use the following format: http://<server.company.com>:<port>
stsURL	Microsoft Dynamics CRM security token service URL. For example, https:// sts1.company.com. Required for IFD authentication.
agentId	Secure Agent ID. Required for Active Directory authentication only.

Microsoft SQL Server Connections

When you create or update a Microsoft SQL Server connection, you can configure additional attributes, such as the connection ID and the connection name.

The following table describes attributes that you can use for Microsoft SQL Server connections:

Attribute	Description
id	Connection ID.
orgId	Organization ID.
name	Connection name.
description	Optional connection description.
type	Connection type. Use one of the following codes: - SqlServer. Microsoft SQL Server 2000. - SqlServer2005. Microsoft SQL Server 2005. - SqlServer2008. Microsoft SQL Server 2008. - SqlServer2012. Microsoft SQL Server 2012. In the user interface, this attribute is the SQL Server Version field. In the REST API response that populates the value in the user interface, the name of this attribute is subType.
authenticationType	Authentication method for the connection. Use one of the following options: - Windows. Use Microsoft Windows authentication to access Microsoft SQL Server. Available when users access Data Integration in Windows. - SqlServer. Use Microsoft SQL Server authentication to access Microsoft SQL Server.
username	User name for the database login. Use when authenticationType is SqlServer.
password	Password for the database login. Use when authenticationType is SqlServer.
host	Name of the machine hosting the database server.
port	Network port number used to connect to the database server. Default port number is 1433.
instanceName	Instance name of the Microsoft SQL Server database.
database	Database name for the Microsoft SQL Server target. Database name is case sensitive if the database is case sensitive. Maximum length is 100 characters. Database names can include alphanumeric and underscore characters.
schema	Schema used for the target connection.
codepage	The code page of the Microsoft SQL Server database. Use one of the following options: - UTF-8. Unicode Transformation Format, multibyte. - MS1252. MS Windows Latin 1 (ANSI), superset of Latin 1. - ISO-8859-15. Latin 9, Western European. - ISO-8859-2. Eastern European. - ISO-8859-3. Southeast European. - ISO-8859-5. Cyrillic. - ISO-8859-9. Latin 5, Turkish. - IBM500. IBM EBCDIC International Latin-1.
agentId	Secure Agent ID.

MySQL Connections

When you create or update a MySQL connection, you can configure additional attributes, such as the connection ID and the connection name.

The following table describes attributes that you can use for MySQL connections:

Attribute	Description
id	Connection ID.
orgId	Organization ID.
name	Connection name.
description	Optional connection description.
type	Connection type. Use MySQL.
username	User name for the database login.
password	Password for the database login.
host	Name of the machine hosting the database server.
port	Network port number used to connect to the database server. Default is 3306.
database	Database name for the MySQL database target. Database name is case sensitive if the database is case sensitive.
codepage	The code page for the database server. Use one of the following options: - UTF-8. Unicode Transformation Format, multibyte. - MS1252. MS Windows Latin 1 (ANSI), superset of Latin 1. - ISO-8859-15. Latin 9, Western European. - ISO-8859-2. Eastern European. - ISO-8859-3. Southeast European. - ISO-8859-5. Cyrillic. - ISO-8859-9. Latin 5, Turkish. - IBM500. IBM EBCDIC International Latin-1.
agentId	Secure Agent ID.

NetSuite Connections

When you create or update a NetSuite connection, you can configure additional attributes, such as the connection ID and the connection name.

The following table describes attributes that you can use for NetSuite connections:

Attribute	Description
id	Connection ID.
orgId	Organization ID.
name	Connection name.

Attribute	Description
description	Optional connection description.
type	Connection type. Use NetSuite.
username	NetSuite user name.
password	NetSuite password.
accountNumber	NetSuite account ID. To locate your account ID, log in to NetSuite and navigate to Setup > Integration > Web Services Preferences.
serviceURL	WSDL URL. If your NetSuite account does not use the default NetSuite WSDL URL, enter the WSDL URL used by your NetSuite account.

ODBC Connections

When you create or update an ODBC connection, you can configure additional attributes, such as the connection ID and the connection name.

The following table describes attributes that you can use for OBDC connections:

Attribute	Description
id	Connection ID.
orgId	Organization ID.
name	Connection name.
description	Optional connection description.
type	Connection type. Use ODBC.
username	User name for the database login.
password	Password for the database login.
database	Data source name.
schema	Schema used for the target connection. Use uppercase letters when you specify the schema name for an Oracle database. Required to connect to an IBM DB2 database.
codepage	The code page of the database server or flat file defined in the connection. Use one of the following options: - UTF-8. Unicode Transformation Format, multibyte. - MS1252. MS Windows Latin 1 (ANSI), superset of Latin 1. - ISO-8859-15. Latin 9, Western European. - ISO-8859-2. Eastern European. - ISO-8859-3. Southeast European. - ISO-8859-5. Cyrillic. - ISO-8859-9. Latin 5, Turkish. - IBM500. IBM EBCDIC International Latin-1.

Attribute	Description
agentId	Secure Agent ID.
ODBC_SUBTYPE	The ODBC connection subtype to connect to a specific database. Use one of the following options: - Azure DW - DB2 - Google BigQuery - PostgreSQL - Redshift - SAP IQ - Snowflake - Teradata - Other
authenticationType	The authentication method to connect to a database. Required for DB2. Use one of the following options: - Database - Kerberos
kerberosProperties	Additional connection properties to use Kerberos authentication to connect to DB2. If you specify more than one property, separate each key-value pair with a semicolon.

Oracle Connections

When you create or update an Oracle connection, you can configure additional attributes, such as the connection ID and the connection name.

The following table describes attributes that you can use for Oracle connections:

Attribute	Description
id	Connection ID.
orgId	Organization ID.
name	Connection name.
description	Optional connection description.
type	Connection type. Use Oracle.
username	User name for the database login.
password	Password for the database login.
host	Name of the machine hosting the database server.
port	Network port number used to connect to the database server. Default is 1521.
database	Service name that uniquely identifies the Oracle database. This attribute is the Service Name field in the user interface. If the connection fails, contact the database administrator.

Attribute	Description
schema	Schema used for the target connection. Optional.
codepage	The code page of the database server. Use one of the following options: - UTF-8. Unicode Transformation Format, multibyte. - MS1252. MS Windows Latin 1 (ANSI), superset of Latin 1. - ISO-8859-15. Latin 9, Western European. - ISO-8859-2. Eastern European. - ISO-8859-3. Southeast European. - ISO-8859-5. Cyrillic. - ISO-8859-9. Latin 5, Turkish. - IBM500. IBM EBCDIC International Latin-1.
agentId	Secure Agent that Data Integration uses to access the database in the local area network.

Oracle CRM On Demand Connections

When you create or update an Oracle CRM On Demand connection, you can configure additional attributes, such as the connection ID and the connection name.

The following tables describes attributes that you can use for Oracle CRM On Demand connections:

Attribute	Description
id	Connection ID.
orgId	Organization ID.
name	Connection name.
description	Optional connection description.
type	Connection type. Use OCOD.
username	Oracle CRM On Demand user name. Use the following format: <domain>/<user name> For example: domain/jsmith@companyname.com.
password	Oracle CRM On Demand password.
serviceUrl	URL of the Oracle CRM On Demand service. For example: https://securecompany.crmondemand.com.

Salesforce Connections

When you create or update a Salesforce connection, you can configure additional attributes, such as the connection ID and the connection name.

The following table describes attributes that you can use for Salesforce connections:

Attribute	Description
id	Connection ID.
orgId	Organization ID.
name	Connection name.
description	Optional connection description.
type	Connection type. Use Salesforce.
username	User name for the Salesforce account. <domain>/<user name> For example: domain/jsmith@companyname.com.
password	Password for the Salesforce account.
securityToken	Security token associated with the user name and password. Optional.
serviceUrl	URL of the Salesforce service. Maximum length is 100 characters.

SAP IDoc Reader Connections

When you create or update an SAP IDoc Reader connection, you can configure additional attributes, such as the connection ID and the connection name.

The following table describes attributes that you can use for SAP IDoc Reader connections:

Attribute	Description
id	Connection ID.
orgId	Organization ID.
name	Connection name.
description	Optional connection description.
type	Connection type. Use SAP_ALE_IDoc_Reader.
username	SAP user name with authorization on S_DATASET, S_TABU_DIS, S_PROGRAM, and B_BTCH_JOB objects.
password	Password for the SAP user name.
database	Type A DEST entry in the saprfc.ini file. This attribute is the Destination Entry field in the user interface.

Attribute	Description
codepage	The code page compatible with the SAP source. Use one of the following options: - UTF-8. Unicode Transformation Format, multibyte. - MS1252. MS Windows Latin 1 (ANSI), superset of Latin 1. - ISO-8859-15. Latin 9, Western European. - ISO-8859-2. Eastern European. - ISO-8859-3. Southeast European. - ISO-8859-5. Cyrillic. - ISO-8859-9. Latin 5, Turkish. - IBM500. IBM EBCDIC International Latin-1.
agentId	Secure Agent ID.

SAP IDoc Writer Connections

When you create or update an SAP IDoc Writer connection, you can configure additional attributes, such as the connection ID and the connection name.

The following table describes attributes that you can use for SAP IDoc Writer connections:

Attribute	Description
id	Connection ID.
orgId	Organization ID.
name	Connection name.
description	Optional connection description.
type	Connection type. Use SAP_ALE_IDoc_Writer.
username	SAP user name with authorization on S_DATASET, S_TABU_DIS, S_PROGRAM, and B_BTCH_JOB objects.
password	Password for the SAP user name.
database	Type A DEST entry in the saprfc.ini file. This attribute is the Connection String field in the user interface.
languageCode	Language code that corresponds to the SAP language. A two-letter code, such as en for English.
clientCode	SAP client number. A three-letter code.
codepage	The code page compatible with the SAP target. Use one of the following options: - UTF-8. Unicode Transformation Format, multibyte. - MS1252. MS Windows Latin 1 (ANSI), superset of Latin 1. - ISO-8859-15. Latin 9, Western European. - ISO-8859-2. Eastern European. - ISO-8859-3. Southeast European. - ISO-8859-5. Cyrillic. - ISO-8859-9. Latin 5, Turkish. - IBM500. IBM EBCDIC International Latin-1.
agentId	Secure Agent ID.

Web Service Connections

When you create or update a Web Service connection, you can configure additional attributes, such as the connection ID and the connection name.

The following table describes attributes that you can use for Web Service connections:

Attribute	Description
id	Connection ID.
orgId	Organization ID.
name	Connection name.
description	Optional connection description.
type	Connection type. Use WebServicesConsumer.
username	SAP user name with authorization on S_DATASET, S_TABU_DIS, S_PROGRAM, and B_BTCH_JOB objects.
password	Password for the web service login. If the web service does not require a user name, leave this field empty. Optional.
domain	Domain for authentication. Optional.
serviceUrl	Endpoint URL for the web service that you want to access. The WSDL file specifies this URL in the location element. This attribute is the Endpoint URL field in the user interface. Optional.
timeout	Secure Agent ID. Number of seconds Informatica Intelligent Cloud Services waits for a connection to the web service provider before it closes the connection and fails the session. Also, the number of seconds the Informatica Intelligent Cloud Services waits for a SOAP response after sending a SOAP request before it fails the session. Default is 60. Optional.
trustCertificatesFile	File containing the bundle of trusted certificates that Informatica Intelligent Cloud Services uses when authenticating the SSL certificate of the web services provider. Default is ca-bundle.crt. Optional.
certificateFile	Client certificate that a web service provider uses when authenticating a client. You specify the client certificate file if the web service provider needs to authenticate Informatica Intelligent Cloud Services. Optional.
certificateFilePassword	Password for the client certificate. You specify the certificate file password if the web service provider needs to authenticate Informatica Intelligent Cloud Services. Optional.
certificateFileType	File type of the client certificate. You specify the certificate file type if the web service provider needs to authenticate the Integration Service. Use one of the following codes: - PEM - DER Optional.
privateKeyFile	Private key file for the client certificate. You specify the private key file if the web service provider needs to authenticate Informatica Intelligent Cloud Services. Optional.

Attribute	Description
privateKeyPassword	Password for the private key of the client certificate. You specify the key password if the web service provider needs to authenticate Informatica Intelligent Cloud Services. Optional.
privateKeyFileType	File type of the private key of the client certificate. You specify the key file type if the web service provider needs to authenticate Informatica Intelligent Cloud Services. If necessary, use PEM. Optional.
authenticationType	Authentication type to use when the web service provider does not return an authentication type to Informatica Intelligent Cloud Services. Use one of the following options: - Auto. The Integration Service attempts to determine the authentication type of the web service provider. - Basic. Based on a non-encrypted user name and password. - Digest. Based on an encrypted user name and password. - NTLM. Based on encrypted user name, password, and domain. Default is Auto. Optional.
agentId	ID for the Secure Agent that Informatica Intelligent Cloud Services uses to access the database in the local area network.

connectionMigration

Use this resource to migrate an old version of the connection in referenced assets to the latest connection version within an organization. The migration creates a copy of the project folder and assets and updates the referenced connection in the assets with the latest version.

You can migrate old connections used in Source, Target, and Lookup transformations from assets such as mappings and tasks. If the connection is used across assets in multiple projects, all the referenced assets are updated to use the current connection version.

POST request

To specify the connections to migrate and start the migrate job, use the following URI:

```
saas/api/v2/connectionMigration/migrate
```

Include the following fields in the request:

Field	Type	Required	Description
icSessionId	String	Yes	Informatica Intelligent Cloud Services session ID.
sourceConn	String	Yes	The name of the source connection that you want to migrate. This is the old version of the connector.

Field	Type	Required	Description
targetConn	String	Yes	The target connection name to which you want to migrate. This is the current version of the connector.
projectName	String	No	The project name contains the referenced assets of the source connection that you want to migrate. If you do not specify the project name, the source connection in all assets are migrated.

POST request example

You can use a request similar to the following example:

```
POST <baseURL>/saas/api/v2/connectionMigration/migrate
Content-Type: application/json
Accept: application/json
icSessionId: <sessionId>

{
  "sourceConn": "Migration_V1",
  "targetConn": "Migration_V2",
  "projectName": "V1_Assets"
}
```

POST response

If successful, returns the following information for the export job:

Field	Type	Description
id	String	Global unique identifier for the migration job.
name	String	Name of the migration job.
count	Integer	Number of dependent objects.
status	Complex type	Status of the migration.
objects	Collection	Objects in the migration job.
name	String	Returned in the objects object. Name of the asset that uses the connector.
type	String	Returned in the objects object. The type of asset.
sourcePath	String	Returned in the objects object. Complete path of the referenced asset in the old connection.
targetPath	String	Returned in the objects object. Complete path of the referenced assets in the current connection. The target folder uses the same name as the source folder, with "_Migration_<migration job ID>" appended to the name for identification.
status	String	Returned in the objects object. State of the connection updated in the asset, such as IN PROGRESS, SUCCESS, or FAILED.

POST response examples

If successful, you might receive a response similar to the following example:

```
{
    "id": "4l1XcURpYFjjxpopb7AJ0L",
    "name": "job-1661702962208",
    "count": "1",
    "status": "SUCCESSFUL",
    "objects": [
        {
            "name": "Mapping_V1",
            "type": "DTEMPLATE",
            "sourcePath": "/AA_Migration_Doc",
            "targetPath": "/AA_Migration_Doc_Migration_4l1XcURpYFjjxpopb7AJ0L",
            "status": "SUCCESSFUL"
        }
    ]
}
```

If you receive an error, you might see a response similar to the following example:

```
{
  "error": {
    "code": "MigrationSvc_034",
    "message": "Invalid object id/s [[242973wgfscbwasd23]]. Object resolution failed.",
    "requestId": "2ataXVlgw3ydI1Yb2MA4sq"
  }
}
```

For more information about using the connectionMigration resource to migrate the connectors to the latest version and for the list of connectors that support migration, see the following How-To Library article:
[Migrating a connector from previous versions using the Data Integration REST API](#)

connector

Use this resource to request a list of connectors that are available to an organization along with connector details. You can also use this resource to get attribute information for a specific connector type. You can use the list of attributes that this resource provides when you create a connection for a specific connector type since you need to provide these attributes when you create a connection of a certain type.

GET Request and Response for Available Connectors

To request a list of connectors available for an organization, submit a GET request using the following URI:

```
/api/v2/connector
```

For example, you might use the following request:

```
GET <serverUrl>/api/v2/connector
Content-Type: application/xml
Accept: application/xml
icSessionId: <icSessionId>
```

A successful response returns the following attributes in the connector object:

Field	Type	Description
name	String	Name of the connector.
type	String	Type of connector. Includes the following values: - Salesforce - Oracle - SqlServer - MySQL - CSVFile - ODBC - MS_ACCESS - FTP - SAP - WebServicesConsumer - MSD
publisher	String	Name of the entity that published the connector.
connectorVersion	Int	Connector version.
shortName	String	Connector short name.
isPublic	Boolean	Whether the connector is a public or private connector. If you are interested in a connector that is private, contact Informatica Global Customer Support.

GET Request and Response for Connector Metadata

To get metadata for a specific connector type, submit a GET request using the following URI:

```
/api/v2/connector/metadata?connectorName=<connectorName>
```

For example, you might use the following request:

```
GET <serverUrl>/api/v2/connector/metadata?connectorName=SQLServer
Content-Type: application/xml
Accept: application/xml
icSessionId: <icSessionId>
```

A successful response returns the following attributes in the connectorMetadata object:

Field	Type	Description
type	String	Type of connector, such as Salesforce or Oracle.
isStandardConnType	Boolean	Whether the connector is standard or custom. A "True" value indicates the connector is standard.
attributes		Connector attributes for the specified connector type. Includes information in the attribute object for each connector object.
name	String	Included in the attribute object. Name of the attribute, such as database or codePage.
label	String	Included in the attribute object. Label.

Field	Type	Description
id	String	Included in the attribute object. ID.
value	String	Included in the attribute object. Value of the attribute.
type	Int	Included in the attribute object. Data type. For values, see "Connector data types" on page 434.
isMandatory	Boolean	Included in the attribute object. Whether the attribute is mandatory.
visible	Boolean	Included in the attribute object. Whether the attribute is visible.
list	String	Included in the attribute object. A list of types for the selected connector type. For example, SQL Server includes the types SqlServer2000, SqlServer2005, SqlServer2008, and so on.

customFunc

Use this resource to request the details of a PowerCenter mapplet or to request a list of all PowerCenter mapplets in the organization. You can also use this resource to upload or delete a PowerCenter mapplet.

Note: To request a list of all mapplets in the organization, use the objects resource. For more information, see "Finding an asset" on page 163.

GET request

To request a list of all PowerCenter mapplets in the organization, use the following URI:

```
/api/v2/customFunc
```

To request the details of a single mapplet, you can use the mapplet ID or mapplet name in the request. Use one of the following URIs:

```
/api/v2/customFunc/<id>
```

```
/api/v2/customFunc/name/<name>
```

If you use the mapplet name and the mapplet name includes a space, replace the space with `%20`. For example:

```
/api/v2/customFunc/name/my%20mapplet
```

GET response

If the request for a list of mapplets is successful, returns the customFunc object for every PowerCenter mapplet in the organization without the input, output, and connection details.

If the request for the details of a single mapplet is successful, returns the customFunc object.

Returns the error object if errors occur.

The customFunc object includes the following attributes:

Field	Type	Description
id	String	Mapplet ID.
orgId	String	Organization ID.
name	String	Mapplet name.
description	String	Mapplet description.
createTime	Date/time	Time the mapplet was created.
updateTime	Date/time	Time the mapplet was last updated.
createdBy	String	User who created the mapplet.
updatedBy	String	User who last updated the mapplet.
mappletName	String	Name of the Mapplet transformation used in the mapplet.
active	Boolean	Whether the mapplet is active. Returns true or false.
mappletXmlFile	String	The mapplet XML file.

inputs	String	Input fields for the mapplet. Includes the following information for each field in the field object:
		- id
		- name
		- type
		- label
		- parentObject
		- precision
		- pcType
		- scale
		- columnIndex
		- isKey
		- isExternalId
		- isNullable
		- isUnique
		- isCreateable
		- isCalculated
		- isUpdateable
		- isFilterable
		- linkedFields
		- relatedInfos. Includes the following information in the fieldRelatedInfo object:
		  - id
		  - referenceObject
		  - relationshipName
		- javaType
		- showLabel
		- naturalOrder
		- customProperties

Field	Type	Description

outputs	String	Output fields for the mapplet. Includes the following information for each field in the field object:
		- id
		- name
		- type
		- label
		- parentObject
		- precision
		- pcType
		- scale
		- columnIndex
		- isKey
		- isExternalId
		- isNullable
		- isUnique
		- isCreateable
		- isCalculated
		- isUpdateable
		- isFilterable
		- linkedFields
		- relatedInfos. Includes the following information in the fieldRelatedInfo object:
		  - id
		  - referenceObject
		  - relationshipName
		- javaType
		- showLabel
		- naturalOrder
		- customProperties

connections		Connection information for the mapplet. Includes a pcsConnection object for each connection.
id	Long	Included in the pcsConnection object.
name	String	Included in the pcsConnection object. Connection name.
type	String	Included in the pcsConnection object. Connection type.
subtype	String	Included in the pcsConnection object. Connection subtype.
description	String	Included in the pcsConnection object. Description of the connection.
connectionId	String	Included in the pcsConnection object. Connection ID.

POST request

To upload a new PowerCenter mapplet, use the following URI:

```
/api/v2/customFunc
```

If you want to specify a location for the mapplet, include the container ID in the request. If the container ID isn't included in the request, the mapplet is created in the Default folder. You can find the container ID for a project or folder in the Data Integration user interface. On the **Explore** page, select the folder. In the URL, the last string of characters is the container ID.

For example, in the following URL, the container ID is dH2DuGJYda7ijgW4Sm32sR:

```
https://na1.dm-us.informaticacloud.com/diUI/products/integrationDesign/main/Explore/
dH2DuGJYda7ijgW4Sm32sR
```

To update an existing PowerCenter mapplet, include the mapplet ID in the following URI:

```
/api/v2/customFunc/<id>
```

Note: Encode the request body as multipart/form-data.

With this URI, you can use the following attributes in the request body:

Field	Type	Required	Description
file	String	Yes	The mapplet XML file exported from Informatica PowerCenter. File content should be in binary format, UTF-8 encoding.
name	String	Yes	The mapplet name.
description	String	-	The mapplet description.
containerId	String	-	ID of the project or folder to contain the mapplet. If not included in request, the mapplet is created in the Default folder.

In addition to the POST attributes, pass the following information in the request body:

- Boundary value. Used to define different parts of the request body.

- File name. Name of the mapplet XML file.

- icSessionId. Informatica Intelligent Cloud Services session ID returned by the login resource. You can pass this information in the request body for clients that do not allow custom headers. If you can pass icSessionId as part of the request header, you can omit this information in the request body.

Use the following template for the customFunc POST request:

```
URL: <serverUrl>/api/v2/customFunc/
HTTP method: POST

Content-Type:multipart/form-data;boundary=<boundary value>
--<boundary value>
Content-Disposition:form-data; name="file";filename="<filename.XML>";Content-Type:text/
<xml|json>

<content of the mapplet XML file encoded as UTF-8>

--<boundary value>
Content-Disposition: form-data; name="name"

<mapplet name>
--<boundary value>
Content-Disposition: form-data; name="desc"

<description of the mapplet>
--<boundary value>
Content-Disposition: form-data; name="icSessionId"

<icSessionID returned from login resource>
--<boundary value>--
```

POST response

If successful, returns the customFunc response object for the PowerCenter mapplet that was created or updated.

Returns the error object if errors occur.

DELETE request

To delete a PowerCenter mapplet, use the mapplet ID in the following URI:

```
/api/v2/customFunc/<id>
```

DELETE response

Returns the 200 response code if the request is successful.

Returns the error object if errors occur.

POST example

To update a PowerCenter mapplet with an ID of 3 with an icSessionId of IV4wOrJmd6YUtmKa8t, you might use the following request. The updated mapplet is named Lookup Mapplet and uses the lookup_mapplet.xml file. XML data should be encoded in UTF-8.

```
URL: https://example.informatica.com/saas/api/v2/customFunc/3
HTTP method: POST

Content-Type:multipart/form-data;boundary=243553118520053
--243553118520053
Content-Disposition:form-data; name="file";filename="<lookup_mapplet.xml>";Content-
Type:text/xml

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE POWERMART SYSTEM "powrmart.dtd">
<POWERMART CREATION_DATE="05/14/2012 12:17:26" REPOSITORY_VERSION="181.90">
<REPOSITORY NAME="pc91hf1" VERSION="181" CODEPAGE="UTF-8" DATABASETYPE="Oracle">
<FOLDER NAME="test" GROUP="" OWNER="Administrator" SHARED="SHARED" DESCRIPTION=""
PERMISSIONS="rwx---r--" UUID="96f9d03b-c2c5-4034-8e3a-838026bbf6e8">
  <SOURCE BUSINESSNAME ="" DATABASETYPE ="Oracle" DBDNAME ="ddicst" DESCRIPTION ="" NAME
="CUSTOMERMASTER" OBJECTVERSION ="1" OWNERNAME ="C01" VERSIONNUMBER ="1">
    <SOURCEFIELD BUSINESSNAME ="" DATATYPE ="nvarchar2" DESCRIPTION ="" FIELDNUMBER ="1"
FIELDPROPERTY ="0" FIELDTYPE ="ELEMITEM" HIDDEN ="NO" KEYTYPE ="PRIMARY KEY" LENGTH ="0"
LEVEL ="0" NAME ="CUSTOMERID" NULLABLE ="NOTNULL" OCCURS ="0" OFFSET ="0" PHYSICALLENGTH
="30" PHYSICALOFFSET ="0" PICTURETEXT ="" PRECISION ="30" SCALE ="0" USAGE_FLAGS =""/>
.
.
.
 <ATTRIBUTE NAME ="Parameter Filename" VALUE =""/>
        <ATTRIBUTE NAME ="Write Backward Compatible Workflow Log File" VALUE ="NO"/>
        <ATTRIBUTE NAME ="Workflow Log File Name" VALUE ="wf_plugin_lookup.log"/>
        <ATTRIBUTE NAME ="Workflow Log File Directory" VALUE ="$PMWorkflowLogDir&#x5c;"/>
        <ATTRIBUTE NAME ="Save Workflow log by" VALUE ="By runs"/>
        <ATTRIBUTE NAME ="Save workflow log for these runs" VALUE ="0"/>
        <ATTRIBUTE NAME ="Service Name" VALUE =""/>
        <ATTRIBUTE NAME ="Service Timeout" VALUE ="0"/>
        <ATTRIBUTE NAME ="Is Service Visible" VALUE ="NO"/>
        <ATTRIBUTE NAME ="Is Service Protected" VALUE ="NO"/>
        <ATTRIBUTE NAME ="Fail task after wait time" VALUE ="0"/>
        <ATTRIBUTE NAME ="Enable HA recovery" VALUE ="NO"/>
        <ATTRIBUTE NAME ="Automatically recover terminated tasks" VALUE ="NO"/>
        <ATTRIBUTE NAME ="Service Level Name" VALUE ="Default"/>
        <ATTRIBUTE NAME ="Allow concurrent run with unique run instance name" VALUE
="NO"/>
        <ATTRIBUTE NAME ="Allow concurrent run with same run instance name" VALUE ="NO"/>
        <ATTRIBUTE NAME ="Maximum number of concurrent runs" VALUE ="0"/>
        <ATTRIBUTE NAME ="Assigned Web Services Hubs" VALUE =""/>
        <ATTRIBUTE NAME ="Maximum number of concurrent runs per Hub" VALUE ="1000"/>
        <ATTRIBUTE NAME ="Expected Service Time" VALUE ="1"/>
    </WORKFLOW>
```

```
        </FOLDER>
        </REPOSITORY>
        </POWERMART>

        --243553118520053
        Content-Disposition: form-data; name="name"

        Lookup Mapplet
        --243553118520053
        Content-Disposition: form-data; name="icSessionId"

        IV4wOrJmd6YUtmKa8t
        --243553118520053--
```

A successful request returns the customFunc response object for the mapplet that you updated.

dataPreview

Use this resource to preview data during mapping design. By default, the response returns up to ten rows of data for the specified object.

GET request

To request preview data, specify the connection ID or connection name and the object name in the URI. Optionally, you can include field format information in the request.

Use one of the following URIs:

- To request source data, use one of the following URIs:

      /api/v2/connection/source/<id>/datapreview/<objectName>

      /api/v2/connection/source/name/<name>/datapreview/<objectName>

- To request target data, use one of the following URIs:

      /api/v2/connection/target/<id>/datapreview/<objectName>

      /api/v2/connection/target/name/<name>/datapreview/<objectName>

You can receive field metadata in the response for flat file, Avro, Parquet, Orc, or JSON formats. To receive field metadata, include file format information in the request body.

For flat file format, you can include the following information in the flatFileAttrs object:

Field	Type	Required	Description
id	Long	Yes	Field ID.
delimiter	String	Yes	Character used to separate fields.
textQualifier	String	Yes	Quote character that defines the boundaries of text strings.
escapeChar	String	Yes	Character immediately preceding a field delimiter character embedded in an unquoted string, or immediately preceding the quote character in a quoted string.
headerLineNo	Int	Yes	Number of header lines.

Field	Type	Required	Description
firstDataRow	Int	Yes	The row number where the data begins in the file.
rowDelimiter	Int	--	Line break character. Enter the decimal code for an ASCII character between 1 and 32. Default is 10, which is the line feed character.
consecutiveDelimiter	Boolean	--	Indicates whether one or more consecutive delimiters are treated as one. Default is false.
multiDelimitersAsAnd	Boolean	--	If the delimiter you specify is more than one character, indicates whether the characters are treated as a single delimiter or multiple delimiters. Default is true.

For Avro, Parquet, Orc, or JSON formats, include the following information in the dataFormat object:

Field	Type	Required	Description
formatId	String	Yes	Format type, for example, Avro.
schema	String	--	Schema format.

By default, the dataPreview response returns 10 rows. For flat file connections, you can specify the number of rows using the numRows parameter as shown in the following example:

```
/api/v2/connection/source/<id>/datapreview/?objectName=<object name>&numRows=<number of
rows to view>
```

You can also specify the beginning row using the startRowNum parameter as shown in the following example:

```
/api/v2/connection/source/<id>/datapreview/?objectName=<object name>&startRowNum=<row
number of row to begin with>
```

Note: If you use the connection name in the URI and the connection name includes a space, replace the space with `%20`. For example:

```
/api/v2/connection/target/name/my%20connection/datapreview/SF_ACCOUNT.csv
```

GET response

Returns the dataPreview object for the requested connection ID or connection name and object name.

The dataPreview object includes the following attributes:

Field	Type	Description
connId	String	Connection ID.
objectName	String	Name of the source or target object.
header	String	Column headers.

Field	Type	Description
fieldName	String	Field name.
fieldBusinessName	String	Business field name.
data		Includes the following attribute in the dataPreviewEntry object.
values	String	Included in the dataPreviewEntry object. Field values from the source or target object.

GET request examples

The following example shows a request to preview data from the `SF_ACCOUNT.csv` object:

```
GET <serverUrl>/api/v2/connection/target/0000010B000000000003/datapreview/SF_ACCOUNT.csv
HTTP/1.0
Accept:application/json
icSessionId: <icSessionId>
```

The following example shows a request to preview data from the `customer.parquet` object.

```
POST <serverUrl>/api/v2/connection/source/0000010B000000000009/datapreview?
objectName=customer.parquet
1.0
Content-Type: application/xml
Accept: application/xml
icSessionId: <icSessionId>
{
    "@type": "dataFormat",
    "dformatId": "Parquet",
    "schema": "message AllData_root { optional int32 c_custkey; optional binary c_name
(UTF8); optional binary c_address (UTF8); optional int64 c_nationkey; optional binary
c_phone (UTF8); optional double c_acctbal; optional binary c_mktsegment (UTF8); required
binary c_comment (UTF8);}"
}
```

GET response example

You might receive a response similar to the following example:

```
{

    "@type": "dataPreview",
    "connId": "0000010B000000000003",
    "objectName": "SF_ACCOUNT.csv",
    "header":      [
      "ID",
      "ISDELETED",
      "MASTERRECORDID",
      "NAME",
      "TYPE",
      "PARENTID",
      "BILLINGSTREET",
      "BILLINGCITY",
      "BILLINGSTATE",
      "BILLINGPOSTALCODE",
      "BILLINGCOUNTRY",
      "BILLINGLATITUDE",
      "BILLINGLONGITUDE",
      "SHIPPINGSTREET",
      "SHIPPINGCITY",
      "SHIPPINGSTATE",
      "SHIPPINGPOSTALCODE",
      "SHIPPINGCOUNTRY",
      "SHIPPINGLATITUDE",
      "SHIPPINGLONGITUDE",
      "PHONE",
```

```
            "FAX",
            "ACCOUNTNUMBER",
            "WEBSITE"
        ],

    "fieldName":    [
            "ID",
            "ISDELETED",
            "MASTERRECORDID",
            "NAME",
            "TYPE",
            "PARENTID",
            "BILLINGSTREET",
            "BILLINGCITY",
            "BILLINGSTATE",
            "BILLINGPOSTALCODE",
            "BILLINGCOUNTRY",
            "BILLINGLATITUDE",
            "BILLINGLONGITUDE",
            "SHIPPINGSTREET",
            "SHIPPINGCITY",
            "SHIPPINGSTATE",
            "SHIPPINGPOSTALCODE",
            "SHIPPINGCOUNTRY",
            "SHIPPINGLATITUDE",
            "SHIPPINGLONGITUDE",
            "PHONE",
            "FAX",
            "ACCOUNTNUMBER",
            "WEBSITE"
        ],

    "fieldBusinessName":    [
            "ID",
            "ISDELETED",
            "MASTERRECORDID",
            "NAME",
            "TYPE",
            "PARENTID",
            "BILLINGSTREET",
            "BILLINGCITY",
            "BILLINGSTATE",
            "BILLINGPOSTALCODE",
            "BILLINGCOUNTRY",
            "BILLINGLATITUDE",
            "BILLINGLONGITUDE",
            "SHIPPINGSTREET",
            "SHIPPINGCITY",
            "SHIPPINGSTATE",
            "SHIPPINGPOSTALCODE",
            "SHIPPINGCOUNTRY",
            "SHIPPINGLATITUDE",
            "SHIPPINGLONGITUDE",
            "PHONE",
            "FAX",
            "ACCOUNTNUMBER",
            "WEBSITE"
        ],

    "rows":    [
            {
            "@type": "dataPreviewEntry",
            "values":            [
                "001i000000KIAQGAA5",
                "0",
                "",
                "ABCPoint",
                "Customer - Channel",
                "",
                "345 ABC Park",
                "Mountain View",
```

```
            "CA",
            "94063",
            "",
            "",
            "",
            "345 ABC Park",
            "Mountain View",
            "CA",
            "94063",
            "",
            "",
            "",
            "(650) 555-3450",
            "(650) 555-9895",
            "CC978213",
            "www.ABCpoint.com"
        ]
    },

        {
    "@type": "dataPreviewEntry",
    "values":          [
        "001i000000KIAQHAA5",
        "0",
        "",
        "123 United, UK",
        "Customer - Direct",
        "",
        "123 Estate,\nGateshead, Tyne and Wear NE26 3HS\nUnited Kingdom",
        "",
        "UK",
        "94063",
        "",
        "",
        "",
        "123 Estate,\nGateshead, Tyne and Wear NE26 3HS\nUnited Kingdom",
        "",
        "",
        "94063",
        "",
        "",
        "",
        "+44 123 4567899",
        "+44 123 4567899",
        "CD355119-A",
        "http://www.123United.com"
    ]

}
```

Dynamic mapping tasks

You can create a dynamic mapping task with the REST API to batch jobs together that are based on the same mapping. You can also run the task and get details about the job.

Use the following resources for dynamic mapping tasks:

- Login. Use to log in to Informatica Intelligent Cloud Services and get the session ID to use in dynamic task REST API calls.

- dynamictask. Use to create, view, update, or delete a dynamic mapping task.

- job. Use to start, stop, or get details about a dynamic mapping task run instance.

When you use these resources, note the following rules:

- Use JSON format.
- Use the following URL:

```
<serverUrl>/batch-mapping/api/v1/<API name>
```

- Use the following request header format:

```
<METHOD> <serverUrl>/<URI> HTTP/<HTTP version>
Content-Type: application/json
Accept: application/json
IDS-SESSION-ID: <SessionId>
```

The server URL includes the name and region of the POD that your organization uses and the Informatica Intelligent Cloud Services domain, informaticacloud.com. If you do not know the name and region of your organization's POD, you can find it by logging in to Informatica Intelligent Cloud Services through the user interface. The POD information is located in the browser's address bar.

In the following example, https://usw3.dm-us.informaticacloud.com is the server URL:

```
https://usw3.dm-us.informaticacloud.com/diUI/products/integrationDesign/main/home
```

Use the server URL as the base URL in the header of REST API calls.

Note: If you use a tool such as Postman that automatically includes the HTTP version, do not enter the HTTP version in the URL. If the HTTP version appears twice in the URL, the request fails.

Login

Use this resource to log into Informatica Intelligent Cloud Services when you use resources that require the IDS-SESSION-ID in the call header. The IDS-SESSION-ID is included in a successful login response.

POST request

Use the following URL:

```
<login URL>/identity-service/api/v1/Login
```

The login URL includes the region where your organization is located and the Informatica Intelligent Cloud Services domain, informaticacloud.com. You can find your organization's login region by opening the Informatica Intelligent Cloud Services log in page. The regional login URL is located in the browser's address bar before you log in to Informatica Intelligent Cloud Services.

In the following example, https://dm-us.informaticacloud.com, is the region URL:

```
https://dm-us.informaticacloud.com/identity-service/home
```

The following table describes the fields to include in the request:

Field	Type	Required	Description
username	String	Yes	Informatica Intelligent Cloud Services user name. Maximum length is 255 characters.
password	String	Yes	Informatica Intelligent Cloud Services password. Maximum length is 255 characters.

POST response

Returns the user object if the request is successful. Returns the error object if errors occur.

Use the session ID returned in the response for subsequent requests.

The user object includes the following attributes:

Field	Type	Description
sessionId	String	REST API session ID for the current session. Use in most REST API request headers.
sessionExpireTime	String	Time the session expires.
id	String	User ID.
name	String	Informatica Intelligent Cloud Services user name.
currentOrgId	String	Current organization ID.
currentOrgName	String	Name of the current organization.
parentOrgId	String	ID of the parent organization.
orgId	String	ID of the organization the user belongs to.
orgName	String	Name of the organization the user belongs to.
groups	String	User group.
effectiveRoles	String	Roles assigned to the user.
effectivePrivileges	String	Privileges assigned to the user.
status	String	Status of the user.
timeZoneId	String	Time zone of the user. Time zone honors Daylight Saving Time. For more information, see "Time zone codes" on page 543.
authenticator	String	User authentication method.

dynamictask

Use this resource to request the details of a dynamic mapping task. You can also create, update, or delete a dynamic mapping task.

To request details or to update a dynamic task that already exists, you need the task ID. You can get the task ID using the V3 lookup resource. To lookup object details with the V3 lookup resource, use the following URI:

```
/saas/public/core/v3/lookup
```

Include BATCH_MAPPING as the object type as shown in the following example:

```
{
  "objects": [
    {
      "path": "Default/DMT_API",
      "type": "BATCH_MAPPING"
    }
  ]
}
```

The response returns details about the objects in the path as shown in the following example:

```
{
  "objects": [
```

```
        {
          "id": "7H67JPHH9Y4g7Hm7JyL5K2",
          "path": "Default/DMT_API",
          "type": "BATCH_MAPPING",
          "description": "",
          "updatedBy": "rl.ma",
          "updateTime": "2021-08-27T23:45:14Z"
        }
      ]
    }
```

For more information about using the V3 lookup resource, see "Lookup" on page 139.

You can also find the task ID by opening the task in the Data Integration user interface. In the URL, the last string of characters in the task ID.

For example, in the following URL, the task ID is 771b8ZpTcfreXm8n5RZUQ5:

```
https://na1.dm-us.informaticacloud.com/diUI/products/integrationDesign/main/
dynamicmapping/771b8ZpTcfreXm8n5RZUQ5
```

GET request

To request the details of a dynamic mapping task, use the task ID.

Use the following URI to request the details of a dynamic mapping task:

```
/batch-mapping/api/v1/dynamictask/<id>
```

GET response

Returns the dynamictask object for the requested task ID.

Returns the error object if errors occurred.

The following table describes the attributes in a dynamictask object:

Field	Type	Description
name	String	Name of the dynamic mapping task.
mappingId	String	ID of the mapping used in the task.
mappingDocType	String	Type of mapping used in the task.
runtimeEnvironmentId	String	Runtime environment used for the task.
groups		Groups in the dynamic mapping task.
groupName	String	Included in the group object. Name of the group.
enabled	Boolean	Included in the group object. Whether the group is enabled or not. Returns true when the group is enabled.
parameters		Parameters in the dynamic mapping task.
name	String	Included in the parameter object. Name of the parameter.

Field	Type	Description
type	String	Included in the parameter object. Type of parameter.
txName	String	Included in the parameter object. Name of the transformation that uses the parameter.
uniqueName	String	Included in the parameter object. Transformation name and parameter name. If no transformation name is present, then parameter name.
scope	String	Included in the parameter object. Scope of the parameter, either DEFAULT or LOCAL.
label	String	Included in the parameter object. Parameter label.
description	String	Included in the parameter object. Parameter description.
retentionPolicy	String	Included in the parameter object. Applicable to in-out parameters. Determines when the task retains the current parameter value.
aggregationType	String	Included in the parameter object. Applicable to in-out parameters. Type of calculation the parameter performs.
job		Jobs in the dynamic mapping task.
jobName	String	Included in the job object. Name of the job.
jobType	String	Included in the job object. Type of job, either USER or DEFAULT.
enabled	Boolean	Included in the job object. Determines if the job is enabled. Default is false.
stopOnError	Boolean	Included in the job object. Stops the job when it encounters an error. Default is false.
stopOnWarning	Boolean	Included in the job object. Stops the job when it encounters a warning. Default is false.
preProcessingCmds	Sting	Included in the job object. List of commands to run before the task.

Field	Type	Description
postProcessingCmds	String	Included in the job object. List of commands to run after the task.
advSessionProperties	Map	Included in the job object. Map of the advanced session properties that are set for the job.
group	String	Included in the job object. Name of the group that the job belongs to.
paramValueBindings		Included in the job object. Parameter attributes in the job.
paramDefnRef	Sting	Included in the paramValueBindings object. Unique name of the parameter.
type	String	Included in the paramValueBindings object. Type of parameter. Value can be one of the following types: - Connection - String - Source - Target - INOUT - Sequence
connection		Attributes for connection type parameters.
connectionId	String	Included in the paramValueBindings object. Connection ID.
connectionType	String	Included in the paramValueBindings object. Type of connection.
runtimeAttrs	Map	Included in the paramValueBindings object. Runtime attributes for the connection.
oprRuntimeAttrs	Map	Included in the paramValueBindings object. Read/write runtime attributes.
source		Attributes for source type parameters.
sourceObject	Object	Included in the paramValueBindings object. Source object.
advancedFilterExpression	String	Included in the paramValueBindings object. Advanced filter condition.
filterFields	List of objects	Included in the paramValueBindings object. List of filter fields.
sortFields	List of objects	Included in the paramValueBindings object. List of sort fields.

Field	Type	Description
srcFFAttrs	Object	Included in the paramValueBindings object. Flat file attributes.
ccmDataFormat	Object	Included in the paramValueBindings object. Data format
customQuery	String	Included in the paramValueBindings object. Custom query
handleSpecialChars	Boolean	Included in the paramValueBindings object. Determines if the task can use special characters.
runtimeAttrs	Map	Included in the paramValueBindings object. Runtime Attributes
oprRuntimeAttrs	Map	Included in the paramValueBindings object. Read/write runtime attributes.
string		Attributes for string type parameters.
text	String	Included in the paramValueBindings object. Text value for the parameter.
target		Attributes for target type parameters.
objectName	String	Included in the paramValueBindings object. Name of the existing target object.
objectLabel	String	Included in the paramValueBindings object. Label of the existing target object.
newObjectName	String	Included in the paramValueBindings object. Name of the new target file.
truncateTarget	Boolean	Included in the paramValueBindings object. Determines if Data Integration truncates the target.
bulkApiDBTarget	Boolean	Included in the paramValueBindings object. Determines if bulk API is used.
operationType	String	Included in the paramValueBindings object. Task operation.
tgtFFAttrs	String	Included in the paramValueBindings object. Flat file attributes.
tgtObjectAttributes	Map	Included in the paramValueBindings object. Target object attributes.

Field	Type	Description
runtimeAttrs	Map	Included in the paramValueBindings object. Run time attributes
oprRuntimeAttrs	Map	Included in the paramValueBindings object. Read/write runtime attributes.
ccmDataFormat	Object	Included in the paramValueBindings object. Data format for CCI targets.
dynamicFileName	Boolean	Included in the paramValueBindings object. Determines if the target file name is dynamic.
handleSpecialChars	Boolean	Included in the paramValueBindings object. Determines if the target object handles special characters.
INOUT		Attributes for INOUT type parameters.
initialValue	String	Included in the paramValueBindings object. Initial value of the in-out parameter.
datatype	String	Included in the paramValueBindings object. Data type of the parameter.
precision	String	Included in the paramValueBindings object. Precision of the parameter.
scale	String	Included in the paramValueBindings object. Scale of the parameter.
retentionPolicy	String	Included in the paramValueBindings object. Determines when the task retains the current parameter value.
aggregationType	String	Included in the paramValueBindings object. Type of calculation the parameter performs.
currentValue	String	Included in the paramValueBindings object. Current value of the in-out parameter.
sequence		Attributes for sequence type parameters.
txName	String	Included in the paramValueBindings object. Name of the Sequence Generator transformation.
initialValue	String	Included in the paramValueBindings object. Initial value of the Sequence Generator transformation.
value	String	Included in the paramValueBindings object. Current value of the Sequence Generator transformation.

POST request

To create a dynamic mapping task, use the following URI:

```
/batch-mapping/api/v1/dynamictask
```

If you want to specify a location for the task, include the container ID as a query parameter in the request. If the container ID isn't included in the request, the task is created in the Default folder. You can find the container ID for a project or folder in the Data Integration user interface. On the Explore page, select the folder. In the URL, the last string of characters in the container ID.

For example, in the following URL, the container ID is dH2DuGJYda7ijgW4Sm32sR:

```
https://na1.dm-us.informaticacloud.com/diUI/products/integrationDesign/main/Explore/
dH2DuGJYda7ijgW4Sm32sR
```

The following table describes the attributes you can include in an dynamictask object:

Field	Type	Required	Description
name	String	Yes	Name of the dynamic mapping task.
mappingId	String	Yes	ID of the mapping used in the task.
mappingDocType	String	Yes	Type of mapping used in the task. Include one of the following types: - MAPPING. Use for mappings outside of advanced mode. - AT_SCALE_MAPPING. Use for mappings in advanced mode.
runtimeEnvironmentId	String	Yes	Runtime environment used for the task.
groups			Groups in the dynamic mapping task.
groupName	String	Yes	Include in the group object. Name of the group.
enabled	Boolean		Include in the group object. Determines if the group is enabled or not.
parameters			Parameters in the dynamic mapping task.
name	String	Yes	Include in the parameters object. Name of the parameter in the mapping.

Field	Type	Required	Description
type	String	Yes	Include in the parameters object. Type of parameter. Include one of the following types: - SOURCE_CONNECTION - SOURCE_OBJECT - TARGET_CONNECTION - TARGET_OBJECT - LOOKUP_CONNECTION - LOOKUP_OBJECT - TRANSFORM_CONNECTION - STRING - GROUPBY - FIELD - EXPRESSION - EXPRESSION_FIELDS - CONDITION - FIELD_MAPPING - SORTLIST - INOUT_INT - INOUT_BIGINT - INOUT_STRING - INOUT_TEXT - INOUT_DECIMAL - INOUT_DOUBLE - INOUT_DATETIME - SEQUENCE
txName	String		Include in the parameters object. Name of the transformation that uses the parameter.
uniqueName	String	Yes	Include in the parameters object. Transformation name and parameter name. If no transformation name is present, then parameter name.
Scope	String	Yes	Include in the parameters object. Scope of the parameter, either default or local.
label	String		Include in the parameters object. Parameter label.
description	String		Include in the parameters object. Parameter description.
retentionPolicy	String		Include in the parameters object. Applicable to in-out parameters. Determines when the task retains the current parameter value. Use one of the following values: - ON_SUCCESS_OR_WARNING - ON_SUCCESS - ON_WARNING - NEVER

Field	Type	Required	Description
aggregationType	String		Include in the parameters object. Applicable to in-out parameters. Type of calculation the parameter performs. Use one of the following values: - MAX - MIN - COUNT
job			Jobs in the dynamic mapping task.
jobName	String	Yes	Include in the job object. Name of the job.
jobType	String	Yes	Include in the job object. Type of job. Use either USER or DEFAULT.
enabled	Boolean		Include in the job object. Determines if the job is enabled. Default is false.
stopOnError	Boolean		Include in the job object. Stops the job when it encounters an error. Default is false.
stopOnWarning	Boolean		Include in the job object. Stops the job when it encounters a warning. Default is false.
preProcessingCmds	String		Include in the job object. List of commands to run before the task.
postProcessingCmds	String		Include in the job object. List of commands to run after the task.
advSessionProperties	Map		Include in the job object. Map of the advanced session properties that are set for the job.
group	String	Yes	Include in the job object. Name of the group that the job belongs to.
paramValueBindings			Parameter attributes in each job.
paramDefnRef	Sting	Yes	Include in the paramValueBindings object. Unique name of the parameter.

Field	Type	Required	Description
type	String	Yes	Include in the paramValueBindings object. Type of parameter. Use one of the following types: - Connection - String - Source - Target - INOUT - Sequence
connection			Attributes for connection type parameters.
connectionId	String		Include in the paramValueBindings object. Connection ID.
connectionType	String		Include in the paramValueBindings object. Type of connection. Use one of the following types: - SOURCE - TARGET - LOOKUP - TRANSFORM
runtimeAttrs	Map		Include in the paramValueBindings object. Runtime attributes for the connection.
oprRuntimeAttrs	Map		Include in the paramValueBindings object. Read/write runtime attributes.
source			Attributes for source type parameters.
sourceObject	Object		Include in the paramValueBindings object. Source object.
advancedFilterExpression	String		Include in the paramValueBindings object. Advanced filter condition.
filterFields	List of objects		Include in the paramValueBindings object. List of filter fields.
sortFields	List of objects		Include in the paramValueBindings object. List of sort fields.
srcFFAttrs	Object		Include in the paramValueBindings object. Flat file attributes.
ccmDataFormat	Object		Include in the paramValueBindings object. Data format for CCI sources.
customQuery	String		Include in the paramValueBindings object. Custom query.
handleSpecialChars	Boolean		Include in the paramValueBindings object. Determines if the task can use special characters.

Field	Type	Required	Description
runtimeAttrs	Map		Include in the paramValueBindings object. Runtime attributes for the source.
oprRuntimeAttrs	Map		Include in the paramValueBindings object. Read/write runtime attributes.
string			Attributes for string type parameters.
text	String		Include in the paramValueBindings object. Text value for parameter.
target			Attributes for target type parameters.
objectName	String		Include in the paramValueBindings object. Name of the existing target object.
objectLabel	String		Include in the paramValueBindings object. Label of the existing target object.
newObjectName	String		Include in the paramValueBindings object. Name of the new target file.
truncateTarget	Boolean		Include in the paramValueBindings object. Determines if Data Integration truncates the target.
bulkApiDBTarget	Boolean		Include in the paramValueBindings object. Determines if bulk API is used.
operationType	String		Include in the paramValueBindings object. Task operation.
tgtFFAttrs	String		Include in the paramValueBindings object. Flat file attributes.
tgtObjectAttributes	Map		Include in the paramValueBindings object. Target object attributes.
runtimeAttrs	Map		Include in the paramValueBindings object. Run time attributes.
oprRuntimeAttrs	Map		Include in the paramValueBindings object. Read/write runtime attributes.
ccmDataFormat	Object		Include in the paramValueBindings object. Data format for CCI targets.
dynamicFileName	Boolean		Include in the paramValueBindings object. Determines if the target file name is dynamic.

Field	Type	Required	Description
handleSpecialChars	Boolean		Include in the paramValueBindings object. Determines if the target object handles special characters.
INOUT			Attributes for INOUT type parameters.
initialValue	String		Include in the paramValueBindings object. Initial value of the in-out parameter.
datatype	String		Include in the paramValueBindings object. Data type of the parameter.
precision	String		Include in the paramValueBindings object. Precision of the parameter.
scale	String		Include in the paramValueBindings object. Scale of the parameter.
retentionPolicy	String		Include in the paramValueBindings object. Determines when the task retains the current parameter value. Use one of the following values: - ON_SUCCESS_OR_WARNING - ON_SUCCESS - ON_WARNING - NEVER
aggregationType	String		Include in the paramValueBindings object. Type of calculation the parameter performs. Use one of the following values: - MAX - MIN - COUNT
currentValue	String		Include in the paramValueBindings object. Current value of the in-out parameter.
sequence			Attributes for sequence type parameters.
txName	String		Include in the paramValueBindings object. Name of the Sequence Generator transformation.
initialValue	String		Include in the paramValueBindings object. Initial value of the Sequence Generator transformation.
value	String		Include in the paramValueBindings object. Current value of the Sequence Generator transformation.

POST response

If successful, returns the dynamictask object that you created or updated. Returns the error object if errors occur.

PUT request

To update a dynamic mapping task, include the task ID as shown in the following example:

```
/batch-mapping/api/v1/dynamictask/<Id>
```

When you update a dynamic mapping task, include the same attributes as a POST request.

PUT response

Returns the task ID, state, and any validation errors as shown in the following example:

```
{
  "frsId": "1JVMWZjVPMhKY4SdxcGd60",
  "state": "VALID",
  "validationErrors": []
}
```

DELETE request

To delete a dynamic mapping task, use the task ID in the following URI:

```
/batch-mapping/api/v1/dynamictask/<id>
```

DELETE response

Returns the 200 response code if the request is successful.

Returns the error object if errors occur.

POST example

To create a new dynamic mapping task with the REST API, you might use the following request:

```
POST https://na1.dm-us.informaticacloud.com/batch-mapping/api/v1/dynamictask
Content-Type: application/json
Accept: application/json
IDS-SESSION-ID: jpaybAKQMsmdt7vLJ02z0s
{
    "orgId": "2ij4X7Pd63ibnquEQyy9wA",
    "name": "DMT_API",
    "description": "",
    "mappingId": "01003Y1700000000005X",
    "mappingDocType": "MAPPING",
    "runtimeEnvironmentId": "01003Y25000000000004",
    "scheduleId": null,
    "state": "VALID",
    "groups": [
        {
      "groupName": "Group_1",
      "enabled": true
    },
    {
      "groupName": "Group_2",
      "enabled": false
    }
  ],
   "parameters": [
      {
        "uniqueName": "Source:SrcCon",
        "name": "SrcCon",
        "txName": "Source",
        "type": "SOURCE_CONNECTION",
        "scope": "DEFAULT",
        "label": null,
        "description": "",
        "retentionPolicy": null,
        "aggregationType": null
      },
        {
        "uniqueName": "Source:SrcObj",
        "name": "SrcObj",
```

```
      "txName": "Source",
      "type": "SOURCE_OBJECT",
      "scope": "LOCAL",
      "label": null,
      "description": "",
      "retentionPolicy": null,
      "aggregationType": null
    },
    {
      "uniqueName": "Target:TrgCon",
      "name": "TrgCon",
      "txName": "Target",
      "type": "TARGET_CONNECTION",
      "scope": "DEFAULT",
      "label": null,
      "description": "",
      "retentionPolicy": null,
      "aggregationType": null
    },
    {
      "uniqueName": "Target:TrgObj",
      "name": "TrgObj",
      "txName": "Target",
      "type": "TARGET_OBJECT",
      "scope": "LOCAL",
      "label": null,
      "description": "",
      "retentionPolicy": null,
      "aggregationType": null
    },
    {
      "uniqueName": "Lookup:Lkcon",
      "name": "Lkcon",
      "txName": "Lookup",
      "type": "LOOKUP_CONNECTION",
      "scope": "DEFAULT",
      "label": null,
      "description": "",
      "retentionPolicy": null,
      "aggregationType": null
    },
    {
      "uniqueName": "Lookup:lkObj",
      "name": "lkObj",
      "txName": "Lookup",
      "type": "LOOKUP_OBJECT",
      "scope": "DEFAULT",
      "label": null,
      "description": "",
      "retentionPolicy": null,
      "aggregationType": null
    },
    {
      "uniqueName": "Lkp",
      "name": "Lkp",
      "txName": null,
      "type": "EXPRESSION",
      "scope": "LOCAL",
      "label": null,
      "description": "",
      "retentionPolicy": null,
      "aggregationType": null
    },
    {
      "uniqueName": "ExParam",
      "name": "ExParam",
      "txName": null,
      "type": "EXPRESSION",
      "scope": "LOCAL",
      "label": null,
      "description": "",
```

```
                    "retentionPolicy": null,
                    "aggregationType": null
                }
            ],
            "jobs": [
                {
                    "jobUUID": "ae1lR3k2ccYgXNeFQe4DIT",
                    "jobName": "<Default>",
                    "jobType": "DEFAULT",
                    "enabled": false,
                    "stopOnError": false,
                    "stopOnWarning": false,
                    "preProcessingCmds": [],
                    "postProcessingCmds": [],
                    "advSessionProperties": {},
                    "paramValueBindings": [
                        {
                            "type": "Connection",
                            "paramDefnRef": "Source:SrcCon",
                            "connectionId": "01003Y0B000000000006",
                            "connectionType": null,
                            "runtimeAttrs": {},
                            "oprRuntimeAttrs": {}
                        },
                        {
                            "type": "Connection",
                            "paramDefnRef": "Target:TrgCon",
                            "connectionId": "01003Y0B000000000001Q",
                            "connectionType": null,
                            "runtimeAttrs": {},
                            "oprRuntimeAttrs": {}
                        },
                        {
                            "type": "Connection",
                            "paramDefnRef": "Lookup:Lkcon",
                            "connectionId": "01003Y0B000000000001Q",
                            "connectionType": null,
                            "runtimeAttrs": {},
                            "oprRuntimeAttrs": {}
                        },
                        {
                            "type": "Source",
                            "paramDefnRef": "Lookup:lkObj",
                            "sourceObject": {
                                "name": "EMP",
                                "label": "EMP",
                                "metadataUpdated": false
                            },
                            "advancedFilterExpression": null,
                            "userDefinedJoin": null,
                            "filterFields": [],
                            "sortFields": [],
                            "srcFFAttrs": null,
                            "overriddenFields": [],
                            "ccmDataFormat": null,
                            "customQuery": null,
                            "handleSpecialChars": false,
                            "runtimeAttrs": {},
                            "oprRuntimeAttrs": {}
                        }
                    ],
                    "group": null
                },
                {
                    "jobUUID": "21rswJo8MnOgUTtfCq96AR",
                    "jobName": "Job_1",
                    "jobType": "USER",
                    "enabled": true,
                    "stopOnError": false,
                    "stopOnWarning": false,
                    "preProcessingCmds": [],
```

```
                "postProcessingCmds": [],
                "advSessionProperties": {},
                "paramValueBindings": [
                    {
                        "type": "Source",
                        "paramDefnRef": "Source:SrcObj",
                        "sourceObject": {
                            "name": "employee.csv",
                            "label": "employee.csv",
                            "metadataUpdated": false
                        },
                        "advancedFilterExpression": null,
                        "userDefinedJoin": null,
                        "filterFields": [],
                        "sortFields": [],
                        "srcFFAttrs": null,
                        "overriddenFields": [],
                        "ccmDataFormat": null,
                        "customQuery": null,
                        "handleSpecialChars": false,
                        "runtimeAttrs": {},
                        "oprRuntimeAttrs": {}
                    },
                    {
                        "type": "Target",
                        "paramDefnRef": "Target:TrgObj",
                        "objectName": "CONTACT",
                        "objectLabel": "CONTACT",
                        "newObjectName": null,
                        "truncateTarget": false,
                        "bulkApiDBTarget": false,
                        "operationType": null,
                        "tgtFieldRefs": {},
                        "targetUpdateColumns": [],
                        "tgtFFAttrs": null,
                        "tgtObjectAttributes": {},
                        "runtimeAttrs": {},
                        "oprRuntimeAttrs": {},
                        "handleSpecialChars": false,
                        "ccmDataFormat": null,
                        "dynamicFileName": false
                    },
                    {
                        "type": "String",
                        "paramDefnRef": "Lkp",
                        "text": " EMP_ID||EMP_NAME"
                    },
                    {
                        "type": "String",
                        "paramDefnRef": "ExParam",
                        "text": " IsNull(EMP_ID)"
                    }
                ],
                "group": "Group_1"
            },
            {
                "jobUUID": "6pavcOH4kwZewe1XL1khoF",
                "jobName": "Job_2",
                "jobType": "USER",
                "enabled": true,
                "stopOnError": false,
                "stopOnWarning": false,
                "preProcessingCmds": [],
                "postProcessingCmds": [],
                "advSessionProperties": {},
                "paramValueBindings": [
                    {
                        "type": "Source",
                        "paramDefnRef": "Source:SrcObj",
                        "sourceObject": {
                            "name": "Boston_Customers.csv",
```

```
            "label": "Boston_Customers.csv",
            "metadataUpdated": false
          },
          "advancedFilterExpression": null,
          "userDefinedJoin": null,
          "filterFields": [],
          "sortFields": [],
          "srcFFAttrs": null,
          "overriddenFields": [],
          "ccmDataFormat": null,
          "customQuery": null,
          "handleSpecialChars": false,
          "runtimeAttrs": {},
          "oprRuntimeAttrs": {}
        },
        {
          "type": "Target",
          "paramDefnRef": "Target:TrgObj",
          "objectName": "CUSTINFO_TYPE",
          "objectLabel": "CUSTINFO_TYPE",
          "newObjectName": null,
          "truncateTarget": false,
          "bulkApiDBTarget": false,
          "operationType": null,
          "tgtFieldRefs": {},
          "targetUpdateColumns": [],
          "tgtFFAttrs": null,
          "tgtObjectAttributes": {},
          "runtimeAttrs": {},
          "oprRuntimeAttrs": {},
          "handleSpecialChars": false,
          "ccmDataFormat": null,
          "dynamicFileName": false
        },
        {
          "type": "String",
          "paramDefnRef": "Lkp",
          "text": "NAME=Firstname"
        },
        {
          "type": "String",
          "paramDefnRef": "ExParam",
          "text": "Firstname||Lastname"
        }
      ],
      "group": "Group_2"
    }
  ]
}
```

job

When you use the REST API to run a dynamic mapping task, use the REST API version 1 job resource to start or stop the job. You can also get details about the job.

Do not use the platform REST API version 2 job resource to get the status of a dynamic mapping task.

If your organization uses projects and folders, use the REST API version 3 lookup resource to retrieve the task ID. This ID is the federated task ID, which you must include in the POST request.

GET request

To get details of a dynamic mapping task run, use the following URI:

```
/batch-mapping/api/v1/Job/monitor/task/<Id>/run/<runId>
```

GET response

If successful, returns the job status.

If unsuccessful, the response includes a reason for the failure.

For example, you might get the following response when you request details of a completed dynamic mapping task:

```
{
    "taskId": "jUJNIX39Z6ZbR8KZCm2ieS",
    "taskFrsId": "k2AE77O06oYg6NvrOtKt6t",
    "taskName": "Dynamic Mapping Task2",
    "instanceId": 1,
    "startedBy": "user@informatica.com",
    "startTime": "2021-08-26T16:28:11.000Z",
    "updateTime": "2021-08-26T16:28:35.000Z",
    "endTime": "2021-08-26T16:28:35.000Z",
    "runtimeEnvironment": "test1",
    "runtimeEnvironmentId": "01000025000000000002",
    "status": "COMPLETED",
    "successRows": 3,
    "errorRows": 0,
    "saasMappingId": "01000017000000000007",
    "mappingName": "dsst__copy_data_new_tgt_With_SortList",
    "mappingFrsId": "5A90bRPboO0dpMQ8F2nkgy",
    "mappingDocType": "MAPPING",
    "runContext": "API",
    "scheduleName": null,
    "jobs": [
            {
                    "jobName": "Job_1",
                    "jobUUID": "78OZ7JlUNSCd09kwQWXbUf",
                    "groupName": "Group_1",
                    "saasJobRunId": 52,
                    "saasLogId": "010000C100000000040H",
                    "startTime": "2021-08-26T16:28:18.000Z",
                    "updateTime": "2021-08-26T16:28:33.000Z",
                    "endTime": "2021-08-26T16:28:33.000Z",
                    "errorMessage": null,
                    "status": "COMPLETED",
                    "failedSourceRows": 0,
                    "successSourceRows": 3,
                    "failedTargetRows": 0,
                    "successTargetRows": 3,
                    "enabled": true,
                    "sessionLogUrl": null
            }
        ]
    }
```

Start POST request

To run a dynamic mapping task, use the following URI:

```
/batch-mapping/api/v1/Job
```

Include the federated task ID in the request as in the following example:

```
{
        "taskFrsId": "k2AE77O06oYg6NvrOtKt6t"
}
```

Start POST response

Returns the run ID and the federated task ID.

For example, if you run a dynamic mapping task for the second time, you get the following response:

```
{
        "runId": 2,
        "taskFrsId": "k2AE77O06oYg6NvrOtKt6t"
}
```

Stop POST request

To stop a dynamic mapping task run, use the following URI

```
/batch-mapping/api/v1/Job/stop
```

Include the task ID and the job run ID attributes in the job object as shown in the following example:

```
{
"taskFrsId": "gScmpuSzjSdcbNPFNYbbcg",
"runId": 10
}
```

Stop POST response

Returns the 200 success object if the request is successful. Returns the error object if errors occur.

expressionValidation

Use this resource to validate expressions.

POST Request

To validate an expression, use the following URI:

```
/saas/api/v2/expression/validate
```

Use the following attributes in the request body:

Field	Type	Required	Description
expr	String	Yes	The expression to validate.
connectionId	String	Yes	Connection ID.
objectName	String	Yes	Name of the source or target object.
isSourceType	Boolean	Yes	Whether the expression is for a source object. Values are True or False.

If the expression is valid, the response returns a message that says the expression is valid. If the expression is not valid, the response returns an error.

POST Example

To validate an expression, you might use the following request:

```
POST <serverURL>/api/v2/expression/validate
Content-Type: application/json
Accept: application/json
    {
    "@type":"expressionValidation",
    "expr":"REPVERSION",
    "connectionId":"0000010B000000000004",
    "objectName":"OPB_REPOSIT",
    "isSourceType":true
    }
```

field

A field is a subset of a data structure that represents a single data item. For example, a database table column is a field. Use this resource to request field details for a source or target object and to update the flat file attributes for a source or target object.

GET request

Request field details of a source object or a target object.

- To request the field details of a source object, use the source connection ID or source connection name and the source object name. Use one of the following URIs:

    ```
    /api/v2/connection/source/<id>/field/<object name>
    /api/v2/connection/source/name/<name>/field/<object name>
    ```

- To request the field details of a target object, use the target connection ID or target connection name and the target object name. Use one of the following URIs:

    ```
    /api/v2/connection/target/<id>/field/<object name>
    /api/v2/connection/target/name/<name>/field/<object name>
    ```

If you use the connection name in the URI and the connection name includes a space, replace the space with `%20`. For example:

```
/api/v2/connection/source/name/my%20connection/field/customer
```

You can also use the following URI, which accommodates searching for an object that includes a forward slash (/):

```
/api/v2/connection/<source or target>/<id>/fields?objectName=<objectName>
```

Note: The object name is case-sensitive.

GET response

Returns the field object for each field in the requested object.

Returns the error object if errors occur.

The field object includes different information based on the connection type. The following attributes might be returned in a field object:

Field	Type	Description
id	Long	Field ID.
name	String	Field name.
type	String	Field type.
label	String	Field label.
parentObject	String	Parent object, if applicable.
precision	Int	Length of the field in bytes.
pcType	String	PowerCenter data type.
scale	Int	Number of digits after the decimal point for numeric values.

Field	Type	Description
columnIndex	Int	Column index.
isKey	Boolean	Whether the field is a used as a key.
isExternalId	Boolean	Whether the field is used as an external ID.
isSfIdLookup	Boolean	Whether the field is used as a Salesforce ID lookup field.
isNullible	Boolean	Whether the field can contain null values.
isUnique	Boolean	Whether the field requires unique values.
isCreateable	Boolean	Whether the field accepts new values.
isCalculated	Boolean	Whether the field is calculated.
isUpdateable	Boolean	Whether the field allows updates.
isFilterable	Boolean	Whether the field can be filtered.
linkedFields	String	For a masking task, the source field mapped to the input field of the mapplet.
relatedInfos		Information about related fields included in a fieldRelatedInfo object for each related field.
fieldId	Long	Included in the fieldRelatedInfo object. Field ID.
referenceObject	String	Included in the fieldRelatedInfo object. Object that includes the field.
relationshipName	String	Included in the fieldRelatedInfo object. Relationship to object.
references		Reference information included in a fieldRelatedInfo object for each related field.
fieldId	Long	Included in the fieldRelatedInfo object. Field ID.
referenceObject	String	Included in the fieldRelatedInfo object. Object that includes the field.
relationshipName	String	Included in the fieldRelatedInfo object. Relationship to object.
javaType	String	Java data type.
showLabel	Boolean	Whether to show the field label.
naturalOrder	Int	Position number of the field in the source.
customProperties		Custom properties for the field.

GET example

To get the field details for the Customer object available through the source connection (ID: 0002D420000000J), you might use the following request:

```
GET <serverUrl>/api/v2/connection/source/0002D420000000J/field/Customer
Content-Type: application/xml
Accept: application/xml
icSessionId: <icSessionId>
```

A successful request returns the fields object for each field in the Customer source object.

POST request for flat file attributes

You can use a POST request to update flat file attributes for a flat file connection.

Use one of the following URI:

```
/api/v2/connection/source/<id>/field/<objectName>
/api/v2/connection/target/<id>/field/<objectName>
```

When you send a request to change flat file attributes, the flat file attributes provided in the request override the default attributes specified in the connection object.

To change flat file attributes, include the following information in the flatFileAttrs object:

Field	Type	Required	Description
id	Long	Yes	Field ID.
delimiter	String	Yes	Character used to separate fields.
textQualifier	String	Yes	Quote character that defines the boundaries of text strings.
escapeChar	String	Yes	Character immediately preceding a field delimiter character embedded in an unquoted string, or immediately preceding the quote character in a quoted string.
headerLineNo	Int	Yes	Number of header lines.
firstDataRow	Int	Yes	The row number where the data begins in the file.
rowDelimiter	Int	--	Line break character. Enter the decimal code for an ASCII character between 1 and 32. Default is 10, which is the line feed character.
consecutiveDelimiter	Boolean	--	Indicates whether one or more consecutive delimiters are treated as one. Default is false.
multiDelimitersAsAnd	Boolean	--	If the delimiter you specify is more than one character, indicates whether the characters are treated as a single delimiter or multiple delimiters. Default is true.

POST request example for flat file attributes

To send a request for field information for a flat file connection, you might use the following request:

```
POST <serverUrl>/api/v2/connection/source/0000010B000000000021/field/test_precision.csv
1.0
Content-Type: application/xml
```

```
Accept: application/xml
icSessionId
{
    "@type": "flatFileAttrs",
    "delimiter": ",",
    "textQualifier": "'",
    "escapeChar": "\\"
}
```

POST request for non-flat file formats

For Avro, Parquet, Orc, and JSON formats, you can receive field information by including the format type and optionally, the schema. If you don't include the schema, the schema format is inferred from the data file.

Use one of the following URIs:

```
/api/v2/connection/source/<id>/fields?objectName=<object name>
/api/v2/connection/target/<id>/fields?objectName=<object name>
```

You can use the following attributes in the dataFormat object:

Field	Type	Required	Description
formatId	String	Yes	Format type, for example, Avro.
schema	String	--	Schema format.

POST request example for a non-flat file formats

To send a request for field information for a Parquet connection, you might use a request that is similar to the following example:

```
POST <serverUrl>/api/v2/connection/source/0100000B00000000000F/fields?
objectName=infa.qa.bucket%2Fcustomer.parquet
1.0
Content-Type: application/xml
Accept: application/xml
icSessionId: <icSessionId>
{
    "@type": "dataFormat",
    "dformatId": "Parquet",
    "schema": "message AllData_root { optional int32 c_custkey; optional binary c_name
(UTF8); optional binary c_address (UTF8); optional int64 c_nationkey; optional binary
c_phone (UTF8); optional double c_acctbal; optional binary c_mktsegment (UTF8); required
binary c_comment (UTF8);}"
}
```

A successful response might look like the following example:

```
[
    {
        "@type": "field",
        "id": -1,
        "name": "c_custkey",
        "type": "parquet_int32",
        "uniqueName": "c_custkey",
        "label": "c_custkey",
        "parentObject": "customer_tgt.parquet",
        "pcType": "INTEGER",
        "precision": 10,
        "scale": 0,
        "columnIndex": -1,
        "isKey": false,
        "isExternalId": false,
        "isSfIdLookup": false,
        "isNullable": true,
        "isUnique": false,
        "isCreateable": false,
```

```
            "isUpdateable": true,
            "isFilterable": true,
            "isCalculated": false,
            "javaType": "java.lang.Integer",
            "showLabel": true,
            "naturalOrder": 0,
            "linkedFields": [],
            "relatedInfos": [],
            "references": []
        },
        {
            "@type": "field",
            "id": -1,
            "name": "c_address",
            "type": "parquet_string",
            "uniqueName": "c_address",
            "label": "c_address",
            "parentObject": "customer_tgt.parquet",
            "pcType": "NSTRING",
            "precision": 4000,
            "scale": 0,
            "columnIndex": -1,
            "isKey": false,
            "isExternalId": false,
            "isSfIdLookup": false,
            "isNullable": true,
            "isUnique": false,
            "isCreateable": false,
            "isUpdateable": true,
            "isFilterable": true,
            "isCalculated": false,
            "javaType": "java.lang.String",
            "showLabel": true,
            "naturalOrder": 2,
            "linkedFields": [],
            "relatedInfos": [],
            "references": []
        },
        {
            "@type": "field",
            "id": -1,
            "name": "c_nationkey",
            "type": "parquet_int64",
            "uniqueName": "c_nationkey",
            "label": "c_nationkey",
            "parentObject": "customer_tgt.parquet",
            "pcType": "BIGINT",
            "precision": 19,
            "scale": 0,
            "columnIndex": -1,
            "isKey": false,
            "isExternalId": false,
            "isSfIdLookup": false,
            "isNullable": true,
            "isUnique": false,
            "isCreateable": false,
            "isUpdateable": true,
            "isFilterable": true,
            "isCalculated": false,
            "javaType": "java.math.BigInteger",
            "showLabel": true,
            "naturalOrder": 3,
            "linkedFields": [],
            "relatedInfos": [],
            "references": []
        },
        {
            "@type": "field",
            "id": -1,
            "name": "FileName",
            "type": "string",
```

```
            "uniqueName": "FileName",
            "label": "FileName",
            "parentObject": "customer_tgt.parquet",
            "pcType": "NSTRING",
            "precision": 1024,
            "scale": 0,
            "columnIndex": -1,
            "isKey": false,
            "isExternalId": false,
            "isSfIdLookup": false,
            "isNullable": false,
            "isUnique": false,
            "isCreateable": false,
            "isUpdateable": true,
            "isFilterable": true,
            "isCalculated": false,
            "javaType": "java.lang.String",
            "showLabel": true,
            "naturalOrder": 8,
            "linkedFields": [],
            "relatedInfos": [],
            "references": []
        }
    ]
```

fileRecord

Use this resource to upload a Visio template XML file or image file to your organization. You can also use this resource to delete a Visio template XML file or image file from the organization.

POST Request

To upload a Visio template XML file or image file, use the following URI.

```
/api/v2/fileRecord
```

You can upload a file up to 5 MB in size.

Note: Encode the request body as multipart/form-data.

Use the following attributes in the request body:

Field	Type	Required	Description
file		Yes	Content of the file that you want to upload. File content should be in binary format, UTF-8 encoding.
type	String	Yes	Type of file that you want to upload. Use one of the following values: - MAPPING. Use to upload a Visio template XML file. Use for XML files only. - IMAGE. Use to update an image file for a Visio template. Use for JPEG or PNG files only.

In addition to the POST attributes, pass the following information in the request body:

- Boundary value. Used to define different parts of the request body.

- File name. The file name of the content you want to upload.

- icSessionId. Informatica Intelligent Cloud Services session ID returned by the login resource. You can pass this information in the request body for clients that do not allow custom headers. If you can pass icSessionId as part of the request header, you can omit this information in the request body.

Use the following template for the fileRecord POST request:

```
URL: <serverUrl>/api/v2/fileRecord/
HTTP method: POST

Content-Type:multipart/form-data;boundary=<boundary value>
--<boundary value>
Content-Disposition:form-data; name="file";filename="<filename.ext>";Content-Type:text/
<xml|json>

<content of the file you want to upload encoded as UTF-8>

--<boundary value>
Content-Disposition: form-data; name="type"

<MAPPING | IMAGE>
--<boundary value>
Content-Disposition: form-data; name="icSessionId"

<icSessionID returned from login resource>
--<boundary value>--
```

POST Response

Returns the fileRecord object if the upload is successful. Returns the error object if errors occur.

The fileRecord object includes the following attributes:

Field	Type	Description
id	String	ID for the uploaded file. You can use this ID to identify the file when you create or update a Visio template with the masterTemplate resource.
orgId	String	Organization ID.
name	String	File name.
description	String	Description of the file.
createTime	Date/time	Time that the file was uploaded to the organization.
updateTime	Date/time	Last time that the file was updated.
createdBy	String	User who first uploaded the file.
updatedBy	String	User who last updated the file.
type	String	File type.
size	Int	File size.
attachTime	String	Time the file was associated with a Visio template.

DELETE Request

You can delete a Visio template XML or image file if the Visio template is not used by a Visio template.

To delete a file, use the file ID in the following URI:

```
/api/v2/fileRecord/<id>
```

DELETE Response

Returns the 200 response code if the request is successful.

Returns the error object if errors occur.

POST Example

To upload the VisioTemplate.xml file with an icSessionId of IV4wOrJmd6YUtmKa8t, you might use the following request. XML data should be encoded in UTF-8.

```
URL: https://example.informatica.com/saas/api/v2/fileRecord/
HTTP method: POST

Content-Type:multipart/form-data;boundary=243553118520053
--243553118520053
Content-Disposition:form-data; name="file";filename="<VisioTemplate.xml>";Content-
Type:text/xml

<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE Graph SYSTEM "graph.dtd">
<Graph Name="" Description="" UItype="" DlgSize="">
  <Groups />
  <Parameters>
    <Parameter Name="$EXTERNALID__C$" Label="" LabelWidth="" IsMandatory="True"
DefaultValue="" Control="" Data="" Description="" />
    <Parameter Name="$TGT$" Label="" LabelWidth="" IsMandatory="True" DefaultValue=""
Control="Combo_Ctrl" Data="Targets" Description="" />
    <Parameter Name="$GroupBy$" Label="" LabelWidth="" IsMandatory="True"
DefaultValue="" Control="" Data="" Description="" />
    <Parameter Name="$o_PERCENT_FLD__C$" Label="" LabelWidth="" IsMandatory="True"
DefaultValue="" Control="" Data="" Description="" />
    <Parameter Name="$SRC$" Label="" LabelWidth="" IsMandatory="True" DefaultValue=""
Control="Combo_Ctrl" Data="Sources" Description="" />
  </Parameters>
  <Node NameID="Source Definition" Name="$SRC$" Reusable="" Type="Source Definition"
InstanceName="$SRC$" Description="" isParameterized="True">
    <CustomProperty Name="Source Table" Value="$SRC$" isParameterized="True" />
    <CustomProperty Name="Database Name" Value="" isParameterized="False" />
    <CustomProperty Name="Owner Name" Value="" isParameterized="False" />
    <CustomProperty Name="Business Name" Value="" isParameterized="False" />
    <CustomProperty Name="Database Type" Value="" isParameterized="False" />
    <CustomProperty Name="Is ShortCut" Value="False" isParameterized="False" />
  </Node>
.
.
.
  <Link Name="Sheet.7" FromNameID="Aggregator" ToNameID="Target Definition"
MasterInputSet="False" isParameterized="False">
    <Rule Text="Datatype:string" isParameterized="False" />
    <Rule Text="EXCLUDE Named:AUTO__C (TO) AUTO__C" isParameterized="False" />
    <Rule Text="Datatype:date/time" isParameterized="False" />
    <Rule Text="Pattern:_o$" isParameterized="False" />
    <Rule Text="Datatype:nstring" isParameterized="False" />
    <Rule Text="Datatype:ntext" isParameterized="False" />
    <Rule Text="Datatype:text" isParameterized="False" />
  </Link>
</Graph>

--243553118520053
Content-Disposition: form-data; name="type"

MAPPING
--243553118520053
Content-Disposition: form-data; name="icSessionId"

IV4wOrJmd6YUtmKa8t
--243553118520053--
```

If the upload is successful, returns the fileRecord response object.

filelisteners

Use the filelisteners resource to create, update, delete, run a file listener, and change the owner of a file listener association from one user to another. Informatica Intelligent Cloud Services can use file listeners to monitor specific folders. Informatica Intelligent Cloud Services are notified by using a call-back API when new files arrive at a monitored folder and when files in the folder are updated or deleted.

Consider the following when you use the filelisteners resource:

- Use JSON format.
- Use the following base URL:

```
<serverUrl>/mftsaas/api/v1/<API name>
```

- Use the following request header format:

```
<METHOD> <serverUrl>/<URI> HTTP/<HTTP version>
Content-Type: application/json
Accept: application/json
IDS-SESSION-ID: <SessionId>
```

Note: If you use a tool such as Postman that automatically includes the HTTP version, do not enter the HTTP version in the URL. If the HTTP version appears twice in the URL, the request fails.

Complete the following tasks to run and monitor file listeners:

- Send a GET request to view details of a file listener. See "View file listener" on page 329.
- Send a POST request to create a file listener. See "Create a file listener" on page 334.
- Send a PUT request to update an existing file listener. See "Update a file listener" on page 339.
- Send a start POST request to start a file listener job. See "Start a file listener" on page 344.
- Send a stop POST request to stop the file listener job manually. See "Stop a file listener" on page 345.
- Send a POST request to change the owner a file listener association. See "Change owner of a file listener association" on page 347.

View file listener

Use the GET request to view file listeners details. You can view the details for a particular file listener or view details for all file listeners in your organization.

GET request

To view the details of a particular file listener, include the file listener ID in the following URI:

```
Get <serverUrl>/mftsaas/api/v1/filelisteners/<filelistener ID>
```

To view the details for all of the file listeners in the organization, omit the file listener ID.

GET response

A request for file listener details returns the following information:

Field	Type	Description
id	String	ID number associated with the file listener.
name	String	Name of the file listener.
description	String	Description of the file listener.
status	String	Status of the file listener. - `enabled`. Listens to files on the designated folder. `disabled`. Does not listen to files on the designated folder.
agentGroup	Numeric	Runtime environment that contains the Secure Agent used to run the file listener.
type	String	Type of the connection to which the file listener listens.
connection	String	Connection to which the file listener listens.
folderPath	String	Path to the folder on the connection to which the file listener listens.
filePattern	String	File name pattern to which the file listener listens.
Post Action	String	Determines the action the file listener must perform after the file listener listens to the events. You can select the post action as **Delete** only if the file pattern is an indicator file. Default is None. The following connection types support the Post Action option: - Local folder - Advanced FTP V2 - Advanced FTPS V2 - Advanced SFTP V2 - Azure Data Lake Store Gen2

patternType	String	The file pattern.
		- wildcard. Use wildcard patterns of file name.
		- regex. Use regular expression to match the file pattern. Consider the following examples:
		  - Use the following syntax to listen to all files except for files with a name that contains out, foo, and baz: `^(?!.*(?:out|baz|foo)).*$ à all except`
		  - Use the following syntax to listen to all files with doc and docx, pdf extensions: `([a-zA-Z0-9\s_\ \.\-\(\):])+(.doc|.docx|.pdf)$ à`
		- indicator file. Use the file name to which the file listener listens.

mandatory	String	Defines whether rule values are mandatory.

Field	Type	Description
recursive	String	Defines whether rule values are recursive.
type	String	Frequency at which the file listener runs, daily, weekly, or monthly.
timezone	String	Time zone that refers to the start and end time.
startDate	Date/Time	Date on which the file listener starts running.
endDate	Date/Time	Date until which the file listener runs.
runIndefinitely	String	Whether the file listener runs without an end date.
startsAt	Date/Time	Time of day when the file listener starts running.
endsAt	Date/Time	Time of day when the file listener stops running.
frequency	Numeric	Frequency at which the file listener checks for files in the folder.
frequencyUnit	String	Unit of frequency to which file listener checks for files in the folder, by seconds, minutes, or hours.
listenerEvents	String	Determines when the file listener sends notifications to the services that are registered to it. Response to each event when the event is set to true is as follows: - arrive. Send notifications when files arrive at the folder to which the file listener listens. - update. Send notifications when files in the folder to which the file listener listens are updated. - delete. Send notifications when files in the folder to which the file listener listens are deleted.
stopWhenRulesMet	String	Whether the file listener stops listening to the folder when the listener rules are met. Set to one of the following values: - `false`. The file listener notifies the registered application on events and continues to listen for subsequent events. - `true`. The file listener stops listening to the folder when the first event of file deletion occurs in the folder.
checkFileStability	String	Enter one of the following values. - `false`.The file listener does not verify whether the entire file is copied to the folder before notifying the registered services. - `true`. The file listener verifies whether the entire file is copied to the folder before notifying the registered services. Default is `true`.

Field	Type	Description
stabilityCheckInterval	Time	Time in seconds that a file listener waits to check for file stability. You can specify a value in the stabilityCheckInterval field only if the checkFileStability option is set to `true`.
notifyExistingFiles	String	The first time the file listener runs, it sends a notification if files exist in the folder to which it listens, of the parameter is set to true.
excludeFileEventsWhenNotRunning	String	Determines if you want to exclude file events that occur when a file listener is not running.
continueOnError	String	Determines if you want the file listener to continue to retry and run in case of failures, such as temporary network disruption.
location	String	Location of the project folder that contains the file listener component.
createTime	Date/Time	Time when the component was created.
lastupdateTime	Date/Time	Time when the component was last updated.

GET response example for one file listener

If your request to view file listener details of the file listener with ID `eX5qlosUfEHbwvNwGpRwQd` is successful, you might receive a response similar to the following example:

```
{
    "id": "eX5qlosUfEHbwvNwGpRwQd",
    "name": "FL512087",
    "description": "Demo",
    "status": "ENABLE",
    "agentGroup": "01000025000000000002",
    "connection": {
        "type": "local",
        "name": "",
        "connId": ""
    },
    "rules": [
        {
            "id": 10052,
            "folderPath": "C:\\temp1",
            "filePattern": "*.txt",
        "postAction": "NONE",
            "patternType": "wildcard",
            "mandatory": false,
            "recursive": false
        }
    ],
    "scheduleDefinition": {
        "type": "DAILY_WITH_INTERVAL",
        "timezone": "IST",
        "startDate": "20181227",
        "endDate": "20181227",
        "runIndefinitely": false,
        "startsAt": "1015",
        "endsAt": "2355",
        "frequency": 15,
        "frequencyUnit": "SECONDS"
    },
```

```
        "stopWhenRulesMet": false,
        "listenerEvents": {
            "arrive": true,
            "update": true,
            "delete": true
        },
        "checkFileStability": true,
        "stabilityCheckInterval": 10,
        "notifyExistingFiles": false,
        "excludeFileEventsWhenNotRunning": true,
        "continueOnError": true,
        "location": {
            "folderId": "avVCKODMM0RdSmcNWDnrKi",
            "folderName": "New",
            "projectId": "3iWWHkLbM2giVppBmJmZgV",
            "projectName": "Default"
        },
        "createTime": "2019-02-12T07:03:49Z",
        "lastUpdatedTime": "2019-02-12T07:03:49Z"
}
```

Response example to view all file listeners

If your request to view file listener details is successful, you might receive a response similar to the following example:

```
{
    "listeners": [
        {
            "id": "8h9hng2kRokf2Db6Xb4pA8",
            "name": "dfgdfg",
            "description": "",
            "status": "ENABLE",
            "stopWhenRulesMet": false,
        "checkFileStability": false,
        "notifyExistingFiles": false,
        "excludeFileEventsWhenNotRunning": true,
        "continueOnError": true,
        "location": {
                "projectId": "3iWWHkLbM2giVppBmJmZgV",
                "projectName": "Default"
            },
            "createTime": "2019-01-28T05:31:00Z",
            "lastUpdatedTime": "2019-01-28T05:31:00Z"
        },
        {
            "id": "bQdKQmGlFUUgS85AevLkqi",
            "name": "FL123",
            "description": "xsdfsdfsdf",
            "status": "ENABLE",
            "stopWhenRulesMet": false,
        "checkFileStability": true,
        "stabilityCheckInterval": 10,
        "notifyExistingFiles": false,
        "excludeFileEventsWhenNotRunning": true,
        "continueOnError": true,
        "location": {
                "projectId": "3iWWHkLbM2giVppBmJmZgV",
                "projectName": "Default"
            },
            "createTime": "2019-01-24T05:20:26Z",
            "lastUpdatedTime": "2019-01-25T06:52:40Z"
        },
        {
            "id": "eX5qlosUfEHbwvNwGpRwQd",
            "name": "FL512087",
            "description": "Demo",
            "status": "ENABLE",
            "stopWhenRulesMet": false,
        "checkFileStability": true,
        "stabilityCheckInterval": 10,
```

```
            "notifyExistingFiles": false,
            "excludeFileEventsWhenNotRunning": true,
            "continueOnError": true,
            "location": {
                        "folderId": "avVCKODMM0RdSmcNWDnrKi",
                        "folderName": "New",
                        "projectId": "3iWWHkLbM2giVppBmJmZgV",
                        "projectName": "Default"
                },
                "createTime": "2019-02-12T07:03:49Z",
                "lastUpdatedTime": "2019-02-12T07:03:49Z"
        }
      ]
    }
```

If the request to view all file listeners is unsuccessful, you might receive a response similar to the following example:

```
File Listener not found (403 Forbidden)
{    "responseCode": "NOT_FOUND",    "message": "File Listener with id
'eX5qlosUfEHbwvNwGpRwQd1' not found." }
```

Create a file listener

Use a POST request to create a file listener and an event listener.

POST request

Use the following URI to create a file listener and an event listener:

```
POST <serverUrl>/mftsaas/api/v1/filelisteners
```

Use the following fields in the POST request:

Field	Type	Required	Description
name	String	Yes	Name of the file listener.
description	String	-	Description of the file listener.
status	String	Yes	Status of the file listener. - `enabled`. Listens to files on the designated folder. - `disabled`. Does not listen to files on the designated folder.
agentGroup	Numeric	Yes	Runtime environment that contains the Secure Agent used to run the file listener.
connectionType	String	Yes	Type of the connection to which the file listener listens.
connection	String	Yes	Connection to which the file listener listens.
folderPath	String	Yes	Path to the folder on the connection to which the file listener listens.
filePattern	String	Yes	File name pattern to which the file listener listens.

Field	Type	Required	Description
Post Action	String	-	Determines the action the file listener must perform after the file listener listens to the events. You can select the post action as **Delete** only if the file pattern is an indicator file. Default is None. The following connection types support the Post Action option: - Local folder - Advanced FTP V2 - Advanced FTPS V2 - Advanced SFTP V2 - Azure Data Lake Store Gen2

patternType	String	Yes	The file pattern.
			- wildcard. Use wildcard patterns of file name.
			- regex. Use regular expression to match the file pattern. Consider the following examples:
			  - Use the following syntax to listen to all files except for files with a name that contains out, foo, and baz: `^(?!.*(?:out|baz|foo)).*$` à all except
			  - Use the following syntax to listen to all files with doc and docx, pdf extensions: `([a-zA-Z0-9\s_ \\.\-\(\):])+(.doc|.docx|.pdf)$` à
			- Indicator File. Use the file name to which the file listener listens.

mandatory	String	-	Defines whether rule values are mandatory.
recursive	String	-	Defines whether rule values are recursive.
scheduleDefinition	String	Yes	Defines the frequency in which the file listener must run.
type	String	Yes	Frequency at which the file listener runs, daily, weekly, or monthly.
timezone	String	Yes	Time zone that refers to the start and end time.
startDate	Date/Time	Yes	Date on which the file listener starts running.
endDate	Date/Time	Yes	Date until which the file listener runs.
runIndefinitely	String	-	The file listener runs without an end date.
startsAt	Date/Time	Yes	Time of day when the file listener starts running.
endsAt	Date/Time	Yes	Time of day when the file listener stops running.
frequency	Numeric	Yes	Frequency at which the file listener checks for files in the folder.

Field	Type	Required	Description
frequencyUnit	String	Yes	Unit of frequency to which file listener checks for files in the folder, by seconds, minutes, or hours.
listenerEvents	String	Yes	Determines when the file listener sends notifications to the services that are registered to it. Response to each event, when the event is set to true is as follows: - arrive. Send notifications when files arrive at the folder to which the file listener listens. - update. Send notifications when files in the folder to which the file listener listens are updated. - delete. Send notifications when files in the folder to which the file listener listens are deleted.
stopWhenRulesMet	String	-	The file listener stops listening to the folder when the listener rules are met. - `false`. The file listener notifies the registered application on events and continues to listen for subsequent events. - `true`. The listener stops listening to the folder when the first event of file deletion occurs in the folder.
checkFileStability	String	-	The file listener verifies that the entire file is copied to the folder before notifying the registered services.
stabilityCheckInterval	Time	-	Time in seconds that a file listener waits to check for file stability. You can specify a value in the stabilityCheckInterval field only if the checkFileStability option is set to `true`.
notifyExistingFiles	String	-	The first time the file listener runs, it sends a notification if files exist in the folder to which it listens.
excludeFileEventsWhenNotRunning	String	-	Determines if you want to exclude file events that occur when a file listener is not running.
continueOnError	String	-	Determines if you want the file listener to continue to retry and run in case of failures, such as temporary network disruption.
emailIds	String	-	List of email addresses to send notifications if the file listener fails. Use commas to separate email addresses in the list.
location	String	-	Location of the project folder that contains the file listener component.

POST request example

Use this sample as a reference to create a file listener.

```
POST <serverUrl>/mftsaas/api/v1/filelisteners
Content-Type: application/json
Accept:application/json
Content-Type:application/json
```

```
IDS-SESSION-ID:{{IDS-SESSION-ID}}
{
    "name": "{{NEWFILELISTENER-NAME}}",
    "description": "Demo",
    "status": "ENABLE",
    "location": {
        "folderId": "avVCKODMM0RdSmcNWDnrKi",
        "folderName": "New",
        "projectId": "3iWWHkLbM2giVppBmJmZgV",
        "projectName": "Default"
    },
    "agentGroup": "01000025000000000002",
    "connection": {
        "type": "local",
        "name": "",
        "connId": "",
        "local": true
    },
    "listenerEvents":{
    "arrive":true,
    "update":true,
    "delete":true},
    "checkFileStability": true,
    "stabilityCheckInterval": 10,
    "notifyExistingFiles": false,
    "excludeFileEventsWhenNotRunning": true,
    "continueOnError": true,
    "emailIDs":"test@gmail.com,infa@hotmail.com",
        "rules": [
        {
            "id": 10070,
            "folderPath": "C:\\temp1",
            "patternType":"wildcard",
            "filePattern": "*.txt",
        "postAction": "NONE",
            "mandatory": false,
            "recursive": false
        }
    ],
  "scheduleDefinition": {
        "type": "DAILY_WITH_INTERVAL",
        "timezone": "IST",
        "startDate": "20181227",
        "endDate": "20181227",
        "runIndefinitely": false,
        "startsAt": "1015",
        "endsAt": "2355",
        "frequency": 15,
        "frequencyUnit": "SECONDS",
        "dayOfMonth": 0
    },
    "stopWhenRulesMet": false

}
```

POST response example

If the post request is successful, you might receive a response similar to the following example:

```
{
    "id": "eX5qlosUfEHbwvNwGpRwQd",
    "name": "FL512087",
    "description": "Demo",
    "status": "ENABLE",
    "agentGroup": "01000025000000000002",
    "connection": {
        "type": "local",
        "name": "",
        "connId": ""
    },
    "rules": [
```

```
                {
                    "id": 10070,
                    "folderPath": "C:\\temp1",
                    "filePattern": "*.txt",
                    "patternType": "wildcard",
                "postAction": "NONE",
                    "mandatory": false,
                    "recursive": false
                }
        ],
        "scheduleDefinition": {
            "type": "DAILY_WITH_INTERVAL",
            "timezone": "IST",
            "startDate": "20181227",
            "endDate": "20181227",
            "runIndefinitely": false,
            "startsAt": "1015",
            "endsAt": "2355",
            "frequency": 15,
            "frequencyUnit": "SECONDS",
            "dayOfMonth": 0
        },
        "stopWhenRulesMet": false,
        "listenerEvents": {
            "arrive": true,
            "update": true,
            "delete": true
        },
            "checkFileStability": true,
            "stabilityCheckInterval": 10,
            "notifyExistingFiles": false,
        "excludeFileEventsWhenNotRunning": true,
        "continueOnError": true,
        "emailIDs":"test@gmail.com,infa@hotmail.com",
            "location": {
            "folderId": "avVCKODMM0RdSmcNWDnrKi",
            "folderName": "New",
            "projectId": "3iWWHkLbM2giVppBmJmZgV",
            "projectName": "Default"
        }
    }
```

POST request example

Use this sample as a reference to create an event listener.

```
POST <serverUrl>/public/core/v1/filelisteners
Content-Type: application/json
Accept:application/json
Content-Type:application/json
IDS-SESSION-ID:{{IDS-SESSION-ID}}
{
    "name": "{{NEWEVENTLISTENER-NAME}}",
    "description": "",
    "agentGroup": "01000025000000000003",
    "sourceType": "Server",
    "location": {
        "projectId": "1UNDIQkHQYKcNLPdxeR56p",
        "projectName": "overRide"
    },
    "eventProvider": "AS2",
    "eventType": "as2_message_receive_failed",
    "rules": [
        {
            "key": "event.userName",
            "value": "Suraj",
            "operator": "NONE",
            "type": "CONTAINS"
        },
        {
            "key": "event.fileName",
```

```
                    "value": "Test",
                    "operator": "AND",
                    "type": "STRING_EQUALS"
                },
                {
                    "key": "event.fileSize",
                    "value": "89",
                    "operator": "OR",
                    "type": "INTEGER_EQUALS"
                }
            ]
        }
```

POST response example

If the post request is successful, you might receive a response similar to the following example:

```
{
    "id": "f11rC9Kwa0UlOeg2TIjBks",
    "name": "EventFL684930",
    "description": "",
    "agentGroup": "01000025000000000003",
    "sourceType": "Server",
    "location": {
        "projectId": "1UNDIQkHQYKcNLPdxeR56p",
        "projectName": "overRide"
    },
    "createTime": "2020-04-06T05:25:55Z",
    "lastUpdatedTime": "2020-04-06T05:25:55Z",
    "eventProvider": "AS2",
    "eventType": "as2_message_receive_failed",
    "rules": [
        {
            "key": "event.userName",
            "value": "Suraj",
            "operator": "NONE",
            "type": "CONTAINS"
        },
        {
            "key": "event.fileName",
            "value": "Test",
            "operator": "AND",
            "type": "STRING_EQUALS"
        },
        {
            "key": "event.fileSize",
            "value": "89",
            "operator": "OR",
            "type": "INTEGER_EQUALS"
        }
    ]
}
```

Update a file listener

Use a PUT request to update a file listener.

PUT Request

Use the following URI to update an existing file listener.

```
PUT <server URL>/mftsaas/api/v1/filelisteners/<filelistener ID>
```

Use the following fields in the PUT request:

Field	Type	Required	Description
id	String	Yes	ID number associated with the file listener.
name	String	Yes	Name of the file listener.
description	String	-	Description of the file listener.
status	String	Yes	Status of the file listener. - `enabled`. Listens to files on the designated folder. - `disabled`. Does not listen to files on the designated folder.
agentGroup	Numeric	Yes	Runtime environment that contains the Secure Agent used to run the file listener.
connectionType	String	Yes	Type of the connection to which the file listener listens.
connection	String	Yes	Connection to which the file listener listens.
folderPath	String	Yes	Path to the folder on the connection to which the file listener listens.
filePattern	String	Yes	File name pattern to which the file listener listens.
Post Action	String	-	Determines the action the file listener must perform after the file listener listens to the events. You can select the post action as **Delete** only if the file pattern is an indicator file. Default is None. The following connection types support the Post Action option: - Local folder - Advanced FTP V2 - Advanced FTPS V2 - Advanced SFTP V2 - Azure Data Lake Store Gen2

patternType	String	Yes	The file pattern.
			- wildcard. Use wildcard patterns of file name.
			- regex. Use regular expression to match the file pattern. Consider the following examples:
			  - Use the following syntax to listen to all files except for files with a name that contains out, foo, and baz: `^(?!.*(?:out|baz|foo)).*$` à all except
			  - Use the following syntax to listen to all files with doc and docx, pdf extensions: `([a-zA-Z0-9\s_ \\.\-\(\):])+(.doc|.docx|.pdf)$` à

mandatory	String	-	Defines whether rule values are mandatory.
recursive	String	-	Defines whether rule values are recursive.
scheduleDefinition	String	Yes	Defines the frequency in which the file listener must run.

Field	Type	Required	Description
type	String	Yes	Frequency at which the file listener runs, daily, weekly, or monthly.
timezone	String	Yes	Time zone that refers to the start and end time.
startDate	Date/Time	Yes	Date on which the file listener starts running.
endDate	Date/Time	Yes	Date until which the file listener runs.
runIndefinitely	String	-	The file listener runs without an end date.
startsAt	Date/Time	Yes	Time of day when the file listener starts running.
endsAt	Date/Time	Yes	Time of day when the file listener stops running.
frequency	Numeric	Yes	Frequency at which the file listener checks for files in the folder.
frequencyUnit	String	Yes	Unit of frequency to which file listener checks for files in the folder, by seconds, minutes, or hours.
listenerEvents	String	Yes	Determines when the file listener sends notifications to the services that are registered to it. Response to each event, when the event is set to true is as follows: - arrive. Send notifications when files arrive at the folder to which the file listener listens. - update. Send notifications when files in the folder to which the file listener listens, are updated. - delete. Send notifications when files in the folder to which the file listener listens are deleted.
stopWhenRulesMet	String	-	The file listener stops listening to the folder when the listener rules are met. - `false`. The file listener notifies the registered application on events and continues to listen for subsequent events. - `true`. The listener stops listening to the folder when the first event of file deletion occurs in the folder.
checkFileStability	String	-	The file listener verifies that the entire file is copied to the folder before notifying the registered services.
stabilityCheckInterval	Time	-	Time in seconds that a file listener waits to check for file stability. You can specify a value in the stabilityCheckInterval field only if the checkFileStability option is set to `true`.

Field	Type	Required	Description
notifyExistingFiles	String	-	The first time the file listener runs, it sends a notification if files exist in the folder to which it listens.
excludeFileEventsWhenNotRunning	String	-	Determines if you want to exclude the file events that occur when a file listener is not running.
continueOnError	String	-	Determines if you want the file listener to continue to retry and run in case of a failure, such as temporary network disruption.
emailIds	String	-	List of email addresses to send notifications if the file listener fails. Use commas to separate email addresses in the list.
location	String	-	Location of the project folder.

PUT request example

Use this sample as a reference to update the file listener.

```
PUT <serverUrl>/public/core/v1/filelisteners
Content-Type: application/json
Accept:application/json
Content-Type:application/json
IDS-SESSION-ID:{{IDS-SESSION-ID}}
{
    "description": "Demo",
    "status": "ENABLE",
    "location": {
        "folderId": "avVCKODMM0RdSmcNWDnrKi",
        "folderName": "New",
        "projectId": "3iWWHkLbM2giVppBmJmZgV",
        "projectName": "Default"
    },
    "agentGroup": "01000025000000000002",
    "connection": {
        "type": "local",
        "name": "",
        "connId": "",
        "local": true
    },
    "listenerEvents":{
    "arrive":true,
    "update":true,
    "delete":true},
    "checkFileStability": true,
    "stabilityCheckInterval": 10,
    "notifyExistingFiles": false,
"excludeFileEventsWhenNotRunning": true,
"continueOnError": true,
"emailIDs":"test@gmail.com,infa@hotmail.com"
    "rules": [
        {
            "id": 10070,
            "folderPath": "C:\\temp1",
            "patternType":"wildcard",
            "filePattern": "*.txt",
            "postAction": "NONE",
            "mandatory": false,
            "recursive": false
        }
```

```
        ],
    "scheduleDefinition": {
        "type": "DAILY_WITH_INTERVAL",
        "timezone": "IST",
        "startDate": "20181227",
        "endDate": "20181227",
        "runIndefinitely": false,
        "startsAt": "1015",
        "endsAt": "2355",
        "frequency": 15,
        "frequencyUnit": "SECONDS",
        "dayOfMonth": 0
    },
    "stopWhenRulesMet": false

}
```

PUT response

If the request to update the file listener is successful, you receive the response similar to the following example:

```
{
    "id": "eX5qlosUfEHbwvNwGpRwQd",
    "name": "FL512087",
    "description": "Demo",
    "status": "ENABLE",
    "agentGroup": "01000025000000000002",
    "connection": {
        "type": "local",
        "name": "",
        "connId": ""
    },
    "rules": [
        {
            "id": 10070,
            "folderPath": "C:\\temp1",
            "filePattern": "*.txt",
            "patternType": "wildcard",
            "postAction": "NONE",
            "mandatory": false,
            "recursive": false
        }
    ],
    "scheduleDefinition": {
        "type": "DAILY_WITH_INTERVAL",
        "timezone": "IST",
        "startDate": "20181227",
        "endDate": "20181227",
        "runIndefinitely": false,
        "startsAt": "1015",
        "endsAt": "2355",
        "frequency": 15,
        "frequencyUnit": "SECONDS",
        "dayOfMonth": 0
    },
    "stopWhenRulesMet": false,
    "listenerEvents": {
        "arrive": true,
        "update": true,
        "delete": true
    },
        "checkFileStability": true,
        "stabilityCheckInterval": 10,
        "notifyExistingFiles": false,
    "excludeFileEventsWhenNotRunning": true,
    "continueOnError": true,
    "emailIDs":"test@gmail.com,infa@hotmail.com"
    "location": {
        "folderId": "avVCKODMM0RdSmcNWDnrKi",
```

```
                        "folderName": "New",
                        "projectId": "3iWWHkLbM2giVppBmJmZgV",
                        "projectName": "Default"
                }
        }
```

If the request to update the file listener is unsuccessful, you might receive an error similar to the following example:

```
{
 "responseCode": "NOT_FOUND",
"message": "File Listener with id 'eX5qlosUfEHbwvNwGpRwQd1' not found."
}
```

Delete a file listener

Use a DELETE request to delete a file listener.

DELETE request

Use the following URI to delete a file listener:

```
DELETE <server URL>/mftsaas/api/v1/filelisteners/<file listener ID>
```

Use the following fields in the delete request:

Field	Type	Description
file listener ID	String	ID number associated with the file listener.

Delete response example

If the delete request is unsuccessful, you receive a response similar to the following example:

```
{
 "responseCode": "NOT_FOUND",
"message": "Document Artifact with Id = bQdKQmGlFUUgS85AevLkqi3 not found."
}
```

Start a file listener

Use a POST request to start a file listener job.

POST request

To start a file listener, use the following URI:

```
POST <server URL>/mftsaas/api/v1/filelisteners/<file listener ID>/start
```

Include the following field in the request:

Field	Type	Description
file listener ID	String	ID number associated with the file listener.

POST response example

If the request to start the file listener is successful, you might receive a response similar to the following example:

```
{
        "status": "STARTED",
```

```
        "jobId": 1038
    }
```

POST error response example

If the request to start the file listener is unsuccessful, you might receive a response similar to the following example:

```
{
 "responseCode": "NOT_FOUND",
"message": "File listener not found for ListenerId:  bQdKQmGlFUUgS85AevLkqisd"
}
```

Response : Agent down (403 Forbidden)

```
{
 "responseCode": "NOT_FOUND",   "message": "Agent \"01000008000000000002\" in Agent
Group \"01000025000000000002\" is not accessible."}
```

Stop a file listener

Use a POST request to stop a file listener job.

POST request

To stop a file listener, use the following URI:

```
POST <server URL>/mftsaas/api/v1/filelisteners/<file listener ID>/stop
```

Include the following field in the request:

Field	Type	Description
fileListenerID	String	ID number associated with the file listener.

POST response example

If the request to stop the file listener is successful, you might receive a response similar to the following example:

```
{
    "status": "STOPPED",
    "jobId": 1038
}
```

POST response example

If the request to stop the file listener is unsuccessful, you might receive a response similar to the following example:

```
{
 "responseCode": "NOT_FOUND",
"message": "File listener not found for ListenerId:  bQdKQmGlFUUgS85AevLkqisd"
}
```

Response : Agent down (403 Forbidden)

```
{
 "responseCode": "NOT_FOUND",   "message": "Agent \"01000008000000000002\" in Agent
Group \"01000025000000000002\" is not accessible."}
```

View the status of a file listener

Use a GET request to request the status of a file listener job.

GET request

To view the status of a file listener, use the following URI:

```
GET <server URL>/mftsaas/api/v1/filelisteners/job/<Job ID>/status
```

Include the following field in the request:

Field	Type	Description
Job ID	String	ID number associated with the file listener job.

GET response example

If the request to view the status of a file listener job is successful, you might receive a response similar to the following example:

```
{
    "status": "STOPPED",
    "jobId": 1038
}
```

GET response example

If the request to view the status of a file listener job is unsuccessful, you might receive a response similar to the following example:

```
{
    "responseCode": "NOT_FOUND",
    "message": "File listener not found for TaskId:  1079"
}
```

View the details of a file listener job

Use a GET request to view the details of a file listener job.

GET request

To view the details of a file listener job, use the following URI:

```
GET <server URL>/mftsaas/api/v1/filelisteners/<Run ID>/activityLog
```

Include the following field in the request:

Field	Type	Description
Run ID	String	ID number associated with the file listener job.

GET response example

If the request to view the details of a file listener job is a success, you might receive a response similar to the following example:

```
{
    "instanceName": "FL_1-1006",
    "jobId": 1006
    "startTime": "2021-02-09T22:38:01Z",
    "updateTime": "2021-02-09T22:38:01Z",
    "endTime": "2021-02-09T22:38:01Z",
```

```
    "status": "Completed",
}
```

The responses vary based on the file listener status.

Change owner of a file listener association

A file listener can be associated with a taskflow or with file ingestion tasks. You can use file listener as a source or schedule(trigger) for file ingestion tasks and as a trigger for a taskflow. A file listener association is the internal link between file listener and its associated file ingestion tasks or taskflows. The owner of a file listener association is the last person who selected the file listener as a source or schedule in a file ingestion task or the last person who published the taskflow with the file listener as a trigger. You might want to change or transfer the owner of a file listener association when a user who owns the file listener association leaves the organization. For example, if the user Alex creates a file ingestion task, the user Beth creates a file listener, and the user Clara selects the file listener as a source or schedule in a file ingestion task, Clara is the owner of the file listener association. If Clara leaves the organization, you must reassign her assigned file listener associations to another user before you can delete her user account.

POST request

Use a POST request to change the owner of a file listener association from one user to an another.

Use the following URI to change the owner of a file listener association:

```
POST <serverUrl>/mftsaas/api/v1/ChangeCallbackOwnership
```

Use the following fields in the POST request:

Field	Type	Required	Description
currentOwner	String	Yes	User ID of the current owner.
newOwner	String	Yes	User ID of the new owner.

POST request example

Use this sample as a reference to change the owner of a file listener association.

```
POST <serverUrl>/mftsaas/api/v1/ChangeCallbackOwnership
Content-Type: application/json
Accept:application/json
Content-Type:application/json
IDS-SESSION-ID:{{IDS-SESSION-ID}}
{
    "currentOwner": "{{UserIDwhoLeftTheOrganization}}",
    "newOwner": "{{NewUserID}}"
}
```

POST response example

If the post request is successful, you might receive the following example:

```
200 OK
```

File transfer

You can send files to a remote server or receive files from a remote server, and get the job status through the REST API.

Use the following resources for file transfer:

- sendfiles. Use to send files to a remote server.

- receivefiles. Use to receive files from a remote server.

- job. Use to get the status of the supported file transfer jobs that are initiated using the sendfiles or receivefiles resource.

When you use these resources, note the following rules:

- Use JSON format.

- Use the following base URL:

      <serverUrl>/mftsaas/api/v1/<API name>

- Use the following request header format:

      <METHOD> <serverUrl>/<URI> HTTP/<HTTP version>
      Content-Type: application/json
      Accept: application/json
      IDS-SESSION-ID: <SessionId>

Note: You must have the appropriate connector license to send and receive files.

sendfiles

Use the sendfiles resource to transfer files to a remote server.

The following connection types use the sendfiles resource to transfer files to the remote server:

- AS2
- Advanced FTP V2
- Advanced FTPS V2
- Advanced SFTP V2

Before you construct a sendfiles request to transfer files, obtain the identifier of the connection that provides access to the server. To get the connection ID, you can send a GET request using the connection resource. The connection resource can return information for each of your organization's connections.

POST Request

To transfer files, include the connection ID in the following URI.

      mftsaas/api/v1/sendfiles/<connection ID>

Include the following information in the request:

Field	Type	Required	Description
targetConnectionType	String	Yes	Connection type. The supported connection types are: - Advanced FTP V2 - Advanced FTPS V2 - Advanced SFTP V2 - AS2
srcDirectoryPath	String	Yes	Directory path from where files are transferred.
tgtDirectoryPath	String	-	Directory path to where files are transferred. This option is available only for Advanced FTP V2, Advanced FTPS V2, and Advanced SFTP V2 connection types. Default is '/'.
srcFilePattern	String	Yes	Source file name pattern. Specify a file name pattern to identify which files to send. You can use the regular expression type.
deleteSourceFiles	String	-	Whether to delete source files after a successful POST request. Use one of the following values: - true. Delete source files. - false. Save source files. Default is true.

For example, to transfer the files that begin with "file_" that are located in the workspace directory, you might use the following request:

```
POST <serverUrl>/mftsaas/api/v1/sendfiles/<connection ID>
Accept:application/json
IDS-SESSION-ID: <icSessionId or INFA-SESSION-ID>
{
    "targetConnectionType":   "as2",
    "srcDirectoryPath":  "C:\\server\\userdata\\workspace",
    "srcFilePattern":  "file_*"
}
```

For example, to transfer the files with ".txt" pattern, you might use the following request:

```
POST <serverUrl>/mftsaas/api/v1/sendfiles/<connection ID>
Accept:application/json
IDS-SESSION-ID: <icSessionId or INFA-SESSION-ID>
{
    "targetConnectionType":   "Advanced SFTP V2",
    "srcDirectoryPath":  "C:\\docstoreLocal2",
    "tgtDirectoryPath":  "C:\\server\\userdata\\workspace",
    "srcFilePattern":  ".*txt"

}
```

POST Response

If successful, Informatica Intelligent Cloud Services returns the run ID for the job. Use the run ID to monitor the job status.

The following example shows a successful response:

```
{
    "projectId": 0,
    "timeTaken": 0,
    "queuePriority": 0,
```

```
            "runPriority": 0,
            "runMode": "UNKNOWN",
            "submitSourceId": -1,
            "correlationId": "OWMxOTc2YjktNzI4YS00Mm",
            "runModeInteractive": false,
            "runModeBatch": false,
            "runModeDebug": false,
            "runModeUnknown": true,
            "formattedTimeTaken": "0.00",
            "id": 1000000000384,
            "runId": 385
    }
```

If unsuccessful, the response includes a reason for the failure.

receivefiles

Use the receivefiles resource to get files from a remote server.

The following connection types use the receivefiles resource to transfer files to the remote server:

- Advanced FTP V2
- Advanced FTPS V2
- Advanced SFTP V2

Before you construct a receivefiles request to receive files, obtain the identifier of the connection that provides access to the server. To get the connection ID, you can send a GET request using the connection resource. The connection resource can return information for each of your organization's connections.

POST Request

To receive files, include the connection ID in the following URI.

```
mftsaas/api/v1/receivefiles/<connection ID>
```

Include the following information in the request:

Field	Type	Required	Description
sourceConnectionType	String	Yes	Connection type. The supported source connection types are: - Advanced FTP V2 - Advanced FTPS V2 - Advanced SFTP V2
tgtDirectoryPath	String	Yes	Directory path to where files are transferred.
srcDirectoryPath	String	-	Directory path from where files are transferred. Default is '/'.
srcFilePattern	String	Yes	Source file name pattern. Specify a file name pattern to identify which files to send. You can use the regular expression type.
processFilesRecursively	String	-	Whether to process files from all sub-folders within the base directory. Default is false.

Field	Type	Required	Description
afterFilePickupAction	String	-	Determines what to do with source files after the files transfer. The following options are available: - Keep the files in the source directory. - Delete the files from the source directory. - Rename the files in the source directory. You must specify a file name suffix that adds to the file name when renaming the files. - Archive the files to a different location. You must specify an archive directory. Default is KEEP.
skipDuplicateFiles	String	-	Do not transfer duplicate files. If files with the same name and creation date were transferred, the task does not transfer them again, and the files are marked as duplicate in the job log. If this option is not selected the task transfers all files. Default is false.
whenFileExists	String	-	Determines what to do with a file if a flat file with the same name exists in the target directory. The following options are available: - rename - overwrite - skip - stop - error Default is rename.

For example, to transfer the files with ".txt" pattern, and rename the file if a flat file with same name exists in the target directory, you might use the following request:

```
POST <serverUrl>/mftsaas/api/v1/sendfiles/<connection ID>
Accept:application/json
IDS-SESSION-ID: <icSessionId or INFA-SESSION-ID>
{
    "sourceConnectionType": "Advanced SFTP V2",
    "tgtDirectoryPath": "C:\\docstoreLocal2",
    "srcDirectoryPath": "C:\\server\\userdata\\workspace",
    "srcFilePattern": ".*txt",
    "processFilesRecursively": false,
    "afterFilePickupAction": "KEEP",
    "skipDuplicateFiles": false,
    "whenFileExists": "rename",
}
```

POST Response

If successful, Informatica Intelligent Cloud Services returns the run ID for the job. Use the run ID to monitor the job status.

The following example shows a successful response:

```
{
    "projectId": 0,
    "timeTaken": 0,
    "queuePriority": 0,
    "runPriority": 0,
    "runMode": "UNKNOWN",
    "submitSourceId": -1,
    "correlationId": "OWMxOTc2YjktNzI4YS00Mm",
    "runModeInteractive": false,
    "runModeBatch": false,
    "runModeDebug": false,
    "runModeUnknown": true,
    "formattedTimeTaken": "0.00",
```

```
        "id": 1000000000384,
        "runId": 385
    }
```

If unsuccessful, the response includes a reason for the failure.

job

When you use the REST API to send or receive files, use the REST API version 1 job resource to get the status of the file transfer.

Do not use the platform REST API version 2 job resource to get the status of an file transfer job.

Get Request

When you send the request for status of an file transfer job, include the run ID returned in the sendfiles POST response. Use the following URI:

```
mftsaas/api/v1/job/<runID>/status
```

Get Response

If successful, Data Integration returns the job status.

If unsuccessful, the response includes a reason for the failure.

Get Response Example

If the request is successful, you might receive a response similar to the following example:

```
{
    "jobStatus": "SUCCESS"
}
```

filetransferTask

Use the filetransferTask resource to decrypt or decompress inbound files and to encrypt or compress outbound files.

You can transfer files in the following ways:

- Transfer files locally on a hosted server.
- Transfer files from or to a remote server.

Hosted file transfer task

You can manage files on a hosted server, and transfer files locally.

You can perform the following actions:

- Compress and transfer files to or within a folder in the home directory of the file server user.
- Decompress uploaded files and transfer them from the home directory of file server user to the target location.
- Encrypt and transfer files from source location to the home directory of the file server user.
- Decrypt and transfer files from the file server user's home directory to the target location.

Compress and transfer files

Compress and transfer inbound files to or within a folder specified in the home directory of the file server user.

POST Request

To compress and transfer files, include the connection ID in the following URI.

```
mftsaas/api/v1/filetransferTask/execute/1003
```

Include the following information in the request:

Field	Type	Required	Description
agentGroupId	String	Yes	The ID of the agent group.
portalUser	String	-	Whether the user is a portal user. Default is false.
fileServerUsername	String	Yes	The user name of the file server.
relativeTargetLocation	String	-	The relative target location within the file server user's home directory.
pattern	String	Yes	The file pattern to identify the files to collect for compression. The regular expression pattern is supported.
sourceLocation	String	Yes	The source directory that contains the files that you want to compress.
COMPRESSION_TYPE	String	Yes	The format of the files that you want to compress. Select one of the following compression methods: - Zip - Tar - Gzip The values are not case sensitive.

For example, to compress and transfer a file, you might use the following request:

```
POST <serverUrl>/mftsaas/api/v1/filetransferTask/execute/1003
Accept:application/json
IDS-SESSION-ID: <icSessionId or INFA-SESSION-ID>
{
    "agentGroupId": "01000125000000000002",
    "fileServerUsername": "arun",
    "portalUser":true,
    "relativeTargetLocation":"",
    "pattern":"arun.csv",
    "sourceLocation":"C:\\Informatica_Source",
    "taskVariables": {
    "COMPRESSION_TYPE": "zip"
    }
}
```

POST Response

If successful, Informatica Intelligent Cloud Services returns the run ID for the job. Use the run ID to monitor the job status.

The following example shows a successful response:

```
{
    "projectId": 0,
    "timeTaken": 0,
    "queuePriority": 0,
```

```
        "runPriority": 0,
        "runMode": "UNKNOWN",
        "submitSourceId": -1,
        "runModeInteractive": false,
        "runModeBatch": false,
        "runModeDebug": false,
        "runModeUnknown": true,
        "formattedTimeTaken": "0.00",
        "id": 1000000007154,
        "runId": 13
    }
```

If unsuccessful, the response includes a reason for the failure.

Decompress and transfer files

Decompress and transfer the uploaded files from the home directory of file server user to the target location.

POST Request

To decompress and transfer files, include the connection ID in the following URI.

```
mftsaas/api/v1/filetransferTask/execute/1004
```

Include the following information in the request:

Field	Type	Required	Description
agentGroupId	String	Yes	The ID of the agent group.
fileServerUsername	String	Yes	The user name of the file server.
pattern	String	Yes	The file pattern of the file to release to the specified target location after decompressing the file. The regular expression pattern is supported.
targetLocation	String	Yes	The target directory to which the file is moved after decompressing.
DECOMPRESSION_TYPE	String	Yes	The format of the files that you want to decompress. Select one of the following decompression methods: - Zip - Tar - Gzip The values are not case sensitive.
PATTERN_CASE_SENSITIVE	String	-	Whether the file pattern is case sensitive. The values are not case sensitive. Default is false.
PATTERN _TO_COLLECT	String	Yes	The pattern of the file that you want to collect to decompress from the file server user's home directory. Use a regular expression to match the file name pattern.

For example, to decompress and transfer a file, you might use the following request

```
POST <serverUrl>/mftsaas/api/v1/filetransferTask/execute/1004
Accept:application/json
IDS-SESSION-ID: <icSessionId or INFA-SESSION-ID>
{
    "agentGroupId": "01000125000000000002",
    "fileServerUsername": "arun",
    "pattern":".*csv",
    "targetLocation":"C:\\Informatica_Target",
```

```
            "taskVariables": {
            "PATTERN_CASE_SENSITIVE": "false",
            "DECOMPRESSION_TYPE": "unzip",
            "PATTERN_TO_COLLECT": ".*zip"
            }
    }
```

POST Response

If successful, Informatica Intelligent Cloud Services returns the run ID for the job. Use the run ID to monitor the job status.

The following example shows a successful response:

```
{
    "projectId": 0,
                "timeTaken": 0,
                "queuePriority": 0,
                "runPriority": 0,
                "runMode": "UNKNOWN",
                "submitSourceId": -1,
                "runModeInteractive": false,
                "runModeBatch": false,
                "runModeDebug": false,
                "runModeUnknown": true,
                "formattedTimeTaken": "0.00",
                "id": 1000000007161,
                "runId": 20
    }
```

If unsuccessful, the response includes a reason for the failure.

Encrypt and transfer files

Encrypt and transfer files from the source location to the home directory of the file server user or the directory specified in the REST API param within the file server user's home directory. You can encrypt single or multiple files.

POST Request

To encrypt and transfer files, include the connection ID in the following URI.

```
mftsaas/api/v1/filetransferTask/execute/1001
```

Include the following information in the request:

Field	Type	Required	Description
agentGroupId	String	Yes	The ID of the agent group.
portalUser	String	-	Whether the user is a portal user. Default is false.
fileServerUsername	String	Yes	The user name of the file server.
relativeTargetLocation	String	-	The relative target location within the file server user's home directory.
pattern	String	Yes	The file pattern to identify the files to collect for encryption. The regular expression pattern is supported.
sourceLocation	String	Yes	The source directory that contains the files you want to encrypt.

Field	Type	Required	Description
SIGN	String	-	Whether the file is signed by PGP. The values are not case sensitive. Default is false.
PUBLIC_KEY_ID	String	Yes	The ID of the key that is used to encrypt the file.
SECRET_KEY_ID	String	Yes	The ID of the secret key that is used to sign the file, if the value of the SIGN variable is true.
SECRET_KEY_PASSPHRASE	String	Yes	The passphrase used to access the secret key if the value of the SIGN variable is true.

For example, to encrypt and transfer a file, you might use the following request:

```
POST <serverUrl>/mftsaas/api/v1/filetransferTask/execute/1001
Accept:application/json
IDS-SESSION-ID: <icSessionId or INFA-SESSION-ID>
{
    "agentGroupId": "01000125000000000002",
    "fileServerUsername": "arun",
    "portalUser":true,
    "pattern":"arun.csv",
    "relativeTargetLocation":"",
    "sourceLocation":"C:\\Informatica_Source",
    "taskVariables": {
        "SIGN":"false",
        "PUBLIC_KEY_ID":"0x51986F687ADACBE1",
        "SECRET_KEY_ID":"0x51986F687ADACBE1",
        "SECRET_KEY_PASSPHRASE":"TESTER"
    }
}
```

POST Response

If successful, Informatica Intelligent Cloud Services returns the run ID for the job. Use the run ID to monitor the job status.

The following example shows a successful response:

```
{
    "projectId": 0,
            "timeTaken": 0,
            "queuePriority": 0,
            "runPriority": 0,
            "runMode": "UNKNOWN",
            "submitSourceId": -1,
            "runModeInteractive": false,
            "runModeBatch": false,
            "runModeDebug": false,
            "runModeUnknown": true,
            "formattedTimeTaken": "0.00",
            "id": 1000000007155,
            "runId": 14
            }
```

If unsuccessful, the response includes a reason for the failure.

Decrypt and transfer files

Decrypt and transfer files from the file server user's home directory to the target location.

POST Request

To decrypt and transfer files, include the connection ID in the following URI.

```
mftsaas/api/v1/filetransferTask/execute/1002
```

Include the following information in the request:

Field	Type	Required	Description
agentGroupId	String	Yes	The ID of the agent group.
fileServerUsername	String	Yes	The user name of the file server.
pattern	String	Yes	The file pattern of the file to release to the specified target location after decrypting the file. The regular expression pattern is supported.
targetLocation	String	Yes	The target directory to which the file is moved after decryption.
PATTERN_CASE_SENSITIVE	String	Yes	Whether the file pattern is case sensitive. The values are not case sensitive. Default is false.
PGP_PASSPHRASE	String	Yes	The PGP passphrase.
PATTERN _TO_COLLECT	String	Yes	The file name pattern of the files that PGP has to collect and decrypt. Use a regular expression to match the file name pattern.

For example, to decrypt and transfer a file, you might use the following request:

```
POST <serverUrl>/mftsaas/api/v1/filetransferTask/execute/1002
Accept:application/json
IDS-SESSION-ID: <icSessionId or INFA-SESSION-ID>
{
        "agentGroupId": "01000125000000000002",
        "fileServerUsername": "arun",
        "pattern":".*csv",
        "targetLocation":"C:\\Informatica_Target",
        "taskVariables": {
            "PATTERN_CASE_SENSITIVE": "false",
            "PGP_PASSPHRASE": "TESTER",
            "PATTERN_TO_COLLECT": ".*pgp"
    }
}
```

POST Response

If successful, Informatica Intelligent Cloud Services returns the run ID for the job. Use the run ID to monitor the job status.

The following example shows a successful response:

```
{
    "projectId": 0,
                "timeTaken": 0,
                "queuePriority": 0,
                "runPriority": 0,
                "runMode": "UNKNOWN",
                "submitSourceId": -1,
```

```
                                "runModeInteractive": false,
                                "runModeBatch": false,
                                "runModeDebug": false,
                                "runModeUnknown": true,
                                "formattedTimeTaken": "0.00",
                                "id": 1000000007160,
                                "runId": 19
            }
```

If unsuccessful, the response includes a reason for the failure.

Remote file transfer task

You can manage the files on a remote server using Advanced FTP V2, Advanced FTPS V2, and Advanced SFTP V2 connectors, and transfer files from or to a remote server.

You can perform the following actions:

- Compress and transfer files to or within a folder in the home directory of the file server user.
- Decompress uploaded files and transfer them from the home directory of file server user to the target location.
- Encrypt and transfer files from source location to the home directory of the file server user.
- Decrypt and transfer files from the file server user's home directory to the target location.

Compress and transfer files

Compress and transfer inbound files to or within a folder specified in the home directory of the remote server.

POST Request

To compress and transfer files, include the connection ID in the following URI.

```
mftsaas/api/v1/filetransferTask/execute/1003
```

Include the following information in the request:

Field	Type	Required	Description
agentGroupId	String	Yes	The ID of the agent group.
portalUser	String	-	Whether the user is a portal user. Default is false.
connectionId	String	Yes	The connection ID of the Advanced FTP, Advanced FTPS, or Advanced SFTP V2 connector.
relativeTargetLocation	String	-	The relative target location within the remote server home directory.
pattern	String	Yes	The file pattern to identify the files to collect for compression. The regular expression pattern is supported.

Field	Type	Required	Description
sourceLocation	String	Yes	The source directory that contains the files that you want to compress.
COMPRESSION_TYPE	String	Yes	The format of the files that you want to compress. Select one of the following compression methods: - Zip - Tar - Gzip The values are not case sensitive.

For example, to compress and transfer a file, you might use the following request:

```
POST <serverUrl>/mftsaas/api/v1/filetransferTask/execute/1003
Accept:application/json
IDS-SESSION-ID: <icSessionId or INFA-SESSION-ID>
{
   "agentGroupId": "01000125000000000002",
   "portalUser": "false",
   "connectionId": "0100010B000000000002",
   "pattern":"arun_zip.txt",
   "relativeTargetLocation":"/",
   "sourceLocation":"C:\\FIS_Home\\DOCSTORE",
   "taskVariables": {
     "COMPRESSION_TYPE": "gzip"
     }
}
```

POST Response

If successful, Informatica Intelligent Cloud Services returns the run ID for the job. Use the run ID to monitor the job status.

The following example shows a successful response:

```
{
    "projectId": 0,
    "timeTaken": 0,
    "queuePriority": 0,
    "runPriority": 0,
    "runMode": "UNKNOWN",
    "submitSourceId": -1,
    "runModeInteractive": false,
    "runModeBatch": false,
    "runModeDebug": false,
    "runModeUnknown": true,
    "formattedTimeTaken": "0.00",
    "id": 1000000007170,
    "runId": 29
}
```

If unsuccessful, the response includes a reason for the failure.

Decompress and transfer files

Decompress and transfer the uploaded files from the home directory of the remote server to the target location.

POST Request

To decompress and transfer files, include the connection ID in the following URI.

```
mftsaas/api/v1/filetransferTask/execute/1004
```

Include the following information in the request:

Field	Type	Required	Description
agentGroupId	String	Yes	The ID of the agent group.
portalUser	String	-	Whether the user is a portal user. Default is false.
relativeSourceLocation	String	-	The relative source location within the remote server home directory.
pattern	String	Yes	The file pattern of the file to release to the specified target location after decompressing the file. The regular expression pattern is supported.
targetLocation	String	Yes	The target directory to which the file is moved after decompressing.
relativeTargetLocation	String	-	The relative target location within the remote server home directory.
connectionId	String	Yes	The connection ID of the Advanced FTP, Advanced FTPS, or Advanced SFTP V2 connector.
afterFilePickupAction	String	-	Determines what to do with the source files after the files are transferred. Select one of the following filter options: - KEEP. Keep the files in the source directory. - DELETE. Delete the files from the source directory. - RENAME. Rename the files in the source directory. - ARCHIVE. Archive the files to a different location. You must specify an archive directory Default is KEEP.
renameSuffix	String	Yes	If afterFilePickupAction is selected as RENAME, the file name suffix to append to the files in the source directory. You can use the following suffix types: - $date - $time - $runId - $timestamp
archiveDirectoryPath	String	Yes	If afterFilePickupAction is selected as ARCHIVE, the archive directory in which to archive the files.
skipDuplicateFiles	String	-	Indicates whether to skip the source files which are already present in the docstore location. Default is false.
processFilesRecursively	String	-	Indicates whether to process files from all sub-folders within the base directory. Default is false.

Field	Type	Required	Description
DECOMPRESSION_TYPE	String	Yes	The format of the files that you want to decompress. Select one of the following decompression methods: - Zip - Tar - Gzip The values are not case sensitive.
PATTERN _TO_COLLECT	String	Yes	The pattern of the file that you want to collect to decompress from the file server user's home directory. Use a regular expression to match the file name pattern.

For example, to decompress and transfer a file, you might use the following request:

```
POST <serverUrl>/mftsaas/api/v1/filetransferTask/execute/1004
Accept:application/json
IDS-SESSION-ID: <icSessionId or INFA-SESSION-ID>
{
    "agentGroupId": "01000125000000000002",
    "portalUser": "false",
    "pattern": "arun_zip.txt",
    "relativeSourceLocation": "/",
    "targetLocation": "C:\\Informatica_Target",
    "relativeTargetLocation": "",
    "connectionId": "0100010B000000000002",
    "afterFilePickupAction": "RENAME",
    "renameSuffix":"_RENAME_",
    "archiveDirectoryPath" :"",
    "skipDuplicateFiles": true,
    "processFilesRecursively": false,
  "taskVariables": {
      "DECOMPRESSION_TYPE": "gunzip",
       "PATTERN_TO_COLLECT": ".*gz"
    }
}
```

POST Response

If successful, Informatica Intelligent Cloud Services returns the run ID for the job. Use the run ID to monitor the job status.

The following example shows a successful response:

```
{
      "projectId": 0,
    "timeTaken": 0,
    "queuePriority": 0,
    "runPriority": 0,
    "runMode": "UNKNOWN",
    "submitSourceId": -1,
    "runModeInteractive": false,
    "runModeBatch": false,
    "runModeDebug": false,
    "runModeUnknown": true,
    "formattedTimeTaken": "0.00",
    "id": 1000000007171,
    "runId": 30
}
```

If unsuccessful, the response includes a reason for the failure.

Encrypt and transfer files

Encrypt and transfer files from the source location to a remote server using connectors or to the directory specified in the REST API param within the remote server's home directory. You can encrypt single or multiple files.

POST Request

To encrypt and transfer files, include the connection ID in the following URI.

```
mftsaas/api/v1/filetransferTask/execute/1001
```

Include the following information in the request:

Field	Type	Required	Description
agentGroupId	String	Yes	The ID of the agent group.
connectionId	String	Yes	The connection ID of the Advanced FTP, Advanced FTPS, or Advanced SFTP V2 connector.
portalUser	String	-	Whether the user is a portal user. Default is false.
relativeTargetLocation	String	-	The relative target location within the remote server home directory.
pattern	String	Yes	The file pattern to identify the files to collect for encryption. The regular expression pattern is supported.
sourceLocation	String	Yes	The source directory that contains the files you want to encrypt.
SIGN	String	-	Whether the file is signed by PGP. The values are not case sensitive. Default is false.
PUBLIC_KEY_ID	String	Yes	The ID of the key that is used to encrypt the file.
SECRET_KEY_ID	String	Yes	The ID of the secret key that is used to sign the file, if the value of the SIGN variable is true.
SECRET_KEY_PASSPHRASE	String	Yes	The passphrase used to access the secret key if the value of the SIGN variable is true.

For example, to encrypt and transfer a file, you might use the following request:

```
POST <serverUrl>/mftsaas/api/v1/filetransferTask/execute/1001
Accept:application/json
IDS-SESSION-ID: <icSessionId or INFA-SESSION-ID>
{
    "agentGroupId": "01000125000000000002",
    "portalUser": "false",
     "connectionId": "0100010B000000000002",
    "pattern":"arun.txt",
    "relativeTargetLocation":"/",
    "sourceLocation":"C:\\FIS_Home\\DOCSTORE",
    "taskVariables": {
        "SIGN":"true",
        "PUBLIC_KEY_ID":"0x51986F687ADACBE1",
        "SECRET_KEY_ID":"0x51986F687ADACBE1",
        "SECRET_KEY_PASSPHRASE":"TESTER"
    }
}
```

POST Response

If successful, Informatica Intelligent Cloud Services returns the run ID for the job. Use the run ID to monitor the job status.

The following example shows a successful response:

```
{
        "projectId": 0,
    "timeTaken": 0,
    "queuePriority": 0,
    "runPriority": 0,
    "runMode": "UNKNOWN",
    "submitSourceId": -1,
    "runModeInteractive": false,
    "runModeBatch": false,
    "runModeDebug": false,
    "runModeUnknown": true,
    "formattedTimeTaken": "0.00",
    "id": 1000000007165,
    "runId": 24
}
```

If unsuccessful, the response includes a reason for the failure.

Decrypt and transfer files

Decrypt and transfer files from the remote server home directory to the target location.

POST Request

To decrypt and transfer files, include the connection ID in the following URI.

```
mftsaas/api/v1/filetransferTask/execute/1002
```

Include the following information in the request:

Field	Type	Required	Description
agentGroupId	String	Yes	The ID of the agent group.
portalUser	String	-	Whether the user is a portal user. Default is false.
relativeSourceLocation	String	-	The relative source location within the remote server home directory.
pattern	String	Yes	The file pattern of the file to release to the specified target location after decrypting the file. The regular expression pattern is supported.
targetLocation	String	Yes	The target directory to which the file is moved after decrypting.
relativeTargetLocation	String	-	The relative target location within the remote server home directory.
connectionId	String	Yes	The connection ID of the Advanced FTP, Advanced FTPS, or Advanced SFTP V2 connector.

Field	Type	Required	Description
afterFilePickupAction	String	-	Determines what to do with the source files after the files are transferred. Select one of the following filter options: - KEEP. Keep the files in the source directory. - DELETE. Delete the files from the source directory. - RENAME. Rename the files in the source directory. - ARCHIVE. Archive the files to a different location. You must specify an archive directory Default is KEEP.
renameSuffix	String	Yes	If afterFilePickupAction is selected as RENAME, the file name suffix to append to the files in the source directory. You can use the following suffix types: - $date - $time - $runId - $timestamp
archiveDirectoryPath	String	Yes	If afterFilePickupAction is selected as ARCHIVE, the archive directory in which to archive the files.
skipDuplicateFiles	String	-	Indicates whether to skip the source files which are already present in the docstore location. Default is false.
processFilesRecursively	String	-	Indicates whether to process files from all sub-folders within the base directory. Default is false.
PATTERN_CASE_SENSITIVE	String	Yes	Whether the file pattern is case sensitive. The values are not case sensitive. Default is false.
PGP_PASSPHRASE	String	Yes	The PGP passphrase.
PATTERN _TO_COLLECT	String	Yes	The pattern of the file that you want to collect to decompress from the file server user's home directory. Use a regular expression to match the file name pattern.

For example, to decrypt and transfer a file, you might use the following request:

```
POST <serverUrl>/mftsaas/api/v1/filetransferTask/execute/1002
Accept:application/json
IDS-SESSION-ID: <icSessionId or INFA-SESSION-ID>
{
    "agentGroupId": "01000125000000000002",
    "portalUser": "false",
    "pattern": "arun.txt",
    "relativeSourceLocation": "/",
    "targetLocation": "C:\\Informatica_Target",
    "relativeTargetLocation": "",
    "connectionId": "0100010B000000000002",
    "afterFilePickupAction": "ARCHIVE",
    "renameSuffix":"_RENAME_",
    "archiveDirectoryPath" :"/ARCH",
    "skipDuplicateFiles": false,
    "processFilesRecursively": false,
    "taskVariables": {
        "PATTERN_CASE_SENSITIVE": "false",
        "PGP_PASSPHRASE": "TESTER",
```

```
            "PATTERN_TO_COLLECT": "arun.txt.pgp"
        }
    }
```

POST Response

If successful, Informatica Intelligent Cloud Services returns the run ID for the job. Use the run ID to monitor the job status.

The following example shows a successful response:

```
{
    "projectId": 0,
    "timeTaken": 0,
    "queuePriority": 0,
    "runPriority": 0,
    "runMode": "UNKNOWN",
    "submitSourceId": -1,
    "runModeInteractive": false,
    "runModeBatch": false,
    "runModeDebug": false,
    "runModeUnknown": true,
    "formattedTimeTaken": "0.00",
    "id": 1000000007169,
    "runId": 28
}
```

If unsuccessful, the response includes a reason for the failure.

HTTPS file transfer

You can send files to a remote HTTPS server or receive files from a remote HTTPS server, and get the job status through the REST API.

Consider the following when you use the resources for HTTPS file transfer:

- You must have the HTTPS license to exchange files through HTTPS servers.
- You should log in to the HTTPS server to perform the API operations.

You can use the following resources for HTTPS file transfer:

- Authentication. Use to authenticate a user.
- Standard operations. Use during a file transfer action.
- File transfer. Use to transfer file to or from a HTTPS server
- Server responses. Details on the server responses.
- Status codes. Details on the HTTPS file transfer status codes.

Authentication

Use the following resources to log in to the HTTP server and log out.

login

You should log in to the HTTPS server to perform any HTTPS API operations. Use this command to lg in to the HTTPS server. To log in to the Informatica Managed File Transfer HTTPS Server send a POST request using the login resource. You must send a login request to start a user session if you don't use client certificate authentication.

Include the following parameters in the request:

- Username. The name of the user on the server.

- Password. The password required to log in.

For example: `https://10.60.40.11:15400/fileservers/login?`
`username=https_automation&password=T@1234`

logout

Use the logout resource to log out and end the user session on the Informatica Managed File Transfer HTTPS Server. You can make a GET or POST request.

For example, `https://10.60.40.11:15400/fileservers/logout`

Standard operations

You can use the following standard operational commands:

PWD

Use the PWD (Print Working Directory) command to retrieve the current working directory on the server. The response includes the absolute path to the current working directory as part of the X-GDX-Reply header message. The path is enclosed in double quotes.

For example, `https://localhost:15400/fileservers/pwd`

delete

Use this command to remove files from the server. Include the relative or absolute file path to delete in the file parameter.

For example, `https://10.60.40.11:15400/fileservers/delete?file=/abc/1.txt`

rename

Use this command to rename files on the server. If the current working directory contains the files that you want to rename, then the from and to parameters might contain only the file names. You can also use the rename command to move files on the server. To move files, include the full paths in the from and to parameters.

For example, the following command changes the name of the newInput.txt file to Input.txt:

`https://10.60.40.11:15400/fileservers/rename?from=/newInput.txt&;to=/Input.txt`

For example, the following command moves the newInput.txt file from the current working directory to the parent directory:

`https://10.60.40.11:15400/fileservers/rename?from=/newInput.txt&to=/aa/newInput.txt`

Include the following parameters in the request:

Request Type	Parameters
GET or POST	- from: The relative or absolute path of the file or directory to rename. - to: The relative or absolute path of the new name.

list

Use this command to list the contents of a directory on the server. Include the target directory as a parameter to this command. If you do not include the directory, the command lists the contents of the current working directory.

For example, `https://10.60.40.11:15400/fileservers/list?dir=/`

The response body includes the contents of the directory in content type text/plain. The following example shows the format of the directory listing:

```
2009-12-03 14:02:19 D 0 backup
```

The response includes the following information delimited by a tab (\t) character:

- The last modified date of the file or directory. The timestamp is in ISO format yyyy-MM-dd HH:mm:ss. The hour(hh) is displayed as a 24-hour clock.
- Whether the content type is a file, a directory, or unknown.
- The size of the file in bytes.
- The name of the file or directory.

checksum

Use this command to calculate the hash of a remote file. The reply is returned on the first line of the response body. You can compare the response with the hash value of the downloaded local file to verify data integrity.

For example, `https://10.60.40.11:15400/fileservers/hash?file=/input.txt`

The supported hash algorithms are SHA1, MD5, and CRC32. Include the following parameters in the request:

Request Type	Parameters
GET or POST	- file: Required. The path relative to the current working directory, or an absolute path to the file. - algorithm: The hash algorithm to use when calculating a checksum. Valid values are SHA1, MD5, or CRC32. Default is SHA1. - length: The starting position within the file. This value is used for calculating partial file checksums. The default value is 0, which performs the checksum on the entire file.

CD (Change Directory)

Use this command to change the current working directory. The absolute path to the new working directory returns as part of the X-GDX-Reply header message. The path is enclosed in double quotes.

For example, `https://10.60.40.11:15400/fileservers/cd?dir=/`

CDUP (Change Directory Up)

Use this command to change the current working directory to the parent directory. The absolute path to the new working directory returns as part of the X-GDX-Reply header message. The path is enclosed in double quotes.

For example, `https://10.60.40.11:15400/fileservers/cdup`

MKDIR (Make Directory)

Use this command to create a new directory on the server. The absolute path to the newly created directory returns as part of the X-GDX-Reply header message. The path is enclosed in double quotes.

For example, `https://10.60.40.11:15400/fileservers/mkdir?dir=/a/b/c/mkdri1`

file information

Use this command to retrieve information about a specific file or directory. The response includes the information in the response body with content type text/plain. The format of the file information is

identical to the listing returned from the List command. If no information is returned in the response body, then the file or directory does not exist.

For example, `https://10.60.40.11:15400/fileservers/fileInfo?file=/TEST.txt`

File transfer

Use the following commands to transfer an HTTPS file to or from a server:

upload

Use this command to transfer a file to the server. The request must be a multipart POST request and only one file is uploaded per request. A file is a required part of the multipart request, but any parameter name given to the file part is ignored.

For example, `https://10.60.40.11:15400/fileservers/upload`

The response includes the following parameters:

Request Type	Parameters
POST/Multipart	- to: The relative or absolute path of the destination file. - file: The file being uploaded as part of the multipart request.

upload2

Use this command to transfer a file to the server. The request must be a multipart POST request and only one file is uploaded per request. A file is a required part of the multipart request, but any parameter name given to the file part is ignored.

For example, `https://10.60.40.11:15400/fileservers/upload2`

The response includes the following parameters:

Request Type	Parameters
POST/ Multipart	- to: The relative or absolute path of the destination file. - append: Optional. If the file exists on the target directory, set this parameter to `true` to append the new file to the existing one. - transferMode: Use B for binary transfers or A for ascii transfers. Default is B. - file: The file being uploaded as part of the multipart request.

upload3

Use this command to transfer a file to the server. The request must be a multipart POST request and only one file is uploaded per request. A file is a required part of the multipart request, but any parameter name given to the file part is ignored.

For example, `https://10.60.40.11:15400/fileservers/upload3`

You must provide the fileserver username and password for basic authorization.

The response includes the following parameters:

Request Type	Parameters
POST/ Multipart	- to: The relative or absolute path of the destination file. - append: Optional. If the file exists on the target directory, set this parameter to `true` to append the new file to the existing one. - transferMode: Use B for binary transfers or A for ascii transfers. Default is B. - file: The file being uploaded as part of the multipart request.

uploadRawData

Use this command to upload data directly to the server where the data is the content of the request body. The request must be a POST request. The name of the file is automatically derived and returned as part of the X-GDX-Reply header message. This is a special command where the request body must contain the file data being uploaded.

For example, `https://10.60.40.11:15400/fileservers/uploadRawData`

uploadRawData2

Use this command to upload data directly to the server where the data is the content of the request body. The request must be a POST request. The name of the file is automatically derived and returned as part of the X-GDX-Reply header message. This is a special command where the request body must contain the file data being uploaded.

For example, `https://10.60.40.11:15400/fileservers/uploadRawData2`

You must provide the fileserver username and password for basic authorization.

download

Use this command to download a file from the server. The file is returned as the response body. The content type is always an application or force download, along with the content disposition field containing the name of the file. The content-length header is also included in the response indicating the size of the file.

For example, `https://10.60.40.11:15400/fileservers/download?file=/ test&downloadReleased=true&transferMode=b`

The response includes the following parameters:

Request Type	Parameters
GET or POST	- file: Required. The file to download. This can be a path relative to the current working directory, or an absolute path to the file. - offset: For downloading partial files. Enter the starting position from where the file must begin to download. - transferMode: Use B for binary transfers or A for ascii transfers. Default is B.

Server responses

For every request, the server responds with success or error codes and messages specific to the HTTPS service in the X-CDX-Reply header. The format of the header message is a status code followed by a single white space, followed by the message details.

For example, 200 Welcome, testuser!

You can use header codes and messages to determine the success or failure of an operation.

Status codes

A response might include any of the following HTTPS file transfer status codes:

200-299

Informational or success status codes: Successful operation performed against the server .

500-509

Internal server error: The server experienced a critical error. Contact the server's administrator immediately.

510-519

Bad or Invalid request: The server could not process the request due to invalid or incomplete information. See the X-GDX-Reply header message for more details.

530-539

Login or account related errors: Indicates that an error occurred with the account or login, such as invalid login or account disabled. See the X-GDX-Reply header message for more details.

550-559

Permission errors: The user does not have permission or authority to perform the requested action. See the X-GDX-Reply header message for more details.

560-569

Errors related to files or directories on the system: An error occurred while accessing a file or directory on the server, such as a file or directory does not exist.

580-589

File I/O Errors: An internal server error occurred while trying to access a file or directory.

590

Unknown error: An unexpected error occurred while trying to process the command. See the X-GDX-Reply header message for more details.

fwConfig

Use the fwConfig resource to configure column widths for flat file source, lookup, and target objects.

GET request

To request all of the fixed-width formats, use the following URI:

```
/api/v2/fwConfig
```

To request the details of a particular fixed-width format, you can include the fixed-width format ID, the fixed-width federated ID, or fixed-width format name in the URI. If you include the federated ID, the response includes the project and folder location of the fixed-width format. Use one of the following URIs:

```
/api/v2/fwConfig/<id>
/api/v2/fwConfig/frs/<id>
/api/v2/fwConfig/name/<name>
```

If you use the fixed-width format name in the URI and the fixed-width format name includes a space, replace the space with `%20`. For example:

```
/api/v2/fwConfig/name/my%20fixedwidth%20format
```

GET response

The fwConfig object returns the following attributes:

Field	Type	Description
id	String	Fixed-width format ID.
name	String	Fixed-width format name.
description	String	Description of the fixed-width format.
createTime	Date/time	Time that the fixed-width format was created.
updateTime	Date/time	Last time that the fixed-width format was updated.
createdBy	String	User who created the fixed-width format.
updatedBy	String	User who last updated the fixed-width format.
lineSequential	Boolean	Whether each row ends with a newline character. - True. Line sequential is enabled. - False. Line sequential is not enabled.
padBytes	Int	Number of bytes between the last column of one row and the first column of the next.
skipRows	Int	Number of rows to skip. You can skip blank or header rows.
nullChar	String	The character to represent a null value.
dateFormat	String	Default date format to use when a date format is not specified in the flat file connection.
nullCharType	String	Determines if the null character is single-byte or multibyte.
repeatNullChar	Boolean	Determines how to treat null characters in a single field. - True. Read repeat null characters as a single null value. - False. does not read repeat null characters as a single null value.
stripTrailingBlank	Boolean	Determines how to treat trailing blanks in string values. - True. Removes trailing blanks from string values. - False. Does not remove trailing blanks in string values.
location	String	The project and folder that the fixedd-width file resides in.
columns	String	Includes the following attributes for each column: - name. Name of the column. - nativeType. Native data type. - precision. Length of the field in bytes. - scale. Number of digits after the decimal point for numeric values.

GET example

The following example shows a request to get details for a fixed-width format using the fixed-width format ID:

```
GET <serverUrl>/api/v2/fwConfig/00001R29000000000002
Accept:application/json
icSessionId: <icSessionId>
```

The following text is a sample response:

```
{
  "@type": "fwConfig",
  "id": "00001R29000000000002",
  "orgId": "00001R",
  "name": "item",
  "description": "",
  "createTime": "2016-10-06T17:08:09.000Z",
  "updateTime": "2016-10-06T17:08:09.000Z",
  "createdBy": "org1@infa.com",
  "updatedBy": "org1@infa.com",
  "lineSequential": true,
  "padBytes": 0,
  "skipRows": 0,
  "nullChar": "*",
  "nullCharType": "ASCII",
  "repeatNullChar": false,
  "stripTrailingBlank": false,
  "location": Default\\MyProject
  "dateFormat": "",
  "columns": [
    {
      "@type": "fwColumn",
      "name": "COLUMN_0",
      "nativeType": "string",
      "precision": 1,
      "physicalLength": 0,
      "scale": 0
    },
    {
      "@type": "fwColumn",
      "name": "COLUMN_1",
      "nativeType": "string",
      "precision": 9,
      "physicalLength": 0,
      "scale": 0
    },
    {
      "@type": "fwColumn",
      "name": "COLUMN_2",
      "nativeType": "string",
      "precision": 10,
      "physicalLength": 0,
      "scale": 0
    }
  ]
}
```

POST request

To create a fixed-width format, use the following URI:

```
/api/v2/fwConfig
```

If you want to specify a location for the fixed-width format, include the container ID in the request. If the container ID isn't included in the request, the fixed-width format is created in the Default folder. You can find the container ID for a project or folder in the Data Integration user interface. On the **Explore** page, select the folder. In the URL, the last string of characters is the container ID.

For example, in the following URL, the container ID is dH2DuGJYda7ijgW4Sm32sR

```
https://na1.dm-us.informaticacloud.com/diUI/products/integrationDesign/main/Explore/
dH2DuGJYda7ijgW4Sm32sR
```

To update a fixed-width format, include the fixed-width format ID in the following URI:

```
/api/v2/fwConfig/<id>
```

You can submit a partial update using partial mode. If you want to update a field in the fwColumn object using partial mode, you must include the name. To submit a request using partial mode, use a JSON request and include the following line in the header:

```
Update-Mode=PARTIAL
```

You can use the following attributes in a fwConfig POST request:

Field	Type	Required	Description
id	String	Yes	Fixed-width format ID.
name	String	Yes	Fixed-width format name.
description	String		Description of the fixed-width format.
containerId	String		ID of the project or folder to contain the linear taskflow. If not included in request, the linear taskflow is created in the Default folder.
lineSequential	Boolean	Yes	Whether each row ends with a newline character. - True. Line sequential is enabled. - False. Line sequential is not enabled.
padBytes	Int	Yes	Number of bytes between the last column of one row and the first column of the next.
skipRows	Int	Yes	Number of rows to skip. You can skip blank or header rows.
nullChar	String	Yes	The character to represent a null value.
dateFormat	String	Yes	Default date format to use when a date format is not specified in the flat file connection.
nullCharType	String	Yes	Determines if the null character is single-byte or multibyte.
repeatNullChar	Boolean	Yes	Determines how to treat null characters in a single field. - True. Read repeat null characters as a single null value. - False. does not read repeat null characters as a single null value.
stripTrailingBlank	Boolean	Yes	Determines how to treat trailing blanks in string values. - True. Removes trailing blanks from string values. - False. Does not remove trailing blanks in string values.
columns	String	Yes	Includes the following attributes for each column: - name. Name of the column. - nativeType. Native data type. - precision. Length of the field in bytes. - scale. Number of digits after the decimal point for numeric values.

POST response

If successful, returns the fwConfig object that you created or updated. Returns the error object if errors occur.

POST example

```
POST <serverURL>/api/v2/fwConfig/00000103000000000004
Content-Type: application/json
Accept: application/json
{
    "@type": "fwConfig",
    "name": "FW_FILE_CONFIG_1",
    "description": "Test description",
    "lineSequential": false,
    "padBytes": 1,
    "skipRows": 2,
    "nullChar": "*",
    "nullCharType": "ASCII",
    "repeatNullChar": false,
    "stripTrailingBlank": false,
    "columns": [
        {
            "@type": "fwColumn",
            "name": "ASCII",
            "nativeType": "string",
            "precision": 10
        }
    ]
}
```

DELETE request

To delete a fixed-width format, use the fixed-width format ID in the following URI:

```
/api/v2/fwConfig/<id>
```

DELETE response

Returns the 200 response code if the request is successful.

Returns the error object if errors occur.

Data Integration job log files

You can request log information for completed Data Integration jobs from Operational Insights.

To get log entries from Operational Insights, use the following resources:

- Login. Use to log in to Informatica Intelligent Cloud Services and get the session ID to use in job log REST API calls.

- jobLogEntries. Use to request log information for completed Data Integration jobs.

When you use these resources, note the following rules:

- Use JSON format.

- Use the following URL:

```
<serverUrl>/cdiinsights-service/api/v1/analytical/Orgs(<orgID>)/<API name>
```

- Use the following request header format:

```
<METHOD> <serverUrl>/<URI> HTTP/<HTTP version>
Content-Type: application/json
```

```
Accept: application/json
IDS-SESSION-ID: <SessionId>
```

The server URL includes the name and region of the POD that your organization uses and the Informatica Intelligent Cloud Services domain, informaticacloud.com. If you don't know the name and region of your organization's POD, you can find it by logging in to Informatica Intelligent Cloud Services through the user interface. The POD information is located in the browser's address bar.

In the following example, https://usw3.dm-us.informaticacloud.com is the server URL:

```
https://usw3.dm-us.informaticacloud.com/diUI/products/integrationDesign/main/home
```

Use the server URL as the base URL in the header of REST API calls.

Note: If you use a tool such as Postman that automatically includes the HTTP version, don't enter the HTTP version in the URL. If the HTTP version appears twice in the URL, the request fails.

Login

Use this resource to log into Informatica Intelligent Cloud Services when you use resources that require the IDS-SESSION-ID in the call header. The IDS-SESSION-ID is included in a successful login response.

POST request

Use the following URL:

```
<login URL>/identity-service/api/v1/Login
```

The login URL includes the region where your organization is located and the Informatica Intelligent Cloud Services domain, informaticacloud.com. You can find your organization's login region by opening the Informatica Intelligent Cloud Services log in page. The regional login URL is located in the browser's address bar before you log in to Informatica Intelligent Cloud Services.

In the following example, https://dm-us.informaticacloud.com, is the region URL:

```
https://dm-us.informaticacloud.com/identity-service/home
```

The following table describes the fields to include in the request:

Field	Type	Required	Description
username	String	Yes	Informatica Intelligent Cloud Services user name. Maximum length is 255 characters.
password	String	Yes	Informatica Intelligent Cloud Services password. Maximum length is 255 characters.

POST response

Returns the user object if the request is successful. Returns the error object if errors occur.

Use the session ID returned in the response for subsequent requests.

The user object includes the following attributes:

Field	Type	Description
sessionId	String	REST API session ID for the current session. Use in most REST API request headers.
sessionExpireTime	String	Time the session expires.

Field	Type	Description
id	String	User ID.
name	String	Informatica Intelligent Cloud Services user name.
currentOrgId	String	Current organization ID.
currentOrgName	String	Name of the current organization.
parentOrgId	String	ID of the parent organization.
orgId	String	ID of the organization the user belongs to.
orgName	String	Name of the organization the user belongs to.
groups	String	User group.
effectiveRoles	String	Roles assigned to the user.
effectivePrivileges	String	Privileges assigned to the user.
status	String	Status of the user.
timeZoneId	String	Time zone of the user. Time zone honors Daylight Saving Time. For more information, see "Time zone codes" on page 543.
authenticator	String	User authentication method.

jobLogEntries

Use the jobLogEntries resource to request log information for completed Data Integration jobs from Operational Insights. When you use this resource, include the organization ID in the URL.

GET Request

You can request all the log information or filter the log response. You can request data for the last 33 days. The response returns a maximum of 500 records for any applied filter.

To request information from the log, use the following URI:

```
/cdiinsights-service/api/v1/analytical/Orgs(<orgID>)/JobLogEntries
```

To filter the log response, use the following URI:

```
/cdiinsights-service/api/v1/analytical/Orgs(<orgID>)/JobLogEntries?<query parameters>
```

You can include the following filters in the URI:

Parameter	Type	Description
filter	String	Query filter.
endTime	Date	Date and time the task ended. Include in the filter parameter.

Parameter	Type	Description
startswith	String	Runtime environment, Secure Agent, or asset that started the task. Include in the filter parameter.
listfilter	String	Job status and asset type.
status in	String	Status of the task. You can filter by any of the following statuses: - COMPLETED - WARNING - FAILED - END - SUCCESS - STOPPED
assetType in	String	Type of asset: - MTT. Mapping task. - TASKFLOW. Advanced taskflow. - MI_TASK. Mass ingestion task. - DSS. Synchronization task. - WORKFLOW. Linear taskflow. - DRS. Replication task. - PCS. PowerCenter task. - DTT. Data transfer task. - DLT. Data loader task. - MAPPING - BATCH_MAPPING. Dynamic mapping task. - DMASK. Data masking task.
count	Int	If true, returns total records available for applied filter.
top	Int	Number of elements to return. For example, a value of 25 returns the first 25 records available for the applied filters. Max is 500.
skip	Int	Number of records to skip. For example, a value of 4 excludes the first four assets in the list.
orderby	String	Sort order in which to return the data.

GET request examples

The following examples show how you can use query parameters to request log information:

- To request a list of mapping tasks that successfully completed between October 8, 2023 at 6am and October 9, 2023 at 6am, you might use the following URI:

```
/cdiinsights-service/api/v1/analytical/Orgs(<orgID>)/JobLogEntries?$filter=(endTime
%20ge%202023-10-08T06:00:00Z)%20and%20(endTime%20le
%202023-10-09T06:00:00Z)&listFilter=status%20in%20(SUCCESS)%20and%20assetType%20in
%20(MTT)&$count=true&$top=500&$skip=0
```

- To request a list of all jobs that completed between October 1, 2023 at 6am and October 31, 2023 at 6am and sort the jobs by descending end time, you might use the following URI:

```
/cdiinsights-service/api/v1/analytical/Orgs(<orgID>)/JobLogEntries?$filter=(endTime
%20ge%202023-10-01T06:00:00Z)%20and%20(endTime%20le
%202023-11-31T06:00:00Z)&listFilter=status%20in
%20(COMPLETED,WARNING,FAILED,END,SUCCESS,STOPPED)%20and%20assetType%20in
%20(MTT,DTT,DMASK,PCS,DRS,DSS,MTT_TEST,BATCH_MAPPING,MAPPING,command,MI_TASK)&
$count=true&$top=500&$skip=0&$oderby=endTime desc
```

GET Response

Returns a jobLogEntry object for each row in the log. Returns the error object if errors occur.

When you request information for each row in the log, the jobLogEntry object includes the following attributes:

Field	Type	Description
logEntryId	String	Log entry ID.
assetId	String	ID for the task.
runId	Long	ID for the task run.
assetName	String	Name of the asset.
instanceName	String	Name of the job in the following format: <asset name>-<instance number>
orgId	String	Organization ID.
status	String	Whether the job completed successfully.
assetType	String	Type of task. For Data Integration jobs, returns one of the following codes: - MTT. Mapping task. - TASKFLOW. Advanced taskflow. - MI_TASK. Mass ingestion task. - DSS. Synchronization task. - WORKFLOW. Linear taskflow. - DRS. Replication task. - PCS. PowerCenter task. - DTT. Data transfer task. - DLT. Data loader task. - MAPPING. - BATCH_MAPPING. Dynamic mapping task. - DMASK. Data masking task.
startTime	Date/time	Start time for the task or linear taskflow. Uses Eastern Time Zone (ET).
endTime	Date/time	End time for the task or linear taskflow. Uses Eastern Time Zone (ET).
updateTime	String	Last time the task was updated.
duration	String	Time in seconds that the task ran.
errorMessage	String	Error message associated with the job.
location	String	Location of the asset.
rowsProcessed	String	Total number of rows processed by the task.
successRows	String	Number of rows that were successfully processed by the task.
errorRows	String	Number of rows that were not processed by the task.
runtimeEnv	String	Runtime environment where the task runs.

Field	Type	Description
runtimeEnvName	String	Name of the runtime environment where the task runs.
startedBy	String	User or schedule that started the task.
subtasks	String	Number of subtasks that the task contains.
runContext	String	Method through which the task was initiated. Includes the following values: - UI. Task was initiated through the user interface. - SCHEDULER. Task was initiated through the task scheduler. - REST-API. Task was initiated through the REST API. - OUTBOUND MESSAGE. Task was initiated through an outbound message.
extraData	String	Additional information including the saasLogId and saasTaskId.
parentEntityId	String	ID of the parent task.
updateDayHour	Date/Time	Last time the task was updated.
taskId	String	ID of the task.
stopOnError	Boolean	Whether or not the runtime environment stops the task when a nonfatal error occurs.
isAtScale	Boolean	Whether or not the mapping associated with the task is in advanced mode.
parAssetName	String	Name of the parent asset.
parRunId	String	Run ID for the parent asset.
parLocation	String	Location of the parent asset.
parAssetType	String	Parent asset type.
agentId	String	Secure Agent that runs the task.
agentName	String	Name of the agent that runs the task.
projectName	String	Project in which the asset resides.

GET response example

You might get the following response when you request job log information for mapping tasks:

```
{
  "@odata.context": "$metadata#JobLogEntries",
  "@odata.count": 2,
  "value": [
    {
      "logEntryId": "9hQOSzOu0GSeG0r73CzK8L",
      "assetId": "jn94HPOUK4zlEoO07eGfEq",
      "runId": "148201",
      "assetName": "CDI_9102",
      "instanceName": "CDI_9102-148201",
      "orgId": "010cdi",
      "status": "COMPLETED",
      "assetType": "MTT",
      "correlationId": "ZGVmN2E3ZDEtMTE2MC00NT",
      "startTime": "2023-07-20T07:55:52Z",
      "endTime": "2023-07-20T07:56:09Z",
```

```
            "updateTime": null,
            "duration": 17,
            "errorMessage": "",
            "location": "9102",
            "rowsProcessed": 2,
            "successRows": 2,
            "errorRows": 0,
            "runtimeEnv": "runtime environment",
            "runtimeEnvName": "ASH0001",
            "startedBy": "every_day",
            "subtasks": 0,
            "runContext": "SCHEDULER",
            "extraData": "{\"saasLogId\":\"010AXYC100000000RC5Q\", \"runtimeType\":\"\",
\"isServerless\":\"false\", \"releaseVersion\":\"4400\", \"saasTaskId
\":\"010AXY0Z0000000005EM\"}",
            "parentEntityId": null,
            "statusDisplayName": "SUCCESS",
            "updateDayHour": "2023-07-20T07:00:00Z",
            "taskId": "010AXY0Z0000000005EM",
            "stopOnError": "0",
            "requestId": null,
            "isAtScale": 0,
            "parAssetName": null,
            "parRunId": null,
            "parLocation": "n/a",
            "parAssetType": null,
            "agentId": "010AXY08000000000003",
            "agentName": "ASH0001",
            "projectName": "9102"
        }
        {
            "logEntryId": "7gQRt15YPsBcdK6RsozzSJ",
            "assetId": "iwm48QrytmidcBrdpMElu6",
            "runId": "145708",
            "assetName": "CDI-9845_test2",
            "instanceName": "CDI-9845_test2-145708",
            "orgId": "010cdi",
            "status": "COMPLETED",
            "assetType": "MTT",
            "correlationId": "NjNlNGFhY2QtYzFjZC00MT",
            "startTime": "2023-07-20T07:55:52Z",
            "endTime": "2023-07-20T07:56:03Z",
            "updateTime": null,
            "duration": 11,
            "errorMessage": "",
            "location": "Default",
            "rowsProcessed": 1000,
            "successRows": 1000,
            "errorRows": 0,
            "runtimeEnv": "runtime environment",
            "runtimeEnvName": "ASH0001",
            "startedBy": "CDI-9845_scheduler",
            "subtasks": 0,
            "runContext": "SCHEDULER",
            "extraData": "{\"saasLogId\":\"010AXYC100000000RC5P\", \"runtimeType\":\"\",
\"isServerless\":\"false\", \"releaseVersion\":\"4400\", \"saasTaskId
\":\"010AXY0Z0000000005N4\"}",
            "parentEntityId": null,
            "statusDisplayName": "SUCCESS",
            "updateDayHour": "2023-07-20T07:00:00Z",
            "taskId": "010AXY0Z0000000005N4",
            "stopOnError": "0",
            "requestId": null,
            "isAtScale": 0,
            "parAssetName": null,
            "parRunId": null,
            "parLocation": "n/a",
            "parAssetType": null,
            "agentId": "agentID",
            "agentName": "ASH0001",
```

```
        "projectName": "Default"
    }
```

mapping

Use this resource to request the details for a mapping or the details of all mappings in the organization.

GET Request

You can request the following information using a mapping GET request:

- Details of all mappings in the organization.
- Details for a particular mapping.
- An image of a mapping.

Details of all mappings in the organization

To request the details of all mappings in the organization, use the following URI:

`/api/v2/mapping`

Details for a particular mapping

To request the details of a particular mapping, include the mapping ID or mapping name in the URI. Use one of the following URIs:

`/api/v2/mapping/<id>`

`/api/v2/mapping/name/<name>`

If you use the mapping name in the URI and the mapping name value includes a space, replace the space with %20. For example:

`/api/v2/mapping/name/my%20mapping`

You can also request a specific mapping by name with the following URI:

`/api/v2/mapping/search?name=<name>`

Image of a mapping

To request an image of a mapping, specify the mapping ID and whether the mapping is deployed or not. Use the following URI:

`/api/v2/mapping/<id>/image?deployed=<true|false>`

For example:

`/api/v2/mapping/N0A1700000000001J/image?deployed=true`

GET Response

If successful, returns the mapping object for the requested mapping.

If you request the details for all mappings, returns the mapping object for every mapping in the organization without parameter details.

Returns the error object if errors occur.

The mapping object includes the following attributes:

Field	Type	Description
id	String	Mapping ID.
orgId	String	Organization ID.
name	String	Mapping name.
description	String	Description of the mapping.
createTime	Date/time	Time the mapping was created.
updateTime	Date/time	Last time the mapping was updated.
createdBy	String	User who created the mapping.
updatedBy	String	User who last updated the mapping.
bundleObjectId	String	ID of the bundle that includes the mapping, if applicable.
bundleVersion	String	Version of the bundle that includes the mapping, if applicable.
templateId	String	ID of the template created internally to represent the mapping.
deployTime	Date/time	Time the mapping was deployed.
hasParameters	Boolean	Indicates if the mapping includes parameters. Returns true or false.
valid	Boolean	Indicates if the mapping is valid. Returns true or false.
fixedConnection	Boolean	Indicates if the mapping has fixed connections. Returns true or false.
hasParametersDeployed	Boolean	Indicates if the mapping has parameters deployed. Returns true or false.
fixedConnectionDeployed	Boolean	Indicates if the mapping has fixed connections deployed. Returns true or false.
deployedTemplateId	String	ID of the template created internally to represent the deployed mapping.
tasks	Int	Number of tasks that use the mapping.
parameters		Parameters used in the mapping. Includes an mtTaskParameter object for each parameter.
id	Long	Included in the mtTaskParameter object. Parameter ID.
name	String	Included in the mtTaskParameter object. Parameter name.

Field	Type	Description
type	String	Included in the mtTaskParameter object. Parameter type.
description	String	Included in the mtTaskParameter object. Parameter description.
customFuncId	String	Mapplet ID for mapplet type parameters.

uiProperties	String	Included in the mtParameter object. Display property for the parameter. Includes the following information:
		- cnxtype. Connection type for the parameter.
		- logcnx. Logical connection.
		- order. Display order.
		- wizstep. Wizard step to display the parameter.
		- default. Default value.
		- visible. Whether the parameter is visible.
		- editable. Whether the parameter is editable.
		- required. Whether the parameter is required.
		- paramtype. UI control type for string parameters. Returns one of the following responses:
		  - Condition. Filter condition input control.
		  - Expression. Expression editor input control.
		  - Field. Field selection input control.
		  - Fieldmap. Field mapping input control. Includes the following attributes:
		    - lefttitle. Left title for the field mapping display.
		    - righttitle. Right title for the field mapping display.
		    - leftfs. Set of fields to display in the left table of the field mapping display.
		    - rightfs. Set of fields to display in the right table of the field mapping display.
		    - leftfilter. Regular expression to limit the fields that display in the left table of the field mapping display.
		    - rightfilter. Regular expression to limit the fields that display in the right table of the field mapping display.
		    - staticlist. List of fields to display on the right side of the field mapping display.

inOutParameters		In-out parameter used in the mapping. Includes a mtTaskInOutParameter object for each in-out parameter.
id	Long	Included in the mtTaskInOutParameter object. Parameter ID.
name	String	Included in the mtTaskInOutParameter object. Parameter name.

Field	Type	Description
description	String	Included in the mtTaskInOutParameter object. Description of the parameter.
initialValue	String	Included in the mtTaskInOutParameter object. Initial value for the parameter.
datatype	String	Included in the mtTaskInOutParameter object. Data type of the parameter.
precision	String	Included in the mtTaskInOutParameter object. Precision of the parameter.
scale	String	Included in the mtTaskInOutParameter object. Scale of the parameter.
retentionPolicy	String	Included in the mtTaskInOutParameter object. Determines when the task retains the current value.
aggregationType	String	Included in the mtTaskInOutParameter object. Determines the final current value of the parameter when the task runs.
currentValue	String	Included in the mtTaskInOutParameter object. Current value for the parameter.
mappingPreviewFileRecordId	String	ID of the image file that is used when previewing a mapping.
deployedMappingPreviewFileRecordId	String	ID of the image file that is used when previewing a deployed mapping.
references		Reference information. Returns the reference object, which includes the following attributes:
refObjectId	String	Included in the reference object.
refType	String	Included in the reference object.

GET Example

To request mapping details for all mappings in the organization, you might use the following request:

```
GET <serverUrl>/api/v2/mapping
Accept: application/xml
icSessionId: <icSessionId>
```

masterTemplate

Use this resource to request the details for a Visio template or the details of all Visio templates in the organization. You can create or update a Visio template, and request a list of mapping tasks that use the template. You can also delete a Visio template.

GET Request

To request the details of all Visio templates in the organization, use the following URI:

```
/api/v2/masterTemplate
```

To request the details of a particular Visio template, include the Visio template ID or Visio template name in the URI. Use one of the following URIs:

```
/api/v2/masterTemplate/<id>
```

```
/api/v2/masterTemplate/name/<name>
```

If you use the Visio template name in the URI and the Visio template name includes a space, replace the space with `%20`. For example:

```
/api/v2/masterTemplate/name/my%20Visio%20template
```

To request a list of mapping tasks that use a Visio template, use the Visio template ID in the following URI:

```
/api/v2/masterTemplate/<id>/tasks
```

GET Response

If successful, returns the masterTemplate object for the requested Visio template. If you request the details for all Visio templates, returns the masterTemplate object without parameter details for every Visio template in the organization.

Returns the error object if errors occur.

The masterTemplate object includes the following attributes:

Field	Type	Description
id	String	Visio template ID.
orgId	String	Organization ID.
name	String	Visio template name.
description	String	Description of the Visio template.
createTime	dateTime	Time the Visio template was created.
updateTime	dateTime	Last time the Visio template was updated.
createdBy	String	User who created the Visio template.
updatedBy	String	User who last updated the Visio template.
diFileRecordId	String	ID of the Visio template XML file.
templateImageId	String	ID of the Visio template image file.

Field	Type	Description
parameters		Parameters used in the Visio template. Includes an mtParameter object for each parameter.
id	Long	Included in the mtParameter object. Parameter ID.
name	String	Included in the mtParameter object. Parameter name.
label	String	Included in the mtParameter object. Parameter label.
type	String	Included in the mtParameter object. Parameter type.
description	String	Included in the mtParameter object. Parameter description.
customFuncId	String	Included in the mtParameter object. Mapplet ID for mapplet type parameters.

uiProperties	String	Included in the mtParameter object.
		Display property for the parameter. Includes the following information:
		- cnxtype. Connection type for the parameter.
		- logcnx. Logical connection.
		- order. Display order.
		- wizstep. Wizard step to display the parameter.
		- default. Default value.
		- visible. Whether the parameter is visible.
		- editable. Whether the parameter is editable.
		- required. Whether the parameter is required.
		- paramtype. UI control type for string parameters. Returns one of the following responses:
		  - Condition. Filter condition input control.
		  - Expression. Expression editor input control.
		  - Field. Field selection input control.
		  - Fieldmap. Field mapping input control. Includes the following attributes:
		    - lefttitle. Left title for the field mapping display.
		    - righttitle. Right title for the field mapping display.
		    - leftfs. Set of fields to display in the left table of the field mapping display.
		    - rightfs. Set of fields to display in the right table of the field mapping display.
		    - leftfilter. Regular expression to limit the fields that display in the left table of the field mapping display.
		    - rightfilter. Regular expression to limit the fields that display in the right table of the field mapping display.
		    - staticlist. List of fields to display on the right side of the field mapping display.

Field	Type	Description
sessionAttrs	String	General and performance session properties for the task. Can include values for the following attributes: - Write Backward Compatible Session Log File. Writes the session log to a file - Session Log File Name. Name for the session log. - Session Log File Directory. Directory where the session log is saved. - $Source Connection Value. Source connection name. - $Target Connection Value. Target connection name. - Treat Source Rows as. When the mapping task reads source data, it marks each row with an indicator to specify the operation to perform when the row reaches the target: - Insert. All rows are marked for insert into the target. - Update. All rows are marked for update in the target. - Delete. All rows are marked for delete from the target. - Data Driven. The task uses the Update Strategyobject in the data flow to mark the operation for each source row. - Commit Type. Commit type to use: - Source. Performs commits based on the number of source rows. - Target. Performs commits based on the number of target rows. - User Defined. Performs commits based on the commit logic defined in the Visio template. If you do not configure a commit type, the task performs a target commit. - Commit Interval. Interval in rows between commits. If you do not configure a commit interval, the task commits every 10,000 rows. - Commit on End of File. Commits data at the end of the file. Returns true or false. - Rollback Transactions on Errors. If the task encounters a non-fatal error, you can choose to roll back the transaction at the next commit point. When the task encounters a transformation error, it rolls back the transaction if the error occurs after the effective transaction generator for the target. - Java Classpath. Java classpath to use. - DTM Buffer Size. Amount of memory allocated to the task from the DTM process. - Incremental Aggregation. Performs incremental aggregation. Returns true or false. - Reinitialize Aggregate Cache. Overwrites existing aggregate files for an incremental aggregation task. Returns true or false. - Enable High Precision. Processes the Decimal data type to a precision of 28. Returns true or false. - Session Retry on Deadlock. The mapping task retries a write on the target when a deadlock occurs. Returns true or false.
wizardMetadata		Metadata for the mapping task wizard steps. Includes an mtWizardStep object for each step.
name	String	Included in the mtWizardStep object. Name of the step.
title	String	Included in the mtWizardStep object. The title of the step, displayed in the mapping task wizard user interface.

POST Request

To update a Visio template, use the Visio template ID in the following URI. To create a new Visio template, omit the optional Visio template ID.

```
/api/v2/masterTemplate/<id>
```

You can submit a partial update using partial mode. If you want to update a field in the mtParameter object using partial mode, you must include the name or type fields. To submit a request using partial mode, use a JSON request and include the following line in the header:

```
Update-Mode=PARTIAL
```

You can use the following attributes in a **masterTemplate** object:

Field	Type	Required	Description
name	String	Yes	Name of the Visio template.
description	String		Description of the Visio template.
diFileRecordId	String	Yes	Visio template XML file ID. Use the ID returned when you upload the file to the organization with the fileRecord resource.
templateImageId	String		Visio template image file ID. This ID is returned when you upload the file to the organization with the fileRecord resource.
parameters			Object that defines parameters associated with the template. Use an mtParameter object to define each parameter.
name	String	Yes	Include in the mtParameter object. Key field for the mtParameter collection. Parameter name.
label	String		Include in the mtParameter object. Parameter label.
type	String	Yes	Include in the mtParameter object. Key field for the mtParameter collection. Parameter type. Use one of the following values: - STRING - SOURCE - TARGET - MAPPLET - LOOKUP
description	String		Include in the mtParameter object. Parameter description.
customFuncId	String		Include in the mtParameter object. Mapplet ID for mapplet type parameters.

Field	Type	Required	Description

uiProperties	String		Include in the mtParameter object.
			
			Display property for the parameter. Use a UIPropertyType object to define the following display properties:
			
			- cnxtype. Connection type for the parameter. Use a valid connection type. For more information, see the connection resource.
			
			- logcnx. Logical connection.
			
			- order. Display order.
			
			- wizstep. Wizard step to display the parameter.
			
			- default. Default value.
			
			- visible. Whether the parameter is visible. Use True or False.
			
			- editable. Whether the parameter is editable. Use True or False.
			
			- required. Whether the parameter is required. Use True or False.
			
			- paramtype. UI control type for string parameters. Use one of the following values:
			
			  - Condition. Filter condition input control.
			
			  - Expression. Expression editor input control.
			
			  - Field. Field selection input control.
			
			  - Fieldmap. Field mapping input control. Include the following attributes:
			
			    - lefttitle. Left title for the field mapping display.
			    - righttitle. Right title for the field mapping display.
			
			    - leftfs. Set of fields to display in the left table of the field mapping display.
			
			    - rightfs. Set of fields to display in the right table of the field mapping display.
			
			    - leftfilter. Regular expression to limit the fields that display in the left table of the field mapping display.
			
			    - rightfilter. Regular expression to limit the fields that display in the right table of the field mapping display.
			
			    - staticlist. List of fields to display on the right side of the field mapping display. Use instead of rightfs.
			
			    List field names and associated data types separated by a line break or semicolon.

Field	Type	Required	Description
sessionAttrs	String		Include in the mtParameter object.
			General and performance session properties for the task. Can include values for the following attributes:
			- Write Backward Compatible Session Log File. Writes the session log to a file
			- Session Log File Name. Name for the session log.
			- Session Log File Directory. Directory where the session log is saved.
			- $Source Connection Value. Source connection name.
			- $Target Connection Value. Target connection name.
			- Treat Source Rows as. When the mapping task reads source data, it marks each row with an indicator to specify the operation to perform when the row reaches the target:
			- Insert. All rows are marked for insert into the target.
			- Update. All rows are marked for update in the target.
			- Delete. All rows are marked for delete from the target.
			- Data Driven. The task uses the Update Strategyobject in the data flow to mark the operation for each source row.
			- Commit Type. Commit type to use:
			- Source. Performs commits based on the number of source rows.
			- Target. Performs commits based on the number of target rows.
			- User Defined. Performs commits based on the commit logic defined in the Visio template.
			If you do not configure a commit type, the task performs a target commit.
			- Commit Interval. Interval in rows between commits. If you do not configure a commit interval, the task commits every 10,000 rows.
			- Commit on End of File. Commits data at the end of the file. Returns true or false.
			- Rollback Transactions on Errors. If the task encounters a non-fatal error, you can choose to roll back the transaction at the next commit point.
			When the task encounters a transformation error, it rolls back the transaction if the error occurs after the effective transaction generator for the target.
			- Java Classpath. Java classpath to use.
			- DTM Buffer Size. Amount of memory allocated to the task from the DTM process.
			- Incremental Aggregation. Performs incremental aggregation. Returns true or false.
			- Reinitialize Aggregate Cache. Overwrites existing aggregate files for an incremental aggregation task. Returns true or false.
			- Enable High Precision. Processes the Decimal data type to a precision of 28. Returns true or false.
			- Session Retry on Deadlock. The mapping task retries a write on the target when a deadlock occurs. Returns true or false.
wizardMetadata			Metadata for the mapping task wizard steps. Include an mtWizardStep object for each step.

Field	Type	Required	Description
name	String		Included in the mtWizardStep object. Name of the step.
title	String		Included in the mtWizardStep object. The title of the step, displayed in the mapping task wizard user interface.

POST Response

If the request to create or update a Visio template is successful, returns the master template object for the Visio template that you created or updated.

Returns the error code if errors occur.

DELETE Request

To delete a Visio template, use the Visio template ID in the following URI:

```
/api/v2/masterTemplate/<id>
```

DELETE Response

Returns the 200 response code if the request is successful.

Returns the error object if errors occur.

GET Example

To request a list of tasks that use a Visio template with an ID of 000043T1000003G, you might use the following request:

```
GET <serverUrl>/api/v2/masterTemplate/000043T1000003G/tasks
Accept: application/xml
icSessionId: <icSessionId>
```

If successful, returns the mtTask object with id, orgId, name, and masterTemplateId for each task that uses the Visio template.

mttask

Use this resource to request the details of a mapping task. You can also create, update, or delete a mapping task.

Note: You cannot use the REST API to create a mapping task based on a mapping that includes a mapplet.

GET request

To request the details of a mapping task, you can use the task ID, federated task ID, or task name. To find the federated task ID, use the lookup resource. The federated task ID is the value of the id field in the lookup response.

Use one of the following URIs:

```
/api/v2/mttask/<id>
```

```
/api/v2/mttask/frs/<id>
```

```
/api/v2/mttask/name/<name>
```

If you use the task name in the URI and the task name includes a space, replace the space with `%20`. For example:

```
/api/v2/mttask/name/task%20name
```

GET response

Returns the mtTask object for the requested task ID or task name.

Returns the error object if errors occurred.

The following table describes attributes in an **mtTask** object:

Field	Type	Description
id	String	Task ID.
orgId	String	Organization ID.
name	String	Task name.
agentId	String	Agent that runs the task.
runtimeEnvironmentId	String	Runtime environment used for the task.
maxLogs	Long	Number of session log files and import log files Data Integration retains.
description	String	Description.
createTime	Date/time	Time the task was created.
updateTime	Date/time	Last time the task was updated.
createdBy	String	User who created the task.
updatedBy	String	User who last updated the task.
schemaMode	String	Mode in which Data Integration refreshes the data object schema.
errorTaskEmail		Object that includes the taskEmail object for error notifications
id	String	Included in taskEmail object for errorTaskEmail. ID.
emails	String	Included in taskEmail object for errorTaskEmail. Email address that receives email notification when a task fails to complete.
successTaskEmail		Object that includes the taskEmail object for success notifications.
id	String	Included in taskEmail object for successTaskEmail. ID.
emails	String	Included in taskEmail object for successTaskEmail. Email address that receives email notification when a task completes successfully.
warningTaskEmail		Object that includes the taskEmail object for warning notifications.

Field	Type	Description
id	String	Included in taskEmail object for warningTaskEmail. ID.
emails	String	Included in taskEmail object for warningTaskEmail. Email address that receives email notification when a task completes with errors.
parameters		Parameters associated with the task. Includes attributes in the mtTaskParameter object for each parameter.
id	Long	Included in the mtTaskParameter object. Parameter ID.
name	String	Included in the mtTaskParameter object. Parameter name.
type	String	Included in the mtTaskParameter object. Parameter type.
text	String	Included in the mtTaskParameter object. Parameter value.
label	String	Included in the mtTaskParameter object. Parameter label.
description	String	Included in the mtTaskParameter object. Parameter description.
sourceConnectionId	String	Included in the mtTaskParameter object. Source connection ID.
targetConnectionId	String	Included in the mtTaskParameter object. Target connection ID.
lookupConnectionId	String	Included in the mtTaskParameter object. Lookup connection ID.
transfConnectionId	String	Included in the mtTaskParameter object. Connection ID of mapplet. Reserved for future use.
midstreamConnectionId	String	Included in the mtTaskParameter object. Connection ID of midstream transformation.
sourceObject	String	Included in the mtTaskParameter object. Source object name.
sourceObjectLabel	String	Included in the mtTaskParameter object. Source object label.

Field	Type	Description
targetObject	String	Included in the mtTaskParameter object. Target object name.
targetObjectLabel	String	Included in the mtTaskParameter object. Target object label.
lookupObject	String	Included in the mtTaskParameter object. Lookup object name.
lookupObjectLabel	String	Included in the mtTaskParameter object. Lookup object label.
midstreamObject	String	Included in the mtTaskParameter object. Midstream object name.
midstreamObjectLabel	String	Included in the mtTaskParameter object. Midstream object label.
newObject	Boolean	Included in the mtTaskParameter object. Whether the application creates a new flat file target. Returns True when it creates a target.
newObjectName	String	Included in the mtTaskParameter object. Name of the flat file target.
operationType	String	Included in the mtTaskParameter object. The task operation for the target.
truncateTarget	Boolean	Included in the mtTaskParameter object. Whether the application truncates a database target before writing to it. Returns True when it truncates the target.
srcFFAttrs		Included in the mtTaskParameter object. Object that contains the source file attributes in the flatFileAttrs object.
tgtFFAttrs		Included in the mtTaskParameter object. Object that contains the target file attributes in the flatFileAttrs object.
lkpFFAttrs		Included in the mtTaskParameter object. Object that contains the lookup file attributes in the flatFileAttrs object.
flatFileAttrs		Object that includes attributes for the source, target, and lookup files.
id	Long	Included in the flatFileAttrs object. Field ID.
delimiter	String	Included in the flatFileAttrs object. Character used to separate fields

Field	Type	Description
textQualifier	String	Included in the flatFileAttrs object. Quote character that defines the boundaries of text strings
escapeChar	String	Included in the flatFileAttrs object. Character immediately preceding a field delimiter character embedded in an unquoted string, or immediately preceding the quote character in a quoted string
headerLineNo	Int	Included in the flatFileAttrs object. Number of header lines
firstDataRow	Int	Included in the flatFileAttrs object. The row number where the data begins in the file.
rowDelimiter	Int	Included in the flatFileAttrs object. Line break character. Returns the decimal code for an ASCII character.
consecutiveDelimiter	Boolean	Included in the flatFileAttrs object. Indicates whether one or more consecutive delimiters are treated as one.
multiDelimitersAsAnd	Boolean	Included in the flatFileAttrs object. If the delimiter is more than one character, indicates whether the characters are treated as a single delimiter or multiple delimiters.
customFuncCfg		Included in the mtTaskParameter object. Object that defines configuration for mapplets used in the task. Includes attributes in the customFuncConfig object for each mapplet.
id	Long	Included in the customFuncConfig object. Mapplet ID.
connections		Included in the customFuncConfig object. Object to define connections used in a mapplet. Includes information in the pcsConnection object for each connection.
id	Long	Included in the pcsConnection object.
name	String	Included in the pcsConnection object. Connection name.
type	String	Included in the pcsConnection object. Connection type.
subtype	String	Included in the pcsConnection object. Connection subtype.
description	String	Included in the pcsConnection object. Description of the connection.
connectionId	String	Included in the pcsConnection object. Connection ID.

Field	Type	Description
showBusinessNames	Boolean	Included in the mtTaskParameter object. Whether the task displays business names. Returns True when it shows business names.
naturalOrder	Boolean	Included in the mtTaskParameter object. The order that the task uses to display fields. Returns True for the order returned by the connection. Returns False for alphabetic order.
isRESTModernSource	Boolean	Included in the mtTaskParameter object. Always set to True to enable extended objects.
customQuery	String	Included in the mtTaskParameter object. The custom query specified in Mapping Designer or mapping task query options.
overriddenFields		Included in the mtTaskParameter object. Changes to field metadata in the mapping task. Includes information in the mtTaskOverriddenField object for each overridden field.
name	String	Included in the mtTaskOverriddenField object. Field name.
type	String	Included in the mtTaskOverriddenField object. Field type.
precision	Int	Included in the mtTaskOverriddenField object. Length of the field in bytes.
scale	Int	Included in the mtTaskOverriddenField object. Number of digits after the decimal point for numeric values.
platformType	String	Included in the mtTaskOverriddenField object. Platform data type for the field.
tgtFieldRefs	String	Included in the mtTaskParameter object. Salesforce field reference IDs.
extendedObject		Included in the mtTaskParameter object. The source or target with more than one object joined.
targetUpdateColumns	String	Included in the mtTaskParameter object. List of column names used to update records in the target object.
runtimeAttrs	String	Included in the mtTaskParameter object. Advanced connection properties for connections used in a task. Use a runtimeAttrs object to define key-value pairs of advanced connection properties. Use an entry object for each key-value pair. For the attribute name, use the advanced connection property name as displayed in the Data Integration user interface. For more information about advanced connection properties, see the Data Integration help.

Field	Type	Description
dataFormat		Included in the mtTaskParameter object. Data format provided by the connector. Includes attributes in the dataFormat object for each connector. The dataFormat object is not applicable to all connectors. To see if dataFormat is applicable to the connectors you are using, see the help for the relevant connectors.
formatId		Included in the dataFormat object. Data format type provided by the connector, such as FLAT, AVRO, PARQUET, JSON, or XML.
dataFormatAttributes		Included in the dataFormat object. Format attributes for the data format type. For example, for a flat file, the dataFormatAttributes object includes values such as escapeChar, delimiter, and qualifier.
sequences		Defines values for the Sequence Generator transformation. Includes the sequenceDefinition object for each sequence transformation.
txName	String	Included in the sequenceDefinition object. Name of the Sequence Generator transformation.
initialValue	String	Included in the sequenceDefinition object. The initial value of the sequence.
currentValue	String	Included in the sequenceDefinition object. The value used for the last row added to the transformation.
inOutParameters		In-out parameter used in the task. Includes a mtTaskInOutParameter object for each in-out parameter.
id	Long	Included in the mtTaskInOutParameter object. Parameter ID.
name	String	Included in the mtTaskInOutParameter object. Parameter name.
description	String	Included in the mtTaskInOutParameter object. Description of the parameter.
initialValue	String	Included in the mtTaskInOutParameter object. Initial value for the parameter.
datatype	String	Included in the mtTaskInOutParameter object. Data type of the parameter.
precision	String	Included in the mtTaskInOutParameter object. Precision of the parameter.
scale	String	Included in the mtTaskInOutParameter object. Scale of the parameter.

Field	Type	Description
retentionPolicy	String	Included in the mtTaskInOutParameter object. Determines when the task retains the current value.
aggregationType	String	Included in the mtTaskInOutParameter object. Determines the final current value of the parameter when the task runs.
currentValue	String	Included in the mtTaskInOutParameter object. Current value for the parameter.
lastRunTime	Date/time	Time the task last run.
masterTemplateId	String	Visio template ID. Returned when a Visio template is the basis of the task.
mappingId	String	Mapping ID. Returned when a mapping is the basis for the task.
scheduleId	String	Schedule associated with the task, if any.
shortDescription	String	The first 50 characters of the description.
sessionProperties	String	Advanced session properties associated with the task. Includes advanced session properties in a sessionProperties object.
outboundMessageUrlToken	String	Outbound message URL token for the task, if it exists.
outboundMessageUrlQueueTime	Long	Outbound message URL queue time for the task, if it exists.
preProcessingCmd	String	Command to run before the task.
postProcessingCmd	String	Command to run after the task completes.
parameterFileName	String	The name of the parameter file used in the task.
verbose	Boolean	Whether Data Integration generates additional data in the logs to use for troubleshooting purposes. Returns True or False.
connRuntimeAttrs		Included in the mtTaskParameter parameter. Includes an mtTaskConnRuntimeAttr object for each connector. The connRuntimeAttrs object applies to the CDC connectors.
id	String	Included in the mtTaskConnRuntimeAttr object. Internal id for each mtTaskConnRuntimeAttr object.
name	String	Included in the mtTaskConnRuntimeAttr object. The internal ID of a connection runtime attribute for a CDC connector.

Field	Type	Description
value	String	Included in the mtTaskConnRuntimeAttr object. The value associated with the name attribute.
connectionID	String	Included in the mtTaskConnRuntimeAttr object. The CDC connection ID.

POST request

To create a mapping task, use the following URI:

```
/api/v2/mttask/
```

If you want to specify a location for the task, include the container ID in the request. If the container ID isn't included in the request, the task is created in the Default folder. You can find the container ID for a project or folder in the Data Integration user interface. On the **Explore** page, select the folder. In the URL, the last string of characters is the container ID.

For example, in the following URL, the container ID is dH2DuGJYda7ijgW4Sm32sR:

```
https://na1.dm-us.informaticacloud.com/diUI/products/integrationDesign/main/Explore/
dH2DuGJYda7ijgW4Sm32sR
```

To update a mapping task, include the task ID or federated task ID in the URI. To find the federated task ID, use the lookup resource. The federated task ID is the value of the id field in the lookup response.

Use one of the following URIs:

```
/api/v2/mttask/<id>
```

```
/api/v2/mttask/frs/<id>
```

You can submit a partial update using partial mode. If you want to update a field that is within a collection using partial mode, you must include the key field for the collection. The following table lists the collections in the mttask resource and the corresponding key fields:

Collection	Key Fields
mtTaskInOutParameter	name
sequenceDefinition	txName
mtTaskOverriddenField	name
mtTaskParameter	name type
objects	name

To submit a request using partial mode, use a JSON request and include the following line in the header:

```
Update-Mode=PARTIAL
```

The following table describes the attributes you can include in an mtTask object:

Field	Type	Required	Description
name	String		Name of the task.
containerId	String		ID of the project or folder to contain the task. If not included in the request, the task is created in the Default folder.
description	String		Description of the task.
runtimeEnvironmentId	String	Yes	ID of the runtime environment used for the task.
mappingId	String	Required when a mapping is the basis for task.	ID of the mapping used in the task.
scheduleId	String		Schedule associated with the task, if any.
sessionProperties	String		Advanced session properties. Use a sessionProperties object to define key-value pairs of advanced session properties. Use an entry object for each key-value pair. For the attribute name, use the advanced session property name as displayed in the Data Integration user interface.
schemaMode	String		Mode in which Data Integration refreshes the data object schema. Use one of the following values: - async - dynamic Default is async. If you run multiple mapping tasks at the same time, Data Integration picks up the latest schema regardless of the schema mode.
errorTaskEmail			Object that includes the taskEmail object for error notifications
id	String		Include in taskEmail object for errorTaskEmail. ID.
emails	String		Include in taskEmail object for errorTaskEmail. Email address that receives email notification when a task fails to complete.
successTaskEmail			Object that includes the taskEmail object for success notifications.

Field	Type	Required	Description
id	String		Include in taskEmail object for successTaskEmail. ID.
emails	String		Include in taskEmail object for successTaskEmail. Email address that receives email notification when a task completes successfully.
warningTaskEmail			Object that includes the taskEmail object for warning notifications.
id	String		Include in taskEmail object for warningTaskEmail. ID.
emails	String		Include in taskEmail object for warningTaskEmail. Email address that receives email notification when a task completes with errors.
parameters			Parameters associated with the task. Use an mtTaskParameter object to define the following attributes for each parameter.
id	Long		Include in the mtTaskParameter object. System generated parameter ID. You cannot update this value
name	String		Include in the mtTaskParameter object. Parameter name. Key field for the mtTaskParameter collection.
type	String		Include in the mtTaskParameter object. Parameter type. Key field for the mtTaskParameter collection. Use one of the following values: - STRING - SOURCE - TARGET - MAPPLET - LOOKUP
text	String		Include in the mtTaskParameter object. Parameter value.
label	String		Include in the mtTaskParameter object. Parameter label.
description	String		Include in the mtTaskParameter object. Parameter description.

Field	Type	Required	Description
sourceConnectionId	String		Include in the mtTaskParameter object. Source connection ID.
targetConnectionId	String		Include in the mtTaskParameter object. Target connection ID.
lookupConnectionId	String		Include in the mtTaskParameter object. Lookup connection ID.
newFlatFile	Boolean		Include in the mtTaskParameter object. Whether Data Integration creates a new flat file target. Use one of the following values: - True. - False.
flatFileName	String		Include in the mtTaskParameter object. Name of the flat file target.
newObject	Boolean		Include in the mtTaskParameter object. Whether the application creates a new flat file target. Returns True when it creates a target.
newObjectName	String		Include in the mtTaskParameter object. Name of the flat file target.
operationType	String		Include in the mtTaskParameter object. The task operation for the target. Use one of the following values: - Insert - Upsert - Update - Delete - Rowbased **Note:** The Rowbased value corresponds to the Data Driven value in the user interface.
dataDrivenCondition	String		Include in the mtTaskParameter object. Applicable when operationType is Rowbased. Defines expressions that flag rows for an insert, update, delete, or reject operation. For example: `IIF(ISNULL (ISDELETED), DD_INSERT`
truncateTarget	Boolean		Include in the mtTaskParameter object. Whether the application truncates a database target before writing to it. Use one of the following values: - True. - False.

Field	Type	Required	Description
srcFFAttrs			Include in the mtTaskParameter object. Object for the source file attributes. Include the attributes in the flatFileAttrs object.
tgtFFAttrs			Include in the mtTaskParameter object. Object for the target file attributes. Include the attributes in the flatFileAttrs object.
lkpFFAttrs			Include in the mtTaskParameter object. Object for the lookup file attributes. Include the attributes in the flatFileAttrs object.
flatFileAttrs			Object to hold attributes for the source, target, and lookup files.
id	Long		Include in the flatFileAttrs object. Field ID.
delimiter	String		Include in the flatFileAttrs object. Character used to separate fields.
textQualifier	String		Include in the flatFileAttrs object. Quote character that defines the boundaries of text strings.
escapeChar	String		Include in the flatFileAttrs object. Character immediately preceding a field delimiter character embedded in an unquoted string, or immediately preceding the quote character in a quoted string.
headerLineNo	Int		Include in the flatFileAttrs object. Number of header lines.
firstDataRow	Int		Include in the flatFileAttrs object. The row number where the data begins in the file.
rowDelimiter	Int		Include in the flatfileAttrs object. Line break character. Enter the decimal code for an ASCII character between 1 and 32. Default is 10, which is the line feed character.
consecutiveDelimiter	Boolean		Included in the flatFileAttrs object. Indicates whether one or more consecutive delimiters are treated as one. Default is false.

Field	Type	Required	Description
multiDelimitersAsAnd	Boolean		Included in the flatFileAttrs object.
			If the delimiter is more than one character, indicates whether the characters are treated as a single delimiter or multiple delimiters. Default is true.
customFuncCfg			Include in the mtTaskParameter object.
			Object to define configuration for mapplets used in the task. Use a customFuncConfig object to define each mapplet.
id	Long		Include in the customFuncConfig object.
			Mapplet ID.
connections			Include in the customFuncConfig object.
			Object to define connections used in a mapplet. Use a pcsConnection object for each connection.
			For more information about connections, see "connection" on page 267.
id	Long		Include in the pcsConnection object.
name	String		Include in the pcsConnection object. Connection name.
type	String		Include in the pcsConnection object. Connection type.
subtype	String		Include in the pcsConnection object. Connection subtype.
description	String		Include in the pcsConnection object. Description of the connection.
connectionId	String		Include in the pcsConnection object. Connection ID.
overriddenFields			Include in the mtTaskParameter object.
			Changes to field metadata in the mapping task. Use the mtTaskOverriddenField object for each overridden field.
name	String		Include in the mtTaskOverriddenField object.
			Field name. Key field for the mtTaskOverriddenField collection.

Field	Type	Required	Description
type	String		Include in the mtTaskOverriddenField object. Field type.
precision	Int		Include in the mtTaskOverriddenField object. Length of the field in bytes.
scale	Int		Include in the mtTaskOverriddenField object. Number of digits after the decimal point for numeric values.
platformType	String		Include in the mtTaskOverriddenField object. Platform data type for the field.
tgtFieldRefs	String		Include in the mtTaskParameter object. Salesforce field reference IDs.
runtimeAttrs	String		Include in the mtTaskParameter object. Advanced connection properties for connections used in a task. Use a runtimeAttrs object to define key-value pairs of advanced connection properties. Use an entry object for each key-value pair. For the attribute name, use the advanced connection property name as displayed in the Data Integration user interface. For more information about advanced connection properties, see the Data Integration help.
parameterFileName	String		Include in the mtTaskParameter object. Name of the parameter file used in the task.
parameterFileDir	String		Include in the mtTaskParameter object. Path for the directory that contains the parameter file.
dataFormat			Include in the mtTaskParameter object. Data format provided by the connector. Include attributes in the dataFormat object for each connector. The dataFormat object is not applicable to all connectors. To see if dataFormat is applicable to the connectors you are using, see the help for the relevant connectors.

Field	Type	Required	Description
formatId	String		Include in the dataFormat object. Data format type provided by the connector, such as FLAT, AVRO, PARQUET, JSON, or XML.
fwConfigId	String	Required when formatID is set to Flat and fixed-width mode is used.	Include in the dataFormat object. Valid fixed-width format ID.
dataFormatAttributes	String		Include in the dataFormat object. Format attributes for the data format type. For example, for a flat file data format, the dataFormatAttributes object includes values such as escapeChar, delimiter, and qualifier.
inOutParameters			In-out parameter used in the task. Include an mtTaskInOutParameter object for each in-out parameter.
id	Long		Include in the mtTaskInOutParameter object. Parameter ID.
name	String		Include in the mtTaskInOutParameter object. Parameter name. Key field in the mtTaskInOutParameter collection.
description	String		Include in the mtTaskInOutParameter object. Description of the parameter.
initialValue	String		Include in the mtTaskInOutParameter object. Initial value for the parameter.
datatype	String		Include in the mtTaskInOutParameter object. Data type of the parameter.
precision	String		Include in the mtTaskInOutParameter object. Precision of the parameter.
scale	String		Include in the mtTaskInOutParameter object. Scale of the parameter.

Field	Type	Required	Description
retentionPolicy	String		Include in the mtTaskInOutParameter object. Determines when the task retains the current value. Include one of the following values: - ON_SUCCESS_OR_WARNING - ON_SUCCESS - ON_WARNING - NEVER
aggregationType	String		Include in the mtTaskInOutParameter object. Determines the final current value of the parameter when the task runs.
currentValue	String		Include in the mtTaskInOutParameter object. Current value for the parameter.
masterTemplateId	String	Required when task uses a Visio template.	ID of the Visio template used in the task.
outboundMessageUrlToken	String		Outbound message URL token for the task, if it exists.
outboundMessageUrlQueueTime	Long		Outbound message URL queue time for the task, if it exists.
preProcessingCmd	String		Command to run before the task.
postProcessingCmd	String		Command to run after the task completes.
maxLogs	Long		Number of session log files and import log files to retain. By default, Data Integration stores each type of log file for 10 runs before it overwrites the log files for new runs.
verbose	Boolean		Whether to generate additional data in the logs to use for troubleshooting purposes. Use True or False.
agentId	String		Agent that runs the task.
sequences			Defines values for the Sequence Generator transformation. Use a sequenceDefinition object for each sequence transformation.
txName	String		Include in the sequenceDefinition object. Name of the Sequence Generator transformation. Key field in the sequenceDefinition collection.

Field	Type	Required	Description
initialValue	String		Include in the sequenceDefinition object. The initial value of the sequence.
currentValue	String		Include in the sequenceDefinition object. The value used for the last row added to the transformation.
connRuntimeAttrs			Include in the mtTaskParameter parameter. Include an mtTaskConnRuntimeAttr object for each CDC connector for which runtime attributes will be changed. The connRuntimeAttrs object applies to the CDC connectors.
id	String	Required for connRuntimeAttrs object.	Include in the mtTaskConnRuntimeAttr object. Include the internal ID of the mtTaskConnRuntimeAttr object.
name	String	Required for mtTaskConnRuntimeAttr.	Include in the mtTaskConnRuntimeAttr object. Use one of the following names for the runtime attributes: - 101 for Maximum Rows Per Commit. - 102 for Minimum Rows Per Commit. - 103 for Maximum Latency in Seconds. - 104 for Real-time Flush Latency in Milliseconds. - 105 for Restart Point. - 106 for Restart Revision. - 107 for UOW Count. - 108 for Update as Delete or Insert. - 110 for Restart Option.

Field	Type	Required	Description
value	String		Include in the mtTaskConnRuntimeAttr object. Use a valid value for the specified attribute name: - Maximum Rows Per Commit. 0 to 999999999. - Minimum Rows Per Commit. 0 to 999999999. - Maximum Latency in Seconds. 2 to 360. - Real-time Flush Latency in Milliseconds. -1 to 999999999. - Restart Point. Set in conjunction with the Restart Option. 0 for earliest available. An empty string for end of log. A valid timestamp for specific timestamp in the log. A valid PWX Token to restart at a specific token in the log. - Restart Revision. A valid revision number from 0 to 2147483647. - UOW Count. -1 to 999999999. - Update as Delete or Insert. 0 to process as update. 1 to process as delete and insert. - Restart Option. Set in conjunction with the Restart Point. 0 for earliest available. 1 for end of log. 2 for time-based. 3 for PWX token.
connectionID	String	Required for mtTaskConnRuntimeAttr object.	Include in the mtTaskConnRuntimeAttr object. Include the CDC connection ID.

POST response

If successful, returns the mtTask object that you created or updated. Returns the error object if errors occur.

DELETE request

To delete a mapping task, use the task ID in the following URI:

```
/api/v2/mttask/<id>
```

Note: You cannot use the federated task ID to delete a mapping task.

DELETE response

Returns the 200 response code if the request is successful.

Returns the error object if errors occur.

POST example

To create a new mapping task with XML, you might use the following request:

```
POST <serverUrl>/api/v2/mttask
Content-Type: application/xml
Accept: application/xml
icSessionId: <icSessionId>

<mtTask>
```

```
          <errorTaskEmail>
            <taskEmail>
              <emails>email_dev@company.com, email2_dev@company.com</emails>
            </taskEmail>
          </errorTaskEmail>
          <successTaskEmail>
            <taskEmail>
              <emails>reviewer@company.com</emails>
            </taskEmail>
          </successTaskEmail>
          <warningTaskEmail>
            <taskEmail>
              <emails>email_dev@company.com, email2_dev@company.com</emails>
            </taskEmail>
          </warningTaskEmail>
          <parameters>
            <mtTaskParameter>
              <name>sort convert plugin</name>
              <type>MAPPLET</type>
            </mtTaskParameter>
          </parameters>
          <parameters>
            <mtTaskParameter>
              <name>DB lookup</name>
              <type>LOOKUP</type>
            </mtTaskParameter>
          </parameters>
          <sessionProperties>
              <entry>
                  <key>Java Classpath</key>
                  <value>C:/test/classpathnew</value>
              </entry>
              <entry>
                  <key>Pushdown Optimization</key>
                  <value>To Source</value>
              </entry>
              <entry>
                  <key>Write Backward Compatible Session Log File</key>
                  <value>no</value>
              </entry>
           </sessionProperties>
          <runtimeEnvironmentId>00000398D00000004</runtimeEnvironmentId>
          <sequences>
            <sequenceDefinition>
              <txName>SeqGen1</txName>
              <initialValue>1</initialValue>
              <currentValue>62</currentValue>
            </sequenceDefinition>
          </sequences>
          <preProcessingCmd>echo CurrentDate is 'date'</preProcessingCmd>
          <postProcessingCmd>echo PR-PostProcess</postProcessingCmd>
          <masterTemplateId>00034234M00000R</masterTemplateId>
      </mtTask>
```

A successful request returns the mtTask object.

Mask Rule Parameter Attributes for Masking Techniques

Define the parameter attribute values of a mask rule parameter when you run the mapping task. The attributes that you define depend on the masking technique that you apply.

For example, to mask a billing city field with the Substitution City masking technique, define the following attributes:

```
[
  {
    "referenceField": "BillingCity",
    "pcType": "string",
```

```
        "precision": 40,
        "paramMap": {
          "isSeeded": "TRUE",
          "seedValue": "190",
          "dicName": "informatica_mask_us_towns.dic",
          "outputPort": "TOWNNAMES",
        },
        "maskingType": "Substitution City"
      }
    ]
```

The following table lists the attributes that you define for each masking technique:

Masking Technique	Attributes
Credit Card	- isSeeded - seedValue - keepCardIssuer - targetIssuer
Custom Substitution	- DicConn - DicName - outputPort - isSeeded - seedValue
Email address	- isSeeded - seedValue
IP address	- isSeeded - seedValue
Key Date	- isSeeded - seedValue
Key Numeric	- override - isSeeded - seedValue
Key String	- isSeeded - seedValue - useMaskFormat - maskFormat - useSrcFilter - srcFilterOption - srcFilterStr - useTargetFilter - targetFilterOption - targetFilterStr
Phone	- isSeeded - seedValue
Random Date	- useRange - minWidth - maxWidth - useBlurring - blurringUnit - blurLow - blurHigh

Masking Technique	Attributes
Random Numeric	- useRange - minWidth - maxWidth - useBlurring - blurringOption - blurLow - blurHigh
Random String	- useRange - minWidth - maxWidth - useMaskFormat - useSrcFilter - srcFilterStr - useTargetFilter - targetFilterOption - targetFilterStr
SIN	- isSeeded - seedValue - startDigit - startDigitValue
SSN	- isSeeded - seedValue
Substitution City	- isSeeded - seedValue - DicName - outputPort
Substitution Country	- isSeeded - seedValue - DicName - outputPort
Substitution Female Name	- isSeeded - seedValue - DicName - outputPort
Substitution Last Name	- isSeeded - seedValue - DicName - outputPort
Substitution Male Name	- isSeeded - seedValue - DicName - outputPort
Substitution Name	- isSeeded - seedValue - DicName - outputPort

Masking Technique	Attributes
Substitution Position	- isSeeded - seedValue - DicName - outputPort
Substitution State	- isSeeded - seedValue - DicName - outputPort
Substitution Street	- isSeeded - seedValue - DicName - outputPort
Substitution U.S. ZIP code	- isSeeded - seedValue - DicName - outputPort
URL	- isSeeded - seedValue

Mask Rule Parameter Attribute Values

Define the required parameter attribute values when you run the mapping task.

The following table describes the attributes and values that you define for the mask rule parameter:

Attribute	Description
blurHigh	Required. The higher bound for blurring. You can specify the value in digits. Default is 0.
blurLow	Required. The lower bound for blurring. You can specify the value in digits. Default is 0.
blurringOption	Required. The unit of blurring for a numeric port. You can specify the following values: - Percent. Blurs the data based on a percent value. - Fixed. Blurs the data based on a fixed value.
blurringUnit	Required. The unit of blurring for a date port. You can specify the following values: - Year. Blurs the year value. - Month. Blurs the month value. - Day. Blurs the day value. - Hour. Blurs the hour value. - Minute. Blurs the minute value. - Second. Blurs the second value. Default is Year.

Attribute	Description
delimiter	Delimiter to separate the first name and last name in a masked email address. You can specify the value as: - . - - - _
DicConn	The connection that contains the dictionary files. Create a flat file connection that points to the directory with the dictionary files. Specify the flat file connection name.
dicName	The name of the flat file dictionary file. The dictionary file must be present in the `rdtmDir` directory of the Secure Agent.
domainConstantValue	Domain name to use in masked email addresses. Default is company.com.
expText	An attribute to configure an expression.
firstNameColumn	The first name column to use in masked email addresses. Specify the name of the port.
firstNameLength	The length of the first name in a masked email address. You can specify the value in digits. Default is 5.
isSeeded	An attribute to configure repeatable output. You can specify the following values: - TRUE. Masks the data with repeatable output. When true, specify a seed value. - FALSE. Masks the data with random output. Default is TRUE.
keepCardIssuer	Masks a credit card field with a credit card number from the same issuer. You can specify the following values: - TRUE. Retains the same card issuer in the masked data. - FALSE. Uses a specified card issuer in the masked data. When false, define the targetIssuer attribute. Default is TRUE.
lastNameColumn	The last name column to use in masked email addresses. Specify the name of the port.
lastNameLength	The maximum length of the last name in masked email addresses. You can enter the value in digits. Default is 5.

Attribute	Description
maskFormat	Defines the type of character to substitute for each character in the input data. You can limit each character to an alphabetic, numeric, or alphanumeric character type. Use the following characters to define a mask format: - A. Alphabetic - D. Digits 0-9 - N. Alphanumeric - X. Any character - R. Rest of the characters. Specify the value as ADNX+R. R must appear as the last character. For example, to ensure the masked output begins with an alphabet, enter the value as A+R. Default is R.
maxWidth	Required. The minimum value for the range. Enter the value in digits. Default is 0.
maxWidth	Required. The maximum value for the range. Enter the datetime value. Default is 01/19/2038 03:13:59.
minWidth	Required. The minimum value for the range. Enter the datetime value. Default is 01/01/1970 00:00:00.
minWdth	Required. The minimum value for the range. Enter the value in digits. Default is 0.
outputPort	The output port column from the dictionary.
seedValue	The seed value. Specify a value between 1 and 999. Default is 190.
srcFilterOption	Required. The type of filter to apply to source filter characters. You can specify the following values: - Mask Only. Masks only the specified characters in the source. - Mask all except. Masks all characters in the source except the characters specified.
srcFilterStr	Required. Defines the characters in the source string that you want to mask.
startDigit	Required. Defines the first digit of the masked SIN. You can specify the following values: - TRUE. Uses the digit that you specify as the first digit of the masked SIN. - FALSE. Uses a random digit as the first digit of the masked SIN. Default is FALSE. When true, define the startDigitValue attribute.
startDigitValue	Required. Defines the first digit of the masked SIN. Specify a value between 0 and 9. Default is 0.
targetFilterOption	Required. The type of filter to apply on target filter characters. You can specify the following values: - Use Only. Uses only the target characters that you specify. - Use All Except. Uses all characters in the target except what you specify.

Attribute	Description
targetFilterStr	Required. Substitutes the characters in a target string with the characters that you define in target filter characters. For example, enter the following characters to configure the masked output to contain all uppercase alphabetic characters: ABCDEFGHIJKLMNOPQRSTUVWXYZ.
targetIssuer	Required. Masked values contain credit card numbers from the issuer that you select. You can specify the following values: - ANY - JCB - VISA - AMEX - DISCOVER - MASTERCARD
useBlurring	Required. Masks dates based on a variance that you apply to a unit of the date. The masked date is within the variance. You can specify the following values: - TRUE. Applies a variance that you specify on a unit of the date. - FALSE. Does not apply a variance. Default is FALSE.
useMaskFormat	Specifies a mask format. You can specify the following values: - TRUE. Masks the data based on a format that you specify. - FALSE. Masks the data in a random format. Default is TRUE. If true, define the maskFormat attribute.
useRange	Required. Specifies a return value between the minimum and maximum values of the range based on field precision. You can specify the following values: - TRUE. Masks the data within a range that you specify. - FALSE. Does not use a specified range to mask the data. To define the range, configure the minimum and maximum ranges or configure a blurring range based on a variance from the original source value. Default is FALSE.
useSrcFilter	Specifies the characters in the source string that you want to mask. You can specify the following values: - TRUE. Masks the characters in the source string that you specify. - FALSE. Masks random characters in the source string. Default is FALSE.
useTargetFilter	Specifies the characters to use in the masked string. You can specify the following values: - TRUE. Uses characters that you specify in the masked string. - FALSE. Uses random characters in the masked string. Default is FALSE.

services

Use the services resource to run industry data services using Informatica Intelligent Cloud Services REST API.

Industry data services refer to a collection of pre-built data services designed to parse, validate, and serialize industry-specific data standards. These data services are available on the data service repository and you can use REST APIs to run them.

When you use the services resource, use the following request header format:

```
<METHOD><base URL>
Content-Type: application/json
Accept: application/json
IDS-SESSION-ID: <SessionId>
```

POST request

To run a data service, use the following URI:

```
/DSRepo/rest/api/v1/services/run/<service name>
```

Include the following fields in the request:

Field	Type	Required	Description
agentGroupId	String	Yes	ID of the Secure agent group to process the request.
input			Default input port for the data service.
type	string	Yes	Whether the input is a file or data. Use one of the following values: - FILE: Input is a file. - BUFFER: Input is data.
value	string	Yes	Value of the input type. - If the type is FILE, value of the input file path. - If the type is BUFFER, the value is a text string.
output			Default output port for the data service.
type	String	Yes	Whether the output is provided as data or saved to a target file path. Use one of the following values: - FILE: Output is saved to a target file. - BUFFER: Output is provided data.
value	String	Required when type is FILE	Value of the output file path.
additionalInputs	Array		Additional input ports for the data service.
name	String		Name of the additional input port.
type	String		Whether the additional input is a file or data. Use one of the following values: - FILE: Input is a file. - BUFFER: Input is data.

Field	Type	Required	Description
value	String		Value of the additional input type. - If the type is FILE, the value of the input file path. - If the type is BUFFER, the value is a text string.
additionalOutputs	Array	Required for some data services.	Additional output ports for the data service. For HL7, provide the following additional outputs: - Errors - ErrorsFound . For HIPPA, provide the following additional outputs: - Errors - ErrorsFound - ValidationReport
name	String	Yes	Name of the additional output port.
type	String	Yes	Whether the additional output is provided data or saved to a target file path. Use one of the following values: - FILE: Output is saved to a target file. - BUFFER: Output is provided as data.
value	String	Required when type is FILE	Value of the additional output file path.
serviceParameters	Array		Variables in the service parameter ports.
name	String		Name of the service parameter.
value	String		Value of the service parameter.

POST request example

To run an HL7 data service, you might send a request similar to the following example:

```
POST <serverUrl>/DSRepo/rest/api/v1/services/run/HL7_2_6_ADT_A01_Parser
Content-Type: application/json
Accept:application/json
IDS-SESSION-ID:2l0oeVx22Rujiej7yTokmT
{
    "agentGroupId": "010BM02500000000000K",
    "input": {
        "type": "FILE",
        "value": "/root/TEST/v25_s2.txt"
    },
    "output": {
        "type": "FILE",
        "value":"/root/TEST/"
    },
    "additionalInputs": [],
    "serviceParameters": [],
    "additionalOutputs": [
        {
            "name": "Errors",
            "type": "FILE",
            "value":"/root/TEST/"
        },
        {
            "name": "ErrorsFound",
            "type": "FILE",
```

```
                "value":"/root/TEST/"
            }
        ]
    }
```

POST response

When you use a POST request to run a data service, it returns a success response if successful or an error object if an error occurs. If the request is unsuccessful, the response includes a reason for the failure.

POST response example

If the request is successful, you might receive a response similar to the following example:

```
{
    "output": "/root/TEST/HL7_2_6_ADT_A01_Parser_output.xml",
    "additionalOutputs": [
    {
            "type": "file",
            "value": "/root/TEST/HL7_2_6_ADT_A01_Parser_Errors.xml",
            "name": "Errors"
        },
        {
            "type": "file",
            "value": "/root/TEST/HL7_2_6_ADT_A01_Parser_ErrorsFound.xml",
            "name": "ErrorsFound"
        }
    ],
    "message": "Success"
}
```

Taskflows

You can use the REST API to get the status of a taskflow and publish multiple taskflows simultaneously.

For more information about how to run a taskflow as an API, see Running a taskflow as an API.

Monitoring taskflow status with the status resource

If you have the privilege to view job results in Monitor, you can use the status resource to get the status of a taskflow. You can get the status of a taskflow using the taskflow run ID as a path parameter or using query parameters such as run ID, run status, start time, end time, offset, and row limit.

GET request

To get the status of a taskflow using the run ID as a path parameter, use the following URI:

```
<Informatica Intelligent Cloud Services URL>/active-bpel/services/tf/status/<run ID>
```

For example:

```
https://na4.dm.us.informaticacloud.com/active-bpel/services/tf/status/
20262247166322413568
```

You can also get the status of multiple taskflows using query parameters. The query parameters are case sensitive.

You can use the following optional query parameters in the GET request URI:

Field	Description
runId	Run ID for the taskflow. For example, to get the status of a particular taskflow with a run ID, use the following URI: `https://na4.dm.us.informaticacloud.com/active-bpel/services/tf/status?runId=20262247166322413568`
runStatus	Execution status of the taskflow. You can specify the status as Success, Failed, Suspended, or Running. For example, to get the status of all the taskflows that were successfully executed in the last 24 hours, use the following URI: `https://na4.dm.us.informaticacloud.com/active-bpel/services/tf/status?runStatus=Success`
startTime	Start time for the taskflow runs at the beginning of the date and time range. Use Coordinated Universal Time (UTC). For example, to get the status of all the taskflows that started on or after 2021-06-**10**T05:48:28Z, use the following URI: `https://na4.dm.us.informaticacloud.com/active-bpel/services/tf/status?startTime=2021-06-`**10**`T05:48:28Z`
endTime	Start time for the taskflow runs at the end of the date and time range. Use Coordinated Universal Time (UTC). For example, to get the status of all the taskflows that started between 2021-06-**10**T05:48:28Z and 2021-06-**11**T05:48:28Z, use the following URI: `https://na4.dm.us.informaticacloud.com/active-bpel/services/tf/status?startTime=2021-06-10T05:48:28Z&endTime=2021-06-`**11**`T05:48:28Z`
offset	Number of rows to skip. For example, you might want to skip the first three rows.
rowLimit	Maximum number of rows to return. The maximum number you can specify is 50. If you omit this parameter, the query returns all available rows, up to a maximum of 10 rows.

You can use any combination of these query parameters to get the status of multiple taskflows. For example, you can use the following URI:

```
<Informatica Intelligent Cloud Services URL>/active-bpel/services/tf/status?
startTime=<startTime>&runStatus=<runStatus>&endTime=<endTime>&rowLimit=<rowLimit>
```

Note: If the startTime or endTime parameters are not used in the query, the response contains status information about the taskflows that ran in the last 24 hours.

Authenticate the GET request in one of the following ways:

- Use basic authorization and specify the Informatica Intelligent Cloud Services user name and password. For example:

```
GET <Informatica Intelligent Cloud Services URL>/active-bpel/services/tf/status/<run
ID>
Accept: application/json
Authorization: Basic Auth
username: <Informatica Intelligent Cloud Services user name>
password: <Informatica Intelligent Cloud Services password>
```

- Use the INFA-SESSION-ID in the HTTP header.

For example:

```
GET <Informatica Intelligent Cloud Services URL>/active-bpel/services/tf/status/<run
ID>
Accept: application/json
INFA-SESSION-ID: <sessionId>
```

To get the INFA-SESSION-ID, use the Platform REST API version 3 login resource. For more information about the login resource, see *REST API Reference*.

Send the request using JSON format. Include the following line in the header: `Accept: application/json`

GET response

Returns the taskflow status information if successful or an error object if errors occur.

If successful, returns the following status information for a taskflow:

Field	Type	Description
assetName	String	Name of the taskflow. The taskflow name also includes the custom name, if you had added a custom name to the taskflow using an API or the RunAJob utility.
assetType	String	Type of the object. Returns the value TASKFLOW.
duration	String	Time in seconds that the taskflow ran before it completed, was suspended, was failed, or was stopped.
endTime	Date/time	End time for the taskflow run. Uses Coordinated Universal Time (UTC).
location	String	Project and folder path where the taskflow is located.
runId	Long	Run ID for the taskflow.
runtimeEnv	String	ID of the runtime environment where the taskflow runs.
runtimeEnvName	String	Name of the runtime environment where the taskflow runs.
startTime	Date/time	Start time for the taskflow run. Uses Coordinated Universal Time (UTC).
startedBy	String	User who started the taskflow.
status	String	Execution status of the taskflow. Returns one of the following values to indicate the taskflow status: - RUNNING. The taskflow is running. - SUCCESS. The taskflow completed successfully. - FAILED. The taskflow did not complete because it encountered errors. - SUSPENDED. The taskflow run was suspended.
subtasks	String	Number of subtasks that the taskflow contains.
updateTime	Date/time	Last time the taskflow run status was updated. Uses Coordinated Universal Time (UTC).
errorMessage	String	Error message string.
subtaskDetails	String	Object that contains status details for all subtasks in the taskflow.

Field	Type	Description
details	String	Status details. Includes status information for each subtask in the tasks object.
tasks	Collection	Status information for all subtasks that the taskflow contains.

The tasks object includes the following status information for each subtask that the taskflow contains:

Field	Type	Description
assetName	String	Name of the subtask in the taskflow.
assetType	String	Type of the subtask. Returns one of the following values: - MTT. Mapping task. - DSS. Synchronization task.
duration	String	Time in seconds that the subtask ran before it completed, was failed, or was stopped.
endTime	Date/time	End time for the subtask run. Uses Coordinated Universal Time (UTC).
errorMessage	String	Error message string.
errorRows	String	Total number of rows that resulted in errors in a subtask.
location	String	Project and folder path where the subtask is located.
rowsProcessed	String	Total number of rows that were processed in a subtask.
runId	Long	Run ID for the subtask.
runtimeEnv	String	ID of the runtime environment where the subtask runs.
runtimeEnvName	String	Name of the runtime environment where the subtask runs.
startTime	Date/time	Start time for the subtask run. Uses Coordinated Universal Time (UTC).
startedBy	String	User who started the task. This field is the same as the user who started the taskflow.
status	String	Execution status of the subtask. Returns one of the following values to indicate the subtask status: - QUEUED. The subtask is queued on a Secure Agent, but it has not started yet. - STARTING. The subtask is starting. - RUNNING. The subtask is running. - COMPLETED. The subtask completed successfully. - SUSPENDED. The subtaskflow is suspended. - STOPPED. The taskflow has stopped running, so the subtask cannot start. - WARNING. The subtask completed with errors. - FAILED. The subtask did not complete because it encountered errors.
subtasks	String	Reserved for future use. When this field is returned for a subtask, the value is always 0.

Field	Type	Description
successRows	String	Total number of rows that were processed successfully in a subtask.
updateTime	Date/time	Last time the subtask run status was updated. Uses Coordinated Universal Time (UTC).

You might receive one of the following responses:

Response	Description
Purged logs	If logs are purged for the instance, the response is as follows: `{ "status": "No status available." }` The HTTP status code is **200 OK**.

Invalid run ID	If the run ID is not valid, the response is as follows:
	
	`{`
	`    "error": "CMN_003-Bad request. Error message - The property`
	`'<runID>', used in a query expression, is not defined in type 'OData.job-`
	`log-service.JobLogEntry'."`
	`}`
	
	The HTTP status code is **400 Bad Request(From JLS)**.

Invalid filter clause	If the parameter value is not valid, the response is as follows:
	
	`{`
	`    "error": "JLS_007-Invalid Filter clause in OData request. RawURI =`
	`http://internal-na1-elb.infacloudops.net:443/jls-di/internal/api/v1"`
	`}`
	
	The HTTP status code is **400 Bad Request(From JLS)**.

Unavailable JLS service	If the JLS service is unavailable, the response is as follows: `{ "error": "503-Service Unavailable." }` The HTTP status code is **503 Service Unavailable**.

GET example using the run ID as a path parameter

The following example shows a taskflow status request that uses the run ID as a path parameter:

```
GET https://pod.ics.dev:444/active-bpel/services/tf/status/20262247166322413568
Accept: application/json
INFA-SESSION-ID: 9KA11tLGqxVcGeul8SQBK3
```

Based on the taskflow configuration and request inputs, the response can be of the following types:

Taskflow without subtasks

If the request is successful and the taskflow does not contain subtasks, the response includes taskflow status information as shown in the following example:

```
{
    "assetName": "Taskflow1",
    "assetType": "TASKFLOW",
    "duration": "2",
    "endTime": "2018-12-25T15:56:39Z",
    "location": "Default",
```

```
            "runId": "262247166322413568",
            "runtimeEnv": "tf_runtime",
            "runtimeEnvName": "",
            "startTime": "2018-12-25T15:56:37Z",
            "startedBy": "sb",
            "status": "SUCCESS",
            "subtasks": "0",
            "updateTime": "2018-12-25T15:56:39Z",
            "errorMessage": {},
            "subtaskDetails": {
                "details": {}
            }
    }
```

The HTTP status code is **200 OK**.

Taskflow with subtasks

If the request is successful and the taskflow contains multiple subtasks, the response includes status
information for each subtask that the taskflow contains as shown in the following example:

```
{
        "assetName": "Taskflow2",
        "assetType": "TASKFLOW",
        "duration": "89",
        "endTime": "2018-12-23T17:25:16Z",
        "location": "Default",
        "runId": 20262247166322413568,
        "runtimeEnv": "tf_runtime",
        "runtimeEnvName": "",
        "startTime": "2018-12-23T17:23:47Z",
        "startedBy": "sb",
        "status": "SUCCESS",
        "subtasks": "2",
        "updateTime": "2018-12-23T17:25:17Z",
        "errorMessage": {},
        "subtaskDetails": {
            "details": {
                "tasks": [
                    {
                        "assetName": "MTR",
                        "assetType": "MTT",
                        "duration": "3",
                        "endTime": "2018-12-23T17:24:45Z",
                        "errorMessage": "",
                        "errorRows": "0",
                        "location": "Default",
                        "rowsProcessed": "7",
                        "runId": "4",
                        "runtimeEnv": "01001Q25000000000002",
                        "runtimeEnvName": "tf_runtime_devagent",
                        "startTime": "2018-12-23T17:24:42Z",
                        "startedBy": "sb",
                        "status": "COMPLETED",
                        "subtasks": "0",
                        "successRows": "7",
                        "updateTime": "2018-12-23T17:24:46Z"
                    },
                    {
                        "assetName": "MTR",
                        "assetType": "MTT",
                        "duration": "10",
                        "endTime": "2018-12-23T17:23:59Z",
                        "errorMessage": "",
                        "errorRows": "0",
                        "location": "Default",
                        "rowsProcessed": "7",
                        "runId": "3",
                        "runtimeEnv": "01001Q25000000000002",
```

```
                        "runtimeEnvName": "tf_runtime_devagent",
                        "startTime": "2018-12-23T17:23:49Z",
                        "startedBy": "sb",
                        "status": "COMPLETED",
                        "subtasks": "0",
                        "successRows": "7",
                        "updateTime": "2018-12-23T17:24:00Z"
                    }
                ]
            }
        }
    }
```

The HTTP status code is **200 OK**.

GET example using query parameters

The following example shows a taskflow status request that uses run ID, run status, and row limit as query parameters:

```
GET https://pod.ics.dev:444/active-bpel/services/tf/status?
runId=20262247166322413568&runStatus=Success&rowLimit=3
Accept: application/json
INFA-SESSION-ID: 9KA11tLGqxVcGeul8SQBK3
```

If the request is successful, the response format is the same as when we use the path parameter, but within square brackets [ ].

If the request is successful and the taskflow does not contain subtasks, the response includes taskflow status information as shown in the following example:

```
[
    {
        "assetName": "Taskflow1",
        "assetType": "TASKFLOW",
        "duration": "2",
        "endTime": "2018-12-25T15:56:39Z",
        "location": "Default",
        "runId": 262247166322413568,
        "runtimeEnv": "tf_runtime",
        "runtimeEnvName": "",
        "startTime": "2018-12-25T15:56:37Z",
        "startedBy": "sb",
        "status": "SUCCESS",
        "subtasks": "0",
        "updateTime": "2018-12-25T15:56:39Z",
        "errorMessage": {},
        "subtaskDetails": {
            "details": {
                "tasks": []
            }
        }
    }
]
```

The HTTP status code is **200 OK**.

GET example without any parameter

The following example shows a taskflow status request without a path parameter or query parameter:

```
<Informatica Intelligent Cloud Services URL>/active-bpel/services/tf/status
```

The response contains status information of the last 10 taskflows that were run in the last 24 hours.

GET example of a running taskflow

If you use the status resource to get the status of a running taskflow, the response includes taskflow status information with the `endTime` as `null` without quotes as shown in the following example:

```
{
    "assetName": "waitStatus",
    "assetType": "TASKFLOW",
    "duration": 27,
    "endTime": null,
    "errorMessage": "",
    "location": "Default",
    "runId": 737194191850250240,
    "runtimeEnv": "taskflow-preview-usw1-r40-app02.infacloudops.net:4430",
    "runtimeEnvName": "",
    "startedBy": "sb",
    "startTime": "2022-07-28T06:26:32Z",
    "status": "RUNNING",
    "subtasks": 0,
    "updateTime": "2022-07-28T06:26:32Z",
    "subtaskDetails": {
        "details": {
            "tasks": []
        }
    }
}
```

After the taskflow is complete, the correct `endTime` value is displayed.

Publishing taskflows in bulk

You can use the publish resource to publish a single taskflow or multiple taskflows simultaneously and save time.

The taskflows are published in the same order as given in the request payload. You can publish a maximum of 199 taskflows at a time.

1. In a REST client, use a POST request with the following URL:

    ```
    <Informatica Intelligent Cloud Services URL>/active-bpel/asset/v1/publish
    ```

    For example: `https://na1.dm-us.informaticacloud.com/active-bpel/asset/v1/publish`

2. Add the following headers:

Key	Value
Accept	application/vnd.api+json
Content-Type	application/vnd.api+json
INFA-SESSION-ID	Use the login resource to get the session ID. For more information about the login resource, see *REST API Reference*.

3. In the body, use the assetPaths attribute to specify one or more locations and names of the taskflows that you want to publish.

    Use the following format:

    ```
    {
      "data": {
        "type": "publish",
        "attributes": {
    ```

```
            "assetPaths": [
                        "Explore/<location-of-taskflow1>/<name-of-
    taskflow1>.TASKFLOW.xml",
                        "Explore/<location of taskflow2>/<name-of-
    taskflow2>.TASKFLOW.xml",
                        "Explore/<location-of-taskflown>/<name-of-
    taskflown>.TASKFLOW.xml"
                        ]
        }
    }
}
```

4. Send the POST request.

    You see a publish ID and a success or failure response. If the request fails, the response also gives the error details.

    The following snippet shows a sample response:

```
{
    "data": {
        "type": "publish",
        "id": "690487059198201856",
        "attributes": {
            "jobState": "NOT_STARTED",
            "jobStatusDetail": {},
            "startedBy": "autouser_pod1",
            "startDate": "2022-03-21T09:09:04.000+0000",
            "totalCount": 1,
            "processedCount": 0,
            "assetPaths": [
                "Explore/Pavan/BulkPublishApi/BPTaskflow1.TASKFLOW.xml"
            ]
        }
    },
    "links": {
        "self": https://na1.dm-us.informaticacloud.com/active-bpel/asset/v1/
    publish/690487059198201856,
        "status": https://na1.dm-us.informaticacloud.com/active-bpel/asset/v1/
    publish/690487059198201856/Status
        }
    }
}
```

    The publish ID in this example is 690487059198201856.

5. To view information about the publish status and publish job, use a GET request with the following URLs:

URL	Description
`<Informatica Intelligent Cloud Services URL>/active-bpel/asset/v1/publish/<publishId>/Status`	Displays the publish status.
`<Informatica Intelligent Cloud Services URL>/active-bpel/asset/v1/publish/<publishId>`	Displays the publish job information.

Unpublishing taskflows in bulk

You can use the unpublish resource to unpublish a single taskflow or multiple taskflows simultaneously and save time.

The taskflows are unpublished in the same order as given in the request payload. You can unpublish a maximum of 199 taskflows at a time.

1. In a REST client, use a POST request with the following URL:

   ```
   <Informatica Intelligent Cloud Services URL>/active-bpel/asset/v1/unpublish
   ```

   For example: `https://na1.dm-us.informaticacloud.com/active-bpel/asset/v1/unpublish`

2. Add the following headers:

Key	Value
Accept	application/vnd.api+json
Content-Type	application/vnd.api+json
INFA-SESSION-ID	Use the login resource to get the session ID. For more information about the login resource, see *REST API Reference*.

3. In the body, use the assetPaths attribute to specify one or more locations and names of the taskflows that you want to unpublish.

   Use the following format:

   ```
   {
     "data": {
       "type": "unpublish",
       "attributes": {
         "assetPaths": [
                         "Explore/<location-of-taskflow1>/<name-of-
   taskflow1>.TASKFLOW.xml",
                         "Explore/<location of taskflow2>/<name-of-
   taskflow2>.TASKFLOW.xml",
                         "Explore/<location-of-taskflown>/<name-of-
   taskflown>.TASKFLOW.xml"
                        ]
       }
     }
   }
   ```

4. Send the POST request.

   You see an unpublish job ID and a success or failure response. If the request fails, the response also gives the error details.

   The following snippet shows a sample response:

   ```
   {
       "data": {
           "type": "unpublish",
           "id": "7645874567965431",
           "attributes": {
               "jobState": "NOT_STARTED",
               "jobStatusDetail": {},
               "startedBy": "autouser_pod1",
               "startDate": "2022-03-21T09:09:04.000+0000",
               "totalCount": 1,
               "processedCount": 0,
               "assetPaths": [
                   "Explore/Pavan/BulkUnpublishApi/BPTaskflow1.TASKFLOW.xml"
   ```

```
                ]
            }
        },
        "links": {
            "self": https://na1.dm-us.informaticacloud.com/active-bpel/asset/v1/
    unpublish/7645874567965431,
            "status": https://na1.dm-us.informaticacloud.com/active-bpel/asset/v1/
    unpublish/7645874567965431/Status
        }
    }
```

The unpublish job ID in this example is 7645874567965431.

5. To view information about the unpublish status and unpublish job, use a GET request with the following URLs:

URL	Description
`<Informatica Intelligent Cloud Services URL>/active-bpel/asset/v1/unpublish/<unpublishId>/Status`	Displays the unpublish status.
`<Informatica Intelligent Cloud Services URL>/active-bpel/asset/v1/unpublish/<unpublishId>`	Displays the unpublish job information.

workflow

Use this resource to request the details of a linear taskflow or the details of all linear taskflows in the organization. You can also create, update, or delete a linear taskflow.

GET request

To request the details of a particular linear taskflow, include the linear taskflow ID or linear taskflow name in the URI. Use one of the following URIs:

`/api/v2/workflow/<id>`

`/api/v2/workflow/name/<name>`

If you use the linear taskflow name in the URI and the linear taskflow name includes a space, replace the space with `%20`. For example:

`/api/v2/workflow/name/my%20linear%20taskflow`

To request the details of all linear taskflows in the organization, use the following URI:

`/api/v2/workflow`

Optionally, you can receive the response in simple mode which significantly improves performance. When you enable simple mode, the response does not include the ScheduleId attribute and the email attributes. To receive the response in simple mode, include `simpleMode=true` in the request. Use the following URI to receive details of all linear taskflows using simple mode:

`/api/v2/workflow/?simpleMode=true`

GET response

If successful, returns the workflow object for the requested linear taskflow. Or, if you request the details for all linear taskflows in the organization, returns a workflow object for each linear taskflow in the organization.

Returns an error object if errors occurred.

The workflow object includes the following attributes:

Field	Type	Description
id	String	Linear taskflow ID.
orgId	String	Organization ID.
name	String	Linear taskflow name.
description	String	Description.
createTime	Date/time	Time the linear taskflow was created.
updateTime	Date/time	Last time the linear taskflow was updated.
createdBy	String	User who created the linear taskflow.
updatedBy	String	User who last updated the linear taskflow.
errorTaskEmail		Object that includes the taskEmail object for error notifications.
id	String	Included in taskEmail object for errorTaskEmail. ID.
emails	String	Included in taskEmail object for errorTaskEmail. Email address that receives email notification when a task fails to complete.
successTaskEmail		Object that includes the taskEmail object for success notifications.
id	String	Included in taskEmail object for successTaskEmail. ID.
emails	String	Included in taskEmail object for successTaskEmail. Email address that receives email notification when a task completes successfully.
warningTaskEmail		Object that includes the taskEmail object for warning notifications.
id	String	Included in taskEmail object for warningTaskEmail. ID.
emails	String	Included in taskEmail object for warningTaskEmail. Email address that receives email notification when a task completes with errors.
agentId	String	Agent that runs the task.
runtimeEnvironmentId	String	Runtime environment used for the task.
scheduleId	String	Schedule associated with the linear taskflow, if any.
preProcessingCmd	String	Command to run before the task.
postProcessingCmd	String	Command to run after the task completes.

Field	Type	Description
tasks		Defines each task associated with the linear taskflow. Includes a workflowTask object for each task.
taskId	String	Included in the workflowTask object. Task ID.
type	String	Included in the workflowTask object. Workflow task type. Returns one of the following codes: - DMASK. Masking task. - DRS. Replication task. - DSS. Synchronization task. - MTT. Mapping task. - PCS. PowerCenter task.
name	String	Included in the workflowTask object. Task name.
stopOnError	Boolean	Included in the workflowTask object. Stops the linear taskflow if a task fails to complete.
stopOnWarning	Boolean	Included in the workflowTask object. Stops the linear taskflow if a task completes with warnings.

POST request

To create a linear taskflow, use the following URI:

```
/api/v2/workflow
```

If you want to specify a location for the linear taskflow, include the container ID in the request. If the container ID isn't included in the request, the linear taskflow is created in the Default folder. You can find the container ID for a project or folder in the Data Integration user interface. On the **Explore** page, select the folder. In the URL, the last string of characters is the container ID.

For example, in the following URL, the container ID is dH2DuGJYda7ijgW4Sm32sR

```
https://na1.dm-us.informaticacloud.com/diUI/products/integrationDesign/main/Explore/
dH2DuGJYda7ijgW4Sm32sR
```

To update a linear taskflow, include the workflow ID as shown in the following example:

```
/api/v2/workflow/<id>
```

When you update a linear taskflow, Data Integration replaces the existing linear taskflow with the update.

You can submit a partial update using partial mode. If you want to update a field in the workflowTask object using partial mode, you must include the taskId field. To submit a request using partial mode, use a JSON request and include the following line in the header:

```
Update-Mode=PARTIAL
```

With this URI, you can use the following attributes in the **workflow** object:

Field	Type	Required	Description
name	String	Yes	Name of the linear taskflow.
description	String		Description of the linear taskflow.
containerId	String		ID of the project or folder to contain the linear taskflow. If not included in the request, the linear taskflow is created in the Default folder.
errorTaskEmail			Object that includes the taskEmail object for error notifications.
id	String		Include in taskEmail object for errorTaskEmail. ID.
emails	String		Include in taskEmail object for errorTaskEmail. Email address that receives email notification when a task fails to complete.
successTaskEmail			Object that includes the taskEmail object for success notifications.
id	String		Include in taskEmail object for successTaskEmail. ID.
emails	String		Include in taskEmail object for successTaskEmail. Email address that receives email notification when a task completes successfully.
warningTaskEmail			Object that includes the taskEmail object for warning notifications.
id	String		Include in taskEmail object for warningTaskEmail. ID.
emails	String		Include in taskEmail object for warningTaskEmail. Email address that receives email notification when a task completes with errors.
tasks			Use a workflowTask object to define the following attributes for each task you want to include in the linear taskflow.
taskId	String	Yes	Include in the workflowTask object. Task ID.
		Yes	Include in the workflowTask object. Workflow task type. Use one of the following codes: - DMASK. Masking task. - DRS. Replication task. - DSS. Synchronization task. - MTT. Mapping task. - PCS. PowerCenter task.
name	String	Yes	Include in the workflowTask object. Name of the task.

Field	Type	Required	Description
stopOnError	Boolean		Include in the workflowTask object. Stops the linear taskflow if the task fails to complete. Use one of the following options: - 1. True. Stop on error. - 2. False. Do not stop on error.
stopOnWarning	Boolean		Include in the workflowTask object. Stops the linear taskflow if a task completes with warnings. Use one of the following options: - 1. True. Stop on error. - 2. False. Do not stop on error.
scheduleId	String		Schedule for the linear taskflow.

POST response

If successful, returns the workflow response object for the linear taskflow that you created or updated.

Returns the error object if errors occur.

DELETE request

To delete a linear taskflow, use the linear taskflow ID in the following URI:

```
/api/v2/workflow/<id>
```

DELETE response

Returns the 200 response code if the request is successful.

Returns the error object if errors occur.

POST example

To update an existing linear taskflow with an ID of 0000342J0000K, you might use the following request:

```
POST <serverUrl>/api/v2/workflow/0000342J0000K
Content-Type: application/json
Accept: application/json
icSessionId: <icSessionId>

{
"@type": "workflow",
"name": "linear taskflow",
"tasks":[{
"@type":"workflowTask",
"taskId":"0000100I00000000001G",
"type":"DSS",
"name":"DSS_DQ5",
"stopOnError":"false"
},{
"@type":"workflowTask",
"taskId":"0000100Z0000000000B8",
"type":"MTT",
"name":"CIT_SimpleTemplate2",
"stopOnError":"false"
},{
"@type":"workflowTask",
"taskId":"0000100G000000000002",
"type":"DRS",
"name":"SF2File",
"stopOnError":"false"
}]
}
```

A successful request returns the workflow object that you updated.

Data Integration REST API supplemental information

This section includes supplemental information such as connector data types and a mapping of connection REST API attributes to user interface fields.

Connector data types

When you submit a request for connector metadata, data type is included in the response. Data types for connector attributes are returned in REST API responses using a numeric value.

The following example shows a response with the type value of 2:

```
{
  "name": "database",
  "label": "",
  "id": "",
  "value": "",
  "type": 2,
  "isMandatory": true,
  "visible": false,
  "list": []
},
```

The type value of 2 means the database attribute can only contain alphabetic characters.

The following table lists the numeric values that might be included in the response and the corresponding data type:

Value	Data Type	Description
1	NUMERIC_TYPE	Attribute value can only contain numbers.
2	ALPHABET_TYPE	Attribute value can only contain alphabetic characters.
3	NUMERIC_TYPE/ ALPHABET_TYPE	Attribute value can only contain numbers or alphabetic characters.
4	SYMBOLS_TYPE	Attribute value can only contain symbols.
5	NUMERIC_TYPE/ SYMBOLS_TYPE	Attribute value can only contain numbers and symbols.
6	ALPHABET_TYPE/ SYMBOLS_TYPE	Attribute value can only contain alphabetic characters and symbols.
7	NUMERIC_TYPE/ ALPHABET_TYPE/ SYMBOLS_TYPE	Attribute value can only contain alphabetic characters, numbers, and symbols.
8	LIST_TYPE	Attribute value can only contain values from a predefined list.
9	NUMERIC_TYPE/ LIST_TYPE	Attribute value can only contain values from a predefined list and the value contains only numbers.

Value	Data Type	Description
10	ALPHABET_TYPE/ LIST_TYPE	Attribute value can only contain values from a predefined list and the value contains only numbers.
11	NUMERIC_TYPE/ ALPHABET_TYPE/ LIST_TYPE	Attribute value can only contain values from a predefined list and the value contains only alphabetic characters and numbers.
12	SYMBOLS_TYPE/ LIST_TYPE	Attribute value can only contain values from a predefined list and the value contains only symbols.
13	NUMERIC_TYPE/ SYMBOLS_TYPE/ LIST_TYPE	Attribute value can only contain values from a predefined list and the value contains only numbers and symbols.
14	ALPHABET_TYPE/ SYMBOLS_TYPE/ LIST_TYPE	Attribute value can only contain values from a predefined list and the value contains only alphabetic characters and symbols.
15	NUMERIC_TYPE/ ALPHABET_TYPE/ SYMBOLS_TYPE/ LIST_TYPE	Attribute value can only contain values from a predefined list and the value contains only alphabetic characters, numbers, and symbols.
16	BOOLEAN	Attribute value is a boolean.
32	PASSWORD	Attribute value is a password.

For more information about requesting connector metadata, see .

Connection user interface fields to REST API attributes mapping

Some connection field names in the user interface do not intuitively map to corresponding REST API attribute names in the connection resource. Additionally, some attribute names used for REST API GET and POST methods for the connection resource do not match the attribute names used in the REST API response that populates the values shown in the user interface.

The following tables map user interface fields with attributes used for REST API GET and POST calls and the REST API response to the user interface, where the correlation between these fields might be confusing.

Connection	UI Field Name	REST API GET and POST Attribute Name	Response to UI Attribute Name
All connections	Runtime Environment	runtimeEnvironmentId	agentGroupId
CSV Flat File	Directory	database	dirName
FTP and SFTP	Directory	database	dirName
Microsoft Access	Data Source Name	database	database
Microsoft SQL Server	SQL Server Version	type	subType
Oracle	Service Name	database	database

Connection	UI Field Name	REST API GET and POST Attribute Name	Response to UI Attribute Name
SAP IDoc Reader	Destination Entry	database	database
SAP IDoc Writer and SAP RFC/ BAPI	Connection String	database	database
Web Service Consumer	Endpoint URL	serviceUrl	serviceUrl

CHAPTER 5

Mass Ingestion Files REST API

Use the file ingestion resources to run and monitor file ingestion tasks.

When you use file ingestion resources, note the following rules:

- Use JSON format.
- Use the following base URL:

  ```
  <serverUrl>/mftsaas/api/v1/<API name>
  ```

- Use the following request header format:

  ```
  <METHOD> <serverUrl>/<URI> HTTP/<HTTP version>
  Content-Type: application/json
  Accept: application/json
  IDS-SESSION-ID: <SessionId>
  ```

Note: If you use a tool such as Postman that automatically includes the HTTP version, do not enter the HTTP version in the URL. If the HTTP version appears twice in the URL, the request fails.

job resource

Use the job resource to start a file ingestion job. You can also use the job resource to retrieve job status or job logs for a file ingestion task. Use the file ingestion REST API version 1 task resource to retrieve the ID and name of the task.

RUN Request

To start a file ingestion task job, use the following URI:

```
mftsaas/api/v1/job
```

Include the following information in the request:

Field	Type	Required	Description
taskId	String	Yes	File ingestion ID.
taskName	String	-	File ingestion name.

437

Use the following source directory and target directory keys for the specified connectors when you start a file ingestion job:

Connector	srcDir	tgtDir
local	sourceDirectory	targetDirectory
ftp,ftps,sftp	sourceDirectory	targetDirectory
gcs	sourceDirectory	gcsTargetLocation
hdfs	sourceDirectory	hdfsTargetLocation
adlsgen2	sourceDirectory	adlsGen2TargetLocation
s3	s3SourceLocation	s3TargetLocation
blob	blobSourceLocation	blobContainer

You can overwrite the following parameters using the job resource REST API:

Category	Parameter	ID
General	Source Connection	sourceConnection
General	Target Connection	targetConnection
General	Parallel Batch Log Level	parallelBatch
General	Log Level	logLevel
Source	Source Directory	sourceDirectory
Source	File Pattern	filePattern
Source	Batch Size	batchSize
Source	Include files from sub-folders	includeSubfolder
Source	Skip Duplicate files	checkDuplicate
Source	Check File Stability	fileStability
Source	Stability Check Interval	stabilityCheckInterval
Target	Target Directory	targetDirectory

Note: You must pass the connection ID to overwrite the source and target connection parameters.

Use the following sample as a reference to start a file ingestion task job:

```
{
        "taskId": "k1YHA1blhcBjbJvCIRQX2s",
        "taskName": "localtolocal_param2"
}
```

Use the following sample request to overwrite the source option values that were passed in the user interface:

```
"variables": [{
        "variable": "<string>",
        "value": "<string>"
    }]
```

In the following example, the parameter value that were passed in the user interface is overwritten to corresponding values provided in JSON POST while using the job resource REST API:

```
{
    "taskId": "0efdVUEZeV2cB0quomeksd",
    "taskName": "localtolocal_param2",
    "parameters": {
        "category": [{

                "id": "General",

                "parameter": [

                    {

                        "id":"sourceConnection",

                        "value":"AdvancedSFTPV2"

                    },

                    {

                        "id":"targetConnection",

                        "value":"AdvancedSFTPV2"

                    },

                    {

                        "id":"parallelBatch",

                        "value":"10"

                    },

                    {

                        "id":"logLevel",

                        "value":"DEBUG"

                    }

                ]

            },{

                "id": "Source",

                "parameter": [{

                        "id": "sourceDirectory",

                        "value": "/root/test1"

                    },

                    {

                        "id":"filePatternType",
```

```
                    "value":"reg"
                },
                {
                    "id": "filePattern",
                    "value": "*.txt"
                },
                {
                    "id": "batchSize",
                    "value": "5"
                },
                {
                    "id":"includeSubfolder",
                    "value":"true"
                },
                {
                    "id":"checkDuplicate",
                    "value":"true"
                },
                {
                    "id":"fileStability",
                    "value":"true"
                },
                {
                    "id":"stabilityCheckInterval",
                    "value":"30"
                }
            ]
        },
        {
            "id": "Target",
            "parameter": [{
                "id": "targetDirectory",
                "value": "/root/test2"
            }]
        }
```

```
        ]

      }

   }
```

The following example shows to override a file ingestion task with filename as a variable:

```
{
   "taskId": "4m24k3UFWMkkqd55YDefIB",
   "taskName": "R41_Local_Local",
   "parameters": {
     "category": [
       {
         "id": "Source",
         "parameter": [
           {
             "id": "sourceDirectory",
             "value": "/${Parentfolder}"
           },
           {
             "id": "filePickupFilePath",
             "value": "${filename}"
           },
           {
             "id": "batchSize",
             "value": "5"
           }
         ]
       },
       {
         "id": "Target",
         "parameter": [
           {
             "id": "targetDirectory",
             "value": "/${Parentfolder}/Target"
           }
         ]
       }
     ]
   },
   "variables": [
     {
       "variable": "Parentfolder",
       "value": "root/Arun"
     },
     {
       "variable": "filename",
       "value": "filepath.txt"
     }
   ]
}
```

The following exampleshows to override a file ingestion task with filelist as a variable:

```
{
   "taskId": "4m24k3UFWMkkqd55YDefIB",
   "taskName": "R41_Local_Local",
   "parameters": {
     "category": [
       {
         "id": "Source",
         "parameter": [
           {
             "id": "sourceDirectory",
             "value": "/${Parentfolder}"
           },
           {
             "id": "filePickupFileList",
             "value": "${filelist}"
           },
           {
```

```
            "id": "batchSize",
            "value": "5"
          }
        ]
      },
      {
        "id": "Target",
        "parameter": [
          {
            "id": "targetDirectory",
            "value": "/${Parentfolder}/Target"
          }
        ]
      }
    ]
  },
  "variables": [
    {
      "variable": "Parentfolder",
      "value": "root/Arun"
    },
    {
      "variable": "filelist",
      "value": "File1.txt,File2.txt,File3.txt,File4.txt"
    }
  ]
}
```

RUN Response

If successful, file ingestion returns the run ID for the job. Use the run ID to monitor the job status and request log files for the job.

If unsuccessful, the response includes a reason for the failure.

GET Status Request

To retrieve the status of a specific file ingestion task job, use the following URI:

```
mftsaas/api/v1/job/<runId>/status
```

GET Status Response

If successful, file ingestion returns the job status and the job details, which includes a list of files and the details and status of each file.

If unsuccessful, the response includes a reason for the failure.

GET Job Logs Request

To retrieve the log files for a specific file ingestion task job, use the following URI:

```
mftsaas/api/v1/job/<runId>/logs
```

GET Job Logs Response

If successful, file ingestion returns the log files for the job.

If unsuccessful, the response includes a reason for the failure.

activityLog resource

Use the activityLog resource to retrieve details for a completed job using the task ID, run ID, or both.

REST API version 1 resource

Use the file ingestion task REST API version 1 resource to retrieve details for a completed job using the task ID, run ID, or both.

GET Request

To request the details for a completed job using the task ID, use the following URI:

```
mftsaas/api/v1/mitasks/activityLog?taskId=<taskId>
```

To request the details for active or a completed job using the run ID, use the following URI:

```
mftsaas/api/v1/mitasks/activityLog?runId=<runId>
```

To specify the number of rows to skip, use the following URI:

```
mftsaas/api/v1/mitasks/activityLog?taskId={{taskID}}&<offset>
```

To specify a row limit, use the following URI:

```
mftsaas/api/v1/mitasks/activityLog?taskId={{taskID}}&<rowLimit>
```

You can use a combination of these options. For example, you can use the following URI:

```
mftsaas/api/v1/mitasks/activityLog?
runId=<runId>&taskId=<taskId>&rowLimit=<rowLimit>&offset=<offset>
```

You can use the following attributes in the activityLog GET URI:

Field	Description
taskId	File ingestion task ID.
runId	File ingestion run ID.
offset	The number of rows to skip. For example, you might want to skip the first three rows.
rowLimit	The maximum number of rows to return. The maximum number you can specify is 100. Default is 25.

Note: You must specify either the taskId or the runId attribute in the GET URI.

GET Response

The activityLog object returns the following attributes:

Field	Description
id	File ingestion job ID.
totaljobCount	Total number of jobs.
taskId	File ingestion task ID.
runId	File ingestion run ID.
startedBy	Name of the user who created the file ingestion task.

Field	Description
startTime	Start time for the job. Uses Coordinated Universal Time (UTC).
endTime	End time for the job. Uses Coordinated Universal Time (UTC).
status	Whether the job completed successfully.
messageText	Error message associated with the job.
successFiles	The number of files that are successfully transferred from source to target.
failedFiles	The number of files that were not transferred from source to target.

GET Example

The following example shows a response to get details for a file ingestion job using task ID:

```
{
    "totalJobCount": 7,
    "jobActivityLog": [
        {
            "id": 1000000200272,
            "taskId": 89882,
            "runId": 137205,
            "startedBy": "b2b_pod1",
            "startTime": "2021-09-13T09:55:13Z",
            "endTime": "2021-09-13T09:55:15Z",
            "status": "FAILED"
        },
        {
            "id": 1000000200270,
            "taskId": 89882,
            "runId": 137204,
            "startedBy": "b2b_pod1",
            "startTime": "2021-09-13T09:52:44Z",
            "endTime": "2021-09-13T09:53:02Z",
            "status": "SUCCESS"
        },
        {
            "id": 1000000200268,
            "taskId": 89882,
            "runId": 137202,
            "startedBy": "b2b_pod1",
            "startTime": "2021-09-13T09:49:55Z",
            "endTime": "2021-09-13T09:50:12Z",
            "status": "SUCCESS"
        },
        {
            "id": 1000000200264,
            "taskId": 89882,
            "runId": 137199,
            "startedBy": "b2b_pod1",
            "startTime": "2021-09-13T09:43:27Z",
            "endTime": "2021-09-13T09:43:42Z",
            "status": "SUCCESS"
        },
        {
            "id": 1000000200262,
            "taskId": 89882,
            "runId": 137198,
            "startedBy": "b2b_pod1",
            "startTime": "2021-09-13T09:13:58Z",
            "endTime": "2021-09-13T09:14:04Z",
            "status": "FAILED"
```

```
        },
        {
            "id": 1000000200261,
            "taskId": 89882,
            "runId": 137197,
            "startedBy": "b2b_pod1",
            "startTime": "2021-09-13T09:13:09Z",
            "endTime": "2021-09-13T09:13:28Z",
            "status": "SUCCESS"
        },
        {
            "id": 1000000200260,
            "taskId": 89882,
            "runId": 137196,
            "startedBy": "b2b_pod1",
            "startTime": "2021-09-13T09:12:21Z",
            "endTime": "2021-09-13T09:12:35Z",
            "status": "SUCCESS"
        }
    ]
}
```

The following example shows a response to get details for a file ingestion job using run ID:

```
{
    "jobActivityLog": [
        {
            "jobStatusResponse": {
                "jobStatus": "FAILED",
                "errorMessage": "[8008 - Create File List] Directory '/root/testnot' not
found ",
                "jobDetails": {
                    "jobNumber": 1000000200262,
                    "status": "Failed",
                    "startTime": "2021-09-13T09:13:58Z",
                    "endTime": "2021-09-13T09:14:04Z",
                    "messageText": "[8008 - Create File List] Directory '/root/testnot'
not found ",
                    "successFiles": 0,
                    "failedFiles": 0,
                    "fileDetails": []
                }
            }
        }
    ]
}


{
    "jobActivityLog": [
        {
            "jobStatusResponse": {
                "jobStatus": "FAILED",
                "errorMessage": "[8008 - Create File List] Directory '/root/testnot' not
found ",
                "jobDetails": {
                    "jobNumber": 1000000200262,
                    "status": "Failed",
                    "startTime": "2021-09-13T09:13:58Z",
                    "endTime": "2021-09-13T09:14:04Z",
                    "messageText": "[8008 - Create File List] Directory '/root/testnot'
not found ",
                    "successFiles": 0,
                    "failedFiles": 0,
                    "fileDetails": []
                }
            }
        }
    ]
}
```

REST API version 2 resource

Use the file ingestion task REST API version 2 resource to retrieve details for a given job type using the task ID, run ID, both, or neither.

GET Request

To request details for all jobs in a file ingestion task, use the following URI:

```
mftsaas/api/v2/mitasks/activityLog
```

To request the details for all jobs using the task ID, use the following URI:

```
mftsaas/api/v2/mitasks/activityLog?taskId=<taskId>
```

To request the details for all jobs using the run ID, use the following URI:

```
mftsaas/api/v2/mitasks/activityLog?runId=<runId>
```

To specify the number of rows to skip, use the following URI:

```
mftsaas/api/v2/mitasks/activityLog?taskId=<taskId>&offset=<offset>
```

To specify a row limit, use the following URI:

```
mftsaas/api/v2/mitasks/activityLog?taskId=<taskId>&rowLimit<rowLimit>
```

To specify a job type, use the following URI:

```
mftsaas/api/v2/mitasks/activityLog?jobType=<jobType>
```

To specify the number of file events to display and include the file event limit, use the following URI:

```
mftsaas/api/v2/mitasks/activityLog?
runId={{runID}}&fetchFileEvents=true&fileEventsLimit=10
```

You can use a combination of these options. For example, you can use the following URI:

```
mftsaas/api/v2/mitasks/activityLog?
runId=<runId>&taskId=<taskId>&rowLimit=<rowLimit>&offset=<offset>
```

You can use the following attributes in the activityLog GET URI:

Field	Description
taskId	File ingestion task ID.
runId	File ingestion run ID.
offset	The number of rows to skip. For example, you might want to skip the first three rows.
rowLimit	The maximum number of rows to return. The maximum number you can specify is 100. Default is 25.
jobType	Types of jobs to include in the response. You can use one of the following options: - all - completed - active Default is all.

Field	Description
fetchFileEvents	Determines if you want to display the file events. Set to one of the following values: - true. The file ingestion task displays the job file events. - false. The file ingestion task does not display the job file events. Default is false.
fileEventsLimit	The number of file events to return. Applies when **fetchFileEvents** is set to `true`. The maximum number you can specify is 1000. Default is 100.

GET Response

The activityLog object returns the following attributes:

Field	Description
totaljobCount	Total number of jobs.
taskId	File ingestion task ID.
runId	File ingestion run ID.
startedBy	Name of the user who created the file ingestion task.
startTime	Start time of the job. Uses Coordinated Universal Time (UTC).
endTime	End time of the job. Uses Coordinated Universal Time (UTC).
status	Whether the job completed successfully.
logLocation	The location of the session log.
messageText	Remarks associated with the job status.
successFiles	The number of files that are successfully transferred from source to target.
failedFiles	The number of files that were not transferred from source to target.

GET Example

The following example shows a response to a request to get details for a file ingestion job using task ID:

```
{
    "totalJobCount": 1,
    "jobActivityLog": [
        {
            "taskId": "gS7ivoMYoOli6v7TR7MIIe",
            "jobs": [
                {
                    "runId": 490176,
                    "startedBy": "b2b_pod1",
                    "startTime": "2023-05-08T00:20:14Z",
                    "endTime": "2023-05-08T00:20:19Z",
                    "status": "SUCCESS"
                     "logLocation": "data/taskLogs/2023-05-08/1000000916286.log"
```

```
                                "messageText": "Job completed",
                                "successFiles": 8,
                                "failedFiles": 0

                        }
                    }
                ]
        }
```

The following example shows a response to a request to get details for a file ingestion job with fetchFileEvents set to true:

```
{
    "totalJobCount": 1,
    "jobActivityLog": [
        {
            "taskId": "34owubT0kvFe18LdELjXop",
            "jobs": [
                {
                    "runId": 496420,
                    "startedBy": "atl",
                    "startTime": "2023-05-08T07:25:25Z",
                    "endTime": "2023-05-08T07:25:54Z",
                    "status": "SUCCESS",
                    "logLocation": "data/taskLogs/2023-05-08/1000000855619.log",
                    "messageText": "Job completed normally",
                    "successFiles": 1003,
                    "failedFiles": 0,
                    "fileDetails": [
                        {
                            "path": "/root/Arun/May/Target/File_5.txt",
                            "size": 12,
                            "lastModified": "2023-05-04T09:00:30Z",
                            "startTime": "2023-05-08T07:25:12Z",
                            "status": "SUCCESS",
                            "duration": 1,
                            "transferDirection": "UPLOAD"
                        },
                        {
                            "path": "/root/Arun/May/Target/File_12.txt",
                            "size": 12,
                            "lastModified": "2023-05-04T09:00:30Z",
                            "startTime": "2023-05-08T07:25:12Z",
                            "status": "SUCCESS",
                            "duration": 0,
                            "transferDirection": "UPLOAD"
                        },
                        {
                            "path": "/root/Arun/May/Target/File_2.txt",
                            "size": 12,
                            "lastModified": "2023-05-04T09:00:30Z",
                            "startTime": "2023-05-08T07:25:12Z",
                            "status": "SUCCESS",
                            "duration": 0,
                            "transferDirection": "UPLOAD"
                        },
                        {
                            "path": "/root/Arun/May/Target/File_8.txt",
                            "size": 12,
                            "lastModified": "2023-05-04T09:00:30Z",
                            "startTime": "2023-05-08T07:25:12Z",
                            "status": "SUCCESS",
                            "duration": 0,
                            "transferDirection": "UPLOAD"
                        },
                        {
                            "path": "/root/Arun/May/Target/File_10.txt",
                            "size": 12,
                            "lastModified": "2023-05-04T09:00:30Z",
                            "startTime": "2023-05-08T07:25:12Z",
                            "status": "SUCCESS",
```

```
                                            "duration": 1,
                                            "transferDirection": "UPLOAD"
                                        },
                                        {
                                            "path": "/root/Arun/May/Target/File_14.txt",
                                            "size": 12,
                                            "lastModified": "2023-05-04T09:00:30Z",
                                            "startTime": "2023-05-08T07:25:12Z",
                                            "status": "SUCCESS",
                                            "duration": 1,
                                            "transferDirection": "UPLOAD"
                                        },
                                        {
                                            "path": "/root/Arun/May/Target/File_16.txt",
                                            "size": 12,
                                            "lastModified": "2023-05-04T09:00:30Z",
                                            "startTime": "2023-05-08T07:25:12Z",
                                            "status": "SUCCESS",
                                            "duration": 0,
                                            "transferDirection": "UPLOAD"
                                        },
                                        {
                                            "path": "/root/Arun/May/Target/File_18.txt",
                                            "size": 12,
                                            "lastModified": "2023-05-04T09:00:30Z",
                                            "startTime": "2023-05-08T07:25:12Z",
                                            "status": "SUCCESS",
                                            "duration": 0,
                                            "transferDirection": "UPLOAD"
                                        },
                                        {
                                            "path": "/root/Arun/May/Target/File_6.txt",
                                            "size": 12,
                                            "lastModified": "2023-05-04T09:00:30Z",
                                            "startTime": "2023-05-08T07:25:12Z",
                                            "status": "SUCCESS",
                                            "duration": 0,
                                            "transferDirection": "UPLOAD"
                                        },
                                        {
                                            "path": "/root/Arun/May/Target/File_13.txt",
                                            "size": 12,
                                            "lastModified": "2023-05-04T09:00:30Z",
                                            "startTime": "2023-05-08T07:25:12Z",
                                            "status": "SUCCESS",
                                            "duration": 0,
                                            "transferDirection": "UPLOAD"
                                        }
                                    ]
                                }
                            ]
                        }
                    ]
                }
```

tasks resource

Use the tasks resource to create, update, delete, and view file ingestion tasks.

Running and monitoring file ingestion tasks involves a series of requests and responses. Use the followings methods to perform file ingestion tasks:

- Send a tasks GET request to view a list of all file ingestion tasks. See <u>"View file ingestion tasks" on page 450</u>.

- Send a tasks POST request to create a file ingestion task. See .
- Send a tasks PUT request to update a file ingestion task. See .
- Send a tasks GET request to view the location of a file ingestion task. See .
- Send a tasks DELETE request to delete a file ingestion task. See .

View file ingestion tasks

Use the GET request to view file ingestion tasks.

GET request

To view the details of a particular file ingestion task, include the file ingestion in the following URI:

```
mftsaas/api/v1/mitasks/{{TASK-ID}}
```

To view the details for all file ingestion tasks in the organization, omit the file ingestion ID.

```
mftsaas/api/v1/mitasks
```

For example:

```
GET https://na1.dm-us.informaticacloud.com/mftsaas/api/v1/mitasks
```

GET response

Returns the task object if successful or an error object if errors occur.

The task object includes the following information about each of the file ingestion tasks in the organization:

Field	Type	Description
taskId	String	ID number associated with the task.
taskName	String	Name of the task.
description	String	Description of the task.
location	String	Project and folder path where the task exists.
createTime	Date/time	Time when the task was created.
updateTime	Date/time	Time when the task was last updated.

Note: The create and update time in the response are in UTC time.

GET response example to view all file ingestion tasks

The following sample response shows that there are three file ingestion tasks in the organization:

```
{
    "mitasks": [
        {
            "id": "1ONE5Vewzztl0tuKR0EDum",
            "name": "A01_UMAR_MITASK2318",
            "description": "",
            "sourceType": "CONNECTION",
            "sourceConnection": {
                "id": "",
                "name": "",
```

```
                    "type": "local"
            },
            "targetConnection": {
                "id": "0100000B000000000002",
                "name": "ftps",
                "type": "Advanced FTPS"
            },
            "agentGroupId": "01000025000000000002",
            "updatedTime": "2019-01-30T11:17:49Z"
        },
        {
            "id": "9D1tGkAxopJeFmUWoG4s48",
            "name": "A01_UMAR_MITASK3354",
            "description": "",
            "sourceType": "CONNECTION",
            "sourceConnection": {
                "id": "0100000B00000000000M",
                "name": "AzureBlob",
                "type": "Azure Blob"
            },
            "targetConnection": {
                "id": "0100000B00000000000L",
                "name": "SFTP_Conn",
                "type": "Advanced SFTP"
            },
            "agentGroupId": "01000025000000000002",
            "updatedTime": "2019-01-30T06:42:19Z"
        },
        {
            "id": "4hcTFqKVOQrl1z4d6pGUMP",
            "name": "A01_UMAR_MITASK5124",
            "description": "",
            "sourceType": "CONNECTION",
            "sourceConnection": {
                "id": "0100000B0000000004IO",
                "name": "S3",
                "type": "AmazonS3"
            },
            "targetConnection": {
                "id": "",
                "name": "",
                "type": "local"
            },
            "agentGroupId": "01000025000000000002",
            "updatedTime": "2019-01-30T06:35:01Z"
        }]
    }
```

Get response example showing a file ingestion task with file pattern as the file pickup option

The following sample response shows details of a file ingestion task.

```
IDS-SESSION-ID:{{IDS-SESSION-ID}}
Accept:application/json
{
    "id": "j9OLB12nqYObykdFSUMpO2",
    "name": "FTPSrcTarget",
    "location": {
        "projectId": "dNC6zbp2lI8ghrKPo6hpwn",
        "projectName": "Hardening"
    },
    "description": "",
    "sourceType": "CONNECTION",
    "sourceConnection": {
        "id": "0100000B00000000028M",
        "name": "CCI_FTPS",
        "type": "Advanced FTPS V2"
    },
    "targetConnection": {
        "id": "0100000B0000000001JR",
        "name": "CCI_FTP_Lin",
```

```
                "type": "Advanced FTP V2"
            },
            "sourceParameters": {
                "filePattern": "*.txt",
                "sourceTransferMode": "AUTO",
                "filePatternType": "WILDCARD",
                "includeSubfolder": "false",
                "sourceDirectory": "/root/suraj/qa/test/automation/RSFiles",
                "checkDuplicate": "false",
                "fileStability": "true",
                "stabilityCheckInterval": "60",
                "postPickupAction": "KEEP"
            },
            "targetParameters": {
                "fileExistsAction": "APPEND_TIMESTAMP",
                "targetDirectory": "/",
                "targetTransferMode": "AUTO"
            },
            "agentGroupId": "01000025000000000003",
            "createdTime": "2019-02-04T10:34:08Z",
            "updatedTime": "2019-02-04T11:04:02Z",
            "filePickupOption": "PATTERN"
    }
```

GET response example showing a file ingestion task with file list (file path) as the file pickup option

The following sample response shows a file ingestion task with filePickupOption type as FILELIST and a filePickupFilePath in its sourceParameters, indicating that this task reads the designated pickup file to identify which files need to be processed.

```
    {
        "id": "aFHWKrr1RwycuBRBLTtt2t",
        "name": "FilePath_CheckStability",
        "location": {
            "projectId": "0ggRhrI8ZziguyBxHBzuG0",
            "projectName": "Default"
        },
        "description": "",
        "sourceType": "CONNECTION",
        "sourceConnection": {
            "id": "",
            "name": "",
            "type": "local"
        },
        "targetConnection": {
            "id": "",
            "name": "",
            "type": "local"
        },
        "sourceParameters": {
            "filePickupFilePath": "test.txt",
            "sourceDirectory": "/root/test",
            "checkDuplicate": "false",
            "stabilityCheckInterval": "60",
            "postPickupAction": "KEEP",
            "filepickupByName": "FILEPATH",
            "batchSize": "5",
                    "fileStability": "true",
                    "stabilityCheckInterval": "60"
        },
        "targetParameters": {
            "fileExistsAction": "OVERWRITE",
            "targetDirectory": "/root/testCheckStability"
        },
        "agentGroupId": "01001D25000000000002",
        "createdTime": "2021-08-13T09:38:03Z",
        "updatedTime": "2021-08-13T09:39:02Z",
        "logLevel": "NORMAL",
```

```
        "filePickupOption": "FILELIST"
    }
```

GET response example showing a file ingestion tasks with file list as the file pickup option

The following sample response shows a file ingestion task with filePickupOption type as FILELIST, filepickupByName as LISTOFFILES, and a filePickupFileList in its sourceParameters, indicating that this task reads and identifies the designated pickup files to be processed.

```
{
    "id": "2bTlAolXbAGlE7I5qauSAW",
    "name": "DedupFilelist_pushdown",
    "location": {
        "projectId": "0ggRhrI8ZziguyBxHBzuG0",
        "projectName": "Default"
    },
    "description": "",
    "sourceType": "CONNECTION",
    "sourceConnection": {
        "id": "01001D0B0000000005PP",
        "name": "ADLSGen2",
        "type": "Azure Data Lake Gen2"
    },
    "targetConnection": {
        "id": "01001D0B0000000005PU",
        "name": "AzureDW_Gen2",
        "type": "Azure DW"
    },
    "sourceParameters": {
        "sourceDirectory": "/B2B/MI",
        "checkDuplicate": "true",
        "postPickupAction": "KEEP",
        "filepickupByName": "LISTOFFILES",
        "blockSize": "8388608",
        "filePickupFileList": "File1.txt,File2.txt",
        "batchSize": "5",
        "timeoutInterval": "60",
        "fileStability": "true",
        "stabilityCheckInterval": "60"
    },
    "targetParameters": {
        "commandType": "auto",
        "targetTableName": "test1234",
        "isPushdown": "true",
        "ingestionMethod": "polybase",
        "targetSchemaName": "testing",
        "isTruncateTarget": "true"
    },
    "agentGroupId": "01001D25000000000002",
    "createdTime": "2021-04-29T08:47:57Z",
    "updatedTime": "2021-04-29T08:47:57Z",
    "logLevel": "NORMAL",
    "filePickupOption": "FILELIST"
}
```

Create a file ingestion task

Use a POST request to create a file ingestion task.

POST request

To create a file ingestion task through the API, use the following URI:

```
mftsaas/api/v1/mitasks
```

Include the following fields in the request:

Field	Type	Required	Description
name	String	Yes	Name of the task.
location	String	-	Location of the project.
projectId	String	-	ID number associated with the project.
projectName	String	-	Name of the project.
description	String	-	Description of the task.
sourceConnection	String	-	Directory from where files are transferred.
sourceType	String	Yes	Determines the type where files are transferred. Enter one of the following options: - `CONNECTION`. Use connection as a source. - `FILELISTENER`. Use file listener as a source.
includesubfolder	String	-	Determines whether to include the files in sub-folders in the transfer. Set the value to `true` to transfer files from all sub-folders under the defined source directory. Values are `true` or `false`.
checkDuplicate	String	-	Determines whether to check for duplicate files. Values are `true` or `false`. Set the value to `true` to check duplicate files and deny file transfer. If the value is set to `false` all files are transferred.
filePickupOption	String	Yes	Determines the file pickup method. Enter one of the following options: - `FILELIST`. The file ingestion task picks up files based on a file list. - `PATTERN`. The file ingestion task picks up files by pattern.
allowConcurrency	String	-	Determines whether to run multiple jobs concurrently. Set the value to `true` to run multiple jobs concurrently, else set the value to `false`. **Warning:** Running concurrent jobs might cause unexpected results if the targets include duplicate files.
filePatternType	String	Yes	This applies when filePickupOption is PATTERN. File pattern type used to select files to transfer. Enter one of the following options: - wildcard - regex

Field	Type	Required	Description

filePattern	String	Yes	Enter file pattern types, depending on the file pattern that you have selected.
			- wildcard. You can use the following wildcard character filters:
			  - An asterisk (*) matches any number of characters.
			  - A question mark (?) matches a single character.
			- regex. Use regular expression to match the file pattern. Consider the following examples:
			  - Use the following syntax to listen to all files except for files with a name that contains out, foo, and baz: `^(?!.*(?:out|baz|foo)).*$ all except`
			  - Use the following syntax to listen to all files with doc, docx, and pdf extensions: `([a-zA-Z0-9\s_\\.\-\(\):])+ (.doc|.docx|.pdf)$`

filepickupByName	String	Yes	This applies when filePickupOption is FILELIST. Enter one of the following options: - `filepath`. Provide the path that contains the list of files to pick up and enter the file path. - `listoffiles`. Provide the list of files to pick up and enter a comma-separated list of file names. Ensure there is no space before or after specifying the file name.
fileStability	Boolean	-	Determines if the task verifies whether the file is stable before picking it up. Enter one of the following values. - `true`. The file ingestion task verifies whether the file is stable before picking it up. - `false`. The file ingestion task does not verify whether the file is stable before picking it up. Default is `false`.
stabilityCheckInterval	Int	-	Time in seconds that a file ingestion task waits to check the file stability. You can specify a value in the stabilityCheckInterval field only if the fileStability option is set to `true`. The stability check interval ranges between 10 seconds to 300 seconds.
postPickupAction	String	-	Determines what to do with source files after the transfer of files. The following options are available: - KEEP. Keep the files in the source directory. - DELETE. Delete the files from the source directory. - RENAME. Rename the files in the source directory. You must specify a file name suffix that File ingestion adds to the file name when renaming the files. - ARCHIVE. Archive the files to a different location. You must specify an archive directory.
targetConnection	String	Yes	Directory details to which files are transferred.
taskActions	String	-	Actions to process files in the file ingestion task. If you add multiple actions, file ingestion processes files in a sequence.

Field	Type	Required	Description
actions			File processing action. Enter the following file processing actions: - To compress files, enter **Compression**. - To decompress files, enter **Decompression**. - To encrypt files, enter **Encryption**. - To dencrypt files, enter **Decryption**. - To move files from multiple folders to a single folder and to rename file in the target directory, enter **File Operations**. - To scan files for viruses by using the ICAP protocol, enter **Virus Scan**.
action type			Enter the action type depending on the action that you add. To compress files use one of the following methods. - Zip - Tar - Gzip - Bzip2 To decompress files use one of the following methods. - Unzip - Untar - Gunzip - Bunzip2 To encrypt files add PGP. Enter the key ID in the properties. **Note:** The file ingestion task uses the PGP method to encrypt files. Generate a key ring using the CLI. Enter the key ring in the **Key ID**. For more information about the keyring CLIs, refer to key ring command reference in *Tasks*. To decrypt files, add PGP. Enter the key passphrase in the properties. **Note:** The file ingestion task uses the PGP method to encrypt files. Generate the key passphrase using the CLI. Enter the key passphrase in the **Key Passphrase**. For more information about the keyring CLIs, refer to key ring command reference in *Tasks*.

POST request example

Use this sample as a reference to create a file ingestion task with file pattern as the file pickup option

```
POST <serverURL>/public/core/v1/mitasks
Content-Type: application/json
Accept:application/json
Content-Type:application/json
IDS-SESSION-ID:{{IDS-SESSION-ID}}
{
    "name": "Green Green v2",
    "location": {
        "projectId": "9JDNOBX9M31e2AD1dIUv6M",
        "projectName": "Default"
    },
    "description": "",
    "sourceType": "CONNECTION",
    "sourceConnection": {
        "id": "",
        "name": "",
        "type": "local"
    },
        "sourceParameters": {
            "filePattern": "*.txt",
            "filePatternType": "WILDCARD",
```

```
                "includeSubfolder": "false",
                "sourceDirectory": "C:\\Monitor",
                "checkDuplicate": "false",
                "fileStability": "true",
                "stabilityCheckInterval": "60",
                "postPickupAction": "KEEP"

        },
        "targetConnection": {
            "id": "0100000B000000000002",
            "name": "ADLS",
            "type": "Azure Data Lake"
    },
            "targetParameters": {
            "adlsTargetLocation": "/satyen/green"

        },
        "agentGroupId": "01000025000000000002",
        "filePickupOption": "PATTERN",
        "logLevel": "NORMAL",
        "allowConcurrency": "true",
            taskActions":[
                    {
                    "action": "Compression",
                    "actionType": "Zip",
                    "properties": {}
                    }
                    ]
    }
```

Use this sample as a reference to create a file ingestion task with file path as the file pickup option.

```
POST <serverURL>/public/core/v1/mitasks
Content-Type: application/json
Accept:application/json
Content-Type:application/json
IDS-SESSION-ID:{{IDS-SESSION-ID}}
{
    "name": "FilePath_RestAPI1",
    "location": {
        "projectId": "0ggRhrI8ZziguyBxHBzuG0",
        "projectName": "Default"
    },
    "description": "",
    "sourceType": "CONNECTION",
    "sourceConnection": {
        "id": "",
        "name": "",
        "type": "local"
    },
    "targetConnection": {
        "id": "",
        "name": "",
        "type": "local"
    },
    "sourceParameters": {
        "filePickupFilePath": "test.txt",
        "sourceDirectory": "/root/test",
        "checkDuplicate": "false",
        "fileStability": "true",
        "stabilityCheckInterval": "60",
        "postPickupAction": "KEEP",
        "filepickupByName": "FILEPATH",
        "batchSize": "5"

    },
    "targetParameters": {
    "fileExistsAction": "OVERWRITE",
    "targetDirectory": "/root/testCheckStability"
    },
```

```
        "agentGroupId": "01001D25000000000002",
        "logLevel": "NORMAL",
        "filePickupOption": "FILELIST",
        "allowConcurrency": "true"
    }
```

Use this sample as a reference to create a file ingestion task with file list as the file pickup option.

```
POST <serverURL>/public/core/v1/mitasks
Content-Type: application/json
Accept:application/json
Content-Type:application/json
IDS-SESSION-ID:{{IDS-SESSION-ID}}
{

    "name": "DedupFilelist_RestAPI",
    "location": {
        "projectId": "0ggRhrI8ZziguyBxHBzuG0",
        "projectName": "Default"
    },
    "description": "",
    "sourceType": "CONNECTION",
    "sourceConnection": {
        "id": "01001D0B0000000005PP",
        "name": "ADLSGen2",
        "type": "Azure Data Lake Gen2"
    },
    "targetConnection": {
        "id": "01001D0B0000000005PU",
        "name": "AzureDW_Gen2",
        "type": "Azure DW"
    },
    "sourceParameters": {
        "sourceDirectory": "/B2B/MI",
        "checkDuplicate": "true",
        "postPickupAction": "KEEP",
        "filepickupByName": "LISTOFFILES",
        "blockSize": "8388608",
        "filePickupFileList": "File1.txt,File2.txt",
        "batchSize": "5",
        "timeoutInterval": "60",
        "fileStability": "true",
        "stabilityCheckInterval": "60"

    },
    "targetParameters": {
        "commandType": "auto",
        "targetTableName": "test1234",
        "isPushdown": "true",
        "ingestionMethod": "polybase",
        "targetSchemaName": "testing",
        "isTruncateTarget": "true"
    },
    "agentGroupId": "01001D25000000000002",
    "logLevel": "NORMAL",
    "filePickupOption": "FILELIST",
    "allowConcurrency": "true"
}
```

POST response example

If the request is successful, you might receive a response similar to the following example:

```
{
    "id": "cEMWKpibm44bNf5aMjbJ4U",
    "name": "Green Green v2",
    "location": {
        "projectId": "9JDNOBX9M31e2AD1dIUv6M",
        "projectName": "Default"
    },
```

```
        "description": "",
        "sourceType": "CONNECTION",
        "sourceConnection": {
            "id": "",
            "name": "",
            "type": "local"
},
        "sourceParameters": {
            "filePattern": "*.txt",
            "filePatternType": "WILDCARD",
            "includeSubfolder": "false",
            "sourceDirectory": "C:\\Monitor",
            "checkDuplicate": "false",
            "fileStability": "true",
            "stabilityCheckInterval": "60",
            "postPickupAction": "KEEP"

        },
        "targetConnection": {
            "id": "0100000B000000000002",
            "name": "ADLS",
            "type": "Azure Data Lake"
},
        "targetParameters": {
            "adlsTargetLocation": "/satyen/green"

        },
        "agentGroupId": "01000025000000000002",
        "createdTime": "2018-08-27T07:03:32Z",
        "updatedTime": "2018-08-29T12:14:58Z""
                    taskActions":[
                     {
                            "action": "Compression",
                            "actionType": "Zip",
                            "properties": {}
                     }
                    ]
    }
    }
```

Note: The created and updated time in the response is displayed in the UTC time.

Update a file ingestion task

Use a PUT request to update a file ingestion task.

PUT request

To update a file ingestion task, use the following URI:

```
mftsaas/api/v1/mitasks/<taskID>
```

Include the following fields in the PUT request:

Field	Type	Required	Description
id	String	-	ID number of the task.
name	String	Yes	Name of the task.
description	String	-	Description of the task.

Field	Type	Required	Description
sourceType	String	Yes	Determines the type where files are transferred. Enter one of the following options: - `CONNECTION`. Use connection as a source. - `FILELISTENER`. Use file listener as a source.
sourceConnection	String	-	Directory from where files are transferred.
includeSubfolder	String	-	Values are `true` or `false`. Set the value to `true` to transfer files from all sub-folders under the defined source directory.
checkDuplicate	String	-	Values are `true` or `false`. Set the value to `true` to check duplicate files and deny file transfer. If the value is set to `false` all files are transferred.
filePatternType	String	Yes	File name pattern used to select the files to transfer. Enter one of the following options: - Wildcard - Regex

filePattern	String	Yes	Enter pattern types, depending on the file pattern that you have selected.
			- wildcard. You can use the following wildcard character filters:
			  - An asterisk (*) matches any number of characters.
			  - A question mark (?) matches a single character.
			- Regex. Use regular expression to match the file pattern. Consider the following examples:
			  - Use the following syntax to listen to all files except for files with a name that contains out, foo, and baz: `^(?!.*(?:out|baz|foo)).*$` à all except
			  - Use the following syntax to listen to all files with doc and docx, pdf extensions: `([a-zA-Z0-9\s_\\.\-\(\):])+ (.doc|.docx|.pdf)$` à

fileStability	Boolean	-	Determines if the task verifies whether the file is stable before picking it up. Enter one of the following values. - `true`. The file ingestion task verifies whether the file is stable before picking it up. - `false`. The file ingestion task does not verify whether the file is stable before picking it up. Default is `false`.
stabilityCheckInterval	Int	-	Time in seconds that a file ingestion task waits to check the file stability. You can specify a value in the stabilityCheckInterval field only if the fileStability option is set to `true`. The stability check interval ranges between 10 seconds to 300 seconds.

Field	Type	Required	Description
postPickupAction	String	-	Determines what to do with source files after the files transfer. The following options are available: - KEEP. Keep the files in the source directory. - DELETE. Delete the files from the source directory. - RENAME. Rename the files in the source directory. You must specify a file name suffix that file ingestion adds to the file name when renaming the files. - ARCHIVE. Archive the files to a different location. You must specify an archive directory.
targetConnection	String	Yes	Directory details to which files are transferred.

PUT request example

Use this sample as a reference to update a file ingestion task.

```
PUT <serverUrl>/public/core/v1/mitasks
Content-Type: application/json
Accept:application/json
Content-Type:application/json
IDS-SESSION-ID:{{IDS-SESSION-ID}}
{
    "id": "cEMWKpibm44bNf5aMjbJ4U",
    "name": "Green Green v2",
    "description": "Green Green v2 Description",
    "sourceType": "CONNECTION",
    "sourceConnection": {
        "id": "",
        "name": "",
        "type": "local"
},
        "sourceParameters": {
            "filePattern": "*.txt",
            "filePatternType": "WILDCARD",
            "includeSubfolder": "false",
            "sourceDirectory": "C:\\Monitor",
            "checkDuplicate": "false",
            "fileStability": "true",
            "stabilityCheckInterval": "60",
            "postPickupAction": "KEEP"

    },
    "targetConnection": {
        "id": "0100000B000000000002",
        "name": "ADLS",
        "type": "Azure Data Lake"
},
        "targetParameters": {
            "adlsTargetLocation": "/satyen/green"
    },
    "agentGroupId": "01000025000000000002
}
```

PUT response example

If the request is successful, you might receive a response similar to the following example:

```
{
    "id": "cEMWKpibm44bNf5aMjbJ4U",
    "name": "Green Green v2",
    "description": "",
    "sourceType": "CONNECTION",
    "sourceConnection": {
        "id": "",
```

```
                   "name": "",
                   "type": "local"
          },
                   "sourceParameters": {
                       "filePattern": "*.txt",
                       "filePatternType": "WILDCARD",
                       "includeSubfolder": "false",
                       "sourceDirectory": "C:\\Monitor",
                       "checkDuplicate": "false",
                       "fileStability": "true",
                       "stabilityCheckInterval": "60",
                       "postPickupAction": "KEEP"

          },
          "targetConnection": {
               "id": "0100000B000000000002",
               "name": "ADLS",
               "type": "Azure Data Lake"
          },
                   "targetParameters": {
                       "adlsTargetLocation": "/satyen/green"

          },
          "agentGroupId": "01000025000000000002",
          "createdTime": "2018-08-27T07:03:32Z",
          "updatedTime": "2018-08-29T12:14:58Z"
      }
```

Note: The created and updated time in the response is displayed in the UTC time.

View the location of a file ingestion task

Use the GET request to view the location of a file ingestion task.

GET request

Use the following URI to get the location of a file ingestion task.

```
/api/v1/mitasks?resolveLocation=true
```

GET response example

If the request to get the file location of a file ingestion task is successful, you might receive a response similar to the following example:

```
{
    "miTasks": [
        {
            "id": "1ONE5Vewzztl0tuKR0EDum",
            "name": "A01_UMAR_MITASK2318",
            "location": {
                "folderId": "digFZU6HMo4gCKYghtQvgD",
                "folderName": "A_01_UMAR",
                "projectId": "503RTpKDSSLlwmkwTXL0Qx",
                "projectName": "Default"
            },
            "description": "",
            "sourceType": "CONNECTION",
            "sourceConnection": {
                "id": "",
                "name": "",
                "type": "local"
            },
            "targetConnection": {
                "id": "0100000B000000000002",
                "name": "ftps",
                "type": "Advanced FTPS"
            },
```

```
                "agentGroupId": "01000025000000000002",
                "createdTime": "2019-01-28T09:54:53Z",
                "updatedTime": "2019-01-30T11:17:49Z"
            },
            {
                "id": "9D1tGkAxopJeFmUWoG4s48",
                "name": "A01_UMAR_MITASK3354",
                "location": {
                    "folderId": "digFZU6HMo4gCKYghtQvgD",
                    "folderName": "A_01_UMAR",
                    "projectId": "503RTpKDSSLlwmkwTXL0Qx",
                    "projectName": "Default"
                },
                "description": "",
                "sourceType": "CONNECTION",
                "sourceConnection": {
                    "id": "0100000B00000000000M",
                    "name": "AzureBlob",
                    "type": "Azure Blob"
                },
                "targetConnection": {
                    "id": "0100000B00000000000L",
                    "name": "SFTP_Conn",
                    "type": "Advanced SFTP"
                },
                "agentGroupId": "01000025000000000002",
                "createdTime": "2019-01-30T06:36:28Z",
                "updatedTime": "2019-01-30T06:42:20Z"
            }
        ]
    }
```

Delete a file ingestion task

Use the DELETE request to delete a file ingestion task.

DELETE request

To delete a file ingestion include the task ID of the task through the API, in the following URI:

```
mftsaas/api/v1/mitasks/<taskID>
```

CHAPTER 6

Mass Ingestion Streaming REST API

Use streaming ingestion resources to deploy, undeploy, start, stop, copy, and update streaming ingestion tasks and to monitor streaming ingestion jobs.

When you use the streaming ingestion resource, use the following request header format:

```
<METHOD><base URL>
Content-Type: application/json
Accept: application/json
IDS-SESSION-ID: <SessionId>
```

Dataflows resource

Use the Dataflows resource to deploy, undeploy, start, and stop streaming ingestion tasks.

Use the following base URL:

```
<server URI>/sisvc/api/v1/Dataflows('<dataflow ID>')/OData.SI.<API name>
```

Note: If you use a tool such as Postman that automatically includes the HTTP version, do not enter the HTTP version in the URL. If the HTTP version appears twice in the URL, the request fails.

Deploying a streaming ingestion task

Use a POST request to deploy a streaming ingestion task.

POST request

To deploy a streaming ingestion task, use the following URL:

```
<server URI>/sisvc/api/v1/Dataflows('<dataflow ID>')/OData.SI.Deploy
```

A request body is not required because the URL passes the dataflow ID.

POST request example

To deploy a streaming ingestion task, you might send a request similar to the following example:

```
POST <serverUrl>/sisvc/api/v1/Dataflows('50077311-d4a4-437c-9218-c3596d1f182f')/
OData.SI.Deploy
Content-Type: application/json
Accept:application/json
IDS-SESSION-ID:2l0oeVx22Rujiej7yTokmT
```

POST response example

If the request is successful, you might receive a response similar to the following example:

```
{
    "@odata.context": "$metadata#OData.SI.DeploymentResult",
    "successful": true,
    "code": null,
    "errorMessage": null
}
```

Undeploying a streaming ingestion task

Use a POST request to undeploy a streaming ingestion task.

POST request

To undeploy a streaming ingestion task, use the following URL:

```
<server URI>/sisvc/api/v1/Dataflows('<dataflow ID>')/OData.SI.Undeploy
```

A request body is not required because the URL passes the dataflow ID.

POST request example

To undeploy a streaming ingestion task, you might send a request similar to the following example:

```
POST <serverUrl>/sisvc/api/v1/Dataflows('50077311-d4a4-437c-9218-c3596d1f182f')/
OData.SI.Undeploy
Content-Type: application/json
Accept:application/json
IDS-SESSION-ID:2l0oeVx22Rujiej7yTokmT
```

POST response example

If the request is successful, you might receive a response similar to the following example:

```
{
    "@odata.context": "$metadata#OData.SI.DeploymentResult",
    "successful": true,
    "code": null,
    "errorMessage": null
}
```

Starting a streaming ingestion task

Use a POST request to start a streaming ingestion task.

POST request

To start a streaming ingestion task, use the following URL:

```
<server URI>/sisvc/api/v1/Dataflows('<dataflow ID>')/OData.SI.Start
```

A request body is not required because the URL passes the dataflow ID.

POST request example

To start a streaming ingestion task, you might send a request similar to the following example:

```
POST <serverUrl>/sisvc/api/v1/Dataflows('50077311-d4a4-437c-9218-c3596d1f182f')/
OData.SI.Start
Content-Type: application/json
Accept:application/json
IDS-SESSION-ID:2l0oeVx22Rujiej7yTokmT
```

POST response example

If the request is successful, you might receive a response similar to the following example:

```
{
    "@odata.context": "$metadata#OData.SI.DeploymentResult",
    "successful": true,
    "code": null,
    "errorMessage": null
}
```

Stopping a streaming ingestion task

Use a POST request to stop a streaming ingestion task.

POST request

To stop a streaming ingestion task, use the following URL:

```
<server URI>/sisvc/api/v1/Dataflows('<dataflowID>')/OData.SI.Stop
```

A request body is not required because the URL passes the dataflow ID.

POST request example

To stop a streaming ingestion task, you might send a request similar to the following example:

```
POST <serverUrl>/sisvc/api/v1/Dataflows('d7572789-dc4c-4c56-bbeb-3772736d61aa')/
OData.SI.Stop
Content-Type: application/json
Accept:application/json
IDS-SESSION-ID:2l0oeVx22Rujiej7yTokmT
```

POST response example

If the request is successful, you might receive a response similar to the following example:

```
{
    "@odata.context": "$metadata#OData.SI.DeploymentResult",
    "successful": true,
    "code": null,
    "errorMessage": null
}
```

CopyEntities resource

Use the CopyEntities resource to copy streaming ingestion tasks.

POST request

To copy streaming ingestion tasks, use the following URL:

```
<server URI>/sisvc/restapi/v1/CopyEntities
```

You can include the following fields in the request:

Field	Type	Required	Description
targetLocationID	String	Yes	ID of the target location to copy the objects to.
sourceEntities	Array	Yes	Configuration of the source and target connections.
sourceId	String	Yes	ID of the source object.
targetName	String	Yes	Name of the target object.
targetDescription	String	-	Description of the target object.

POST request example

To copy two streaming ingestion tasks, you might send a request similar to the following example:

```
POST <serverUrl>/sisvc/restapi/v1/CopyEntities
Content-Type: application/json
Accept:application/json
IDS-SESSION-ID:2l0oeVx22Rujiej7yTokmT
{
  "targetLocationID": "2RGmVdwN15PbfnQP5PSoSB",
  "sourceEntities": [
    {
      "sourceId": "5Ff6jeaSh2UfAqiV01ldKD",
      "targetName": "Test_Copy_A",
      "targetDescription": "Description_1"
    },
    {
      "sourceId": "fZnCSqcWTOQkJOr8VCWZQE",
      "targetName": "Test_Copy_B",
      "targetDescription": "Description_2"
    }
  ]
}
```

If the request is unsuccessful, the response includes a reason for the failure.

POST response

When you use a POST request to copy streaming ingestion tasks, it returns a success response if successful or an error object if an error occurs.

POST response example

If the request is successful, you might receive a response similar to the following example:

```
{
  "Status Message": "Operation succeeded on 2 artifacts.",
  "Success": {
    "Test_Copy_A": "ideNJw6l54gizxofF53HQH",
    "Test_Copy_B": "cOQ3gcWKSYikzVqqg6IOok"
  }
}
```

UpdateEntity resource

Use the UpdateEntity resource to update a streaming ingestion task. You can update streaming ingestion tasks that use the following connectors: Amazon Kinesis, Amazon S3 V2, Microsoft Azure Event Hub, Microsoft Azure Data Lake Storage Gen2, Flat file, JDBC V2, JMS, Kafka, or MQTT.

POST request

Use a POST request to update a streaming ingestion task.

To update a streaming ingestion task, use the following URL:

```
<server URI>/sisvc/restapi/v1/UpdateEntity/Documents('<document ID>')
```

You can include the following fields in the request:

Field	Type	Required	Description
name	String	Yes	Name of the task.
description	String	-	Description of the task.
runtimeId	String	Yes	ID of the runtime environment.
currentVersion	String	Yes	The latest dataflow object version.
nodes	Array	Yes	Details of the task source and target connections.

Fields of the nodes array

The fields in the array provide the name, type, and connection ID of the connection. It includes the configuration of the source and target connections in key-value pairs which you can edit. You can include the following fields in the nodes array:

Field	Type	Required	Description
name	String	Yes	Name of the connection.
type	String	Yes	The connection type, source or target.
connectionId	String	Yes	ID of the connection.
transformationType	String	-	Not applicable.
config	Array	Yes	Configuration of the source and target connections.

Connection configuration for tasks with MQTT as a source

When the source connection of the task source is MQTT, you can include the following fields and key-value pairs in the config array of the source connection:

Key	Type	Required	Description
ClientID	String	-	Unique identifier of the connection between the MQTT source and the MQTT broker. The client ID is the file-based persistence store that the MQTT source uses to store messages while they are processed. You can enter a string of up to 255 characters.
MaxQueueSize	Integer	-	The maximum number of messages that the processor can store in memory. You can enter a value between 1 and 2147483647.
Topic	String	Yes	Name of the MQTT topic.

POST request example

To update a streaming ingestion task with an MQTT source and a flat file target, you might send a request similar to the following example:

```
{
  "name": "mqtt to flatfile",
  "description": "mqtt to flatfile",
  "runtimeId": "01000025000000000003",
  "locationId": "5sJ0JDyJyWLlrosS5qJjsQ",
  "currentVersion": "2",
  "messageFormat": "binary",
  "nodes": [
    {
      "name": "mqtt to flatfile_source",
      "type": "source",
      "connectionId": "012MGS0B0000000000010",
      "transformationType": "",
      "config": [
        {
          "key": "ClientID",
          "value": "test"
        },
        {
          "key": "MaxQueueSize",
          "value": 1024
        },
        {
          "key": "Topic",
          "value": "test"
        }
      ]
    },
    {
      "name": "mqtt to flatfile_target",
      "type": "target",
      "connectionId": "012MGS0B00000000002N",
      "transformationType": "",
      "config": [
        {
          "key": "interimDirectory",
          "value": "/home/agent/test"
        },
        {
          "key": "rolloverSize",
          "value": 1024
        },
        {
```

```
                    "key": "rolloverEvents",
                    "value": 100
                },
                {
                    "key": "rolloverTime",
                    "value": 300000
                },
                {
                    "key": "File Name",
                    "value": "test"
                }
            ]
        }
    ],
    "edges": [
        {
            "from": "mqtt to flatfile_source",
            "to": "mqtt to flatfile_target"
        }
    ]
}
```

Connection configuration for tasks with JMS as a source

When the source connection of the task source is JMS, you can include the following fields and key-value pairs in the config array of the source connection:

Key	Type	Required	Description
destinationType	String	Yes	Type of destination that the source service sends JMS messages to. Enter one of the following values: - QUEUE. The JMS provider delivers messages to a single consumer who is registered for the queue. - TOPIC. The JMS provider delivers messages to all active consumers that subscribe to the topic.
clientId	String	Yes	Unique ID of the JMS connection. You can enter a string of up to 255 characters.
sharedSubscription	String	Yes	Enables multiple consumers to access a single subscription. Applies to the TOPIC destination type. Enter one of the following values: - True - False
durableSubscription	String	Yes	When set to True, the JMS source service enables inactive subscribers to retain messages and then deliver them when the subscriber reconnects. Applies to the TOPIC destination type. Enter one of the following values: - True - False
subscriptionName	String	Yes	Name of the subscription. Applies to the TOPIC destination type, when the topic subscription type is shared, durable, or both.
JMS Destination	String	Yes	Name of the queue or topic that the JMS provider delivers messages to.

POST request example

To update a streaming ingestion task with an JMS source and a flat file target, you might send a request similar to the following example:

```
{
    "name": "crud",
```

```
      "description": "JMS to FileToFile",
      "runtimeId": "01000025000000000003",
      "locationId": "5sJ0JDyJyWLlrosS5qJjsQ",
      "currentVersion": "2",
      "messageFormat": "binary",
      "nodes": [
        {
          "name": "crud_source",
          "type": "source",
          "connectionId": "012MGS0B000000000003",
          "transformationType": "",
          "config": [
            {
              "key": "destinationType",
              "value": "QUEUE"
            },
            {
              "key": "clientId",
              "value": ""
            },
            {
              "key": "JMS Destination",
              "value": "test"
            }
          ]
        },
        {
          "name": "crud_target",
          "type": "target",
          "connectionId": "012MGS0B00000000000H",
          "transformationType": "",
          "config": [
            {
              "key": "interimDirectory",
              "value": "/home/agent/test"
            },
            {
              "key": "rolloverSize",
              "value": 1024
            },
            {
              "key": "rolloverEvents",
              "value": 100
            },
            {
              "key": "rolloverTime",
              "value": 300000
            },
            {
              "key": "File Name",
              "value": "test"
            }
          ]
        }
      ],
      "edges": [
        {
          "from": "crud_source",
          "to": "crud_target"
        }
      ]
    }
```

Connection configuration for tasks with Microsoft Azure Data Lake Storage Gen2 (ADLS Gen2) as a target

When the target connection of the task target is ADLS Gen2, you can include the following fields and key-value pairs in the config array of the target connection:

Key	Type	Required	Description
writeStrategy	String	Yes	The action to take when a file by the same name exists in the ADLS Gen2 storage. Enter one of the following values: - Append. Add data to the existing file. - Overwrite. Replaces the existing file with the new file. - Fail. Fail the request. - Rollover. Close the current file and create a new file based on the configured rollover value.
rolloverSize *	Integer	-	Target file size, in KB, at which to trigger rollover. Applies to a Rollover write strategy. You can enter a value between 1 and 2147483647.
rolloverEvents *	Integer	-	Number of events or messages to accumulate before a rollover. Applies to a Rollover write strategy. You can enter a value between 1 and 2147483647.
rolloverTime *	Integer	-	Length of time, in milliseconds, after which to trigger a rollover. Applies to a Rollover write strategy. You can enter a value between 1 and 2147483647.
filesystemNameOverride	String	-	Overrides the default file system name provided in the connection. This file system name is used write to a file at run time. You can enter a string of up to 1,280 characters.
directoryOverride	String	-	Overrides the default directory path. The ADLS Gen2 directory path to write data to. If left blank, the default directory path is used. You can enter a string of up to 1,280 characters.

compressionFormat	String	-	Compression format to use before the streaming ingestion task writes data to the target file.
			Enter one of the following values:
			- None
			- GZIP
			- BZIP2
			- DEFAULT1
			  Enter this value to use the Zlib format.
			- DEFAULT2
			  Enter this value to use the Deflate format.

File Name/Expression	String	Yes	ADLS Gen2 file name or a regular expression. You can enter a string of up to 249 characters.
* Enter a value for at least one of the fields.			

POST request example

To update a streaming ingestion task with a flat file source and an ADLS Gen2 target, you might send a request similar to the following example:

```
{
  "name": "flatfile to adls",
  "description": "flatfile to adls",
  "runtimeId": "01000025000000000003",
  "locationId": "5sJ0JDyJyWLlrosS5qJjsQ",
  "currentVersion": "2",
  "messageFormat": "binary",
  "nodes": [
    {
      "name": "flatfile to adls_source",
      "type": "source",
      "connectionId": "012MGS0B00000000002N",
      "transformationType": "",
      "config": [
        {
          "key": "File",
          "value": "logfile"
        },
        {
          "key": "initialPosition",
          "value": "Current Time"
        },
        {
          "key": "rolloverPattern",
          "value": "test"
        },
        {
          "key": "tailingMode",
          "value": "Single file"
        }
      ]
    },
    {
      "name": "flatfile to adls_target",
      "type": "target",
      "connectionId": "012MGS0B00000000003D",
      "transformationType": "",
      "config": [
        {
          "key": "writeStrategy",
          "value": "Rollover"
        },
        {
          "key": "filesystemNameOverride",
          "value": "test"
        },
        {
          "key": "File Name/Expression",
          "value": "test"
        },
        {
          "key": "compressionFormat",
          "value": "NONE"
        },
        {
          "key": "directoryOverride",
          "value": "/test"
        },
        {
          "key": "interimDirectory",
          "value": "/home/agent/test"
        },
        {
          "key": "rolloverSize",
          "value": 1024
        },
```

```
                  {
                    "key": "rolloverEvents",
                    "value": 100
                  },
                  {
                    "key": "rolloverTime",
                    "value": 300000
                  }
                ]
              }
            ]
          }
```

Connection configuration for tasks with Amazon S3 as a target

When the target connection of the task target is Amazon S3, you can include the following fields and key-value pairs in the config array of the target connection:

Key	Type	Required	Description
partitionTime	String	-	The time interval according to which the streaming ingestion task creates partitions in the Amazon S3 bucket. Enter one of the following values: - None - 5min - 10min - 15min - 20min - 30min - 1hr - 1day
minUploadPartSize	Integer	-	Minimum part size when uploading a large file as a set of multiple independent parts, in megabytes. Use this property to tune the file load to Amazon S3. You can enter a value between 50 and 5120.
multipartUploadThreshold	Integer	-	Multipart threshold when uploading objects in multiple parts in parallel. You can enter a value between 50 and 5120.
Object Name/Expression	String	Yes	Amazon S3 target file name or a regular expression for the Amazon S3 file name pattern.

POST request example

To update a streaming ingestion task with a flat file source and an Amazon S3 as target, you might send a request similar to the following example:

```
    {
      "name": "flatfile to amazon S3",
      "description": "flatfile to amazon S3",
      "runtimeId": "01000025000000000003",
      "locationId": "5sJ0JDyJyWLlrosS5qJjsQ",
      "currentVersion": "2",
      "messageFormat": "binary",
      "nodes": [
        {
          "name": "flatfile to amazon S3_source",
          "type": "source",
          "connectionId": "012MGS0B00000000002N",
```

```json
        "transformationType": "",
        "config": [
          {
            "key": "File",
            "value": "logfile"
          },
          {
            "key": "initialPosition",
            "value": "Current Time"
          },
          {
            "key": "rolloverPattern",
            "value": "test"
          },
          {
            "key": "tailingMode",
            "value": "Single file"
          }
        ]
      },
      {
        "name": "flatfile to amazon S3_target",
        "type": "target",
        "connectionId": "012MGS0B0000000000I7",
        "transformationType": "",
        "config": [
          {
            "key": "partitionTime",
            "value": "None"
          },
          {
            "key": "minUploadPartSize",
            "value": 5120
          },
          {
            "key": "multipartUploadThreshold",
            "value": 5120
          },
          {
            "key": "Object Name/Expression",
            "value": "test"
          }
        ]
      }
    ],
    "edges": [
      {
        "from": "flatfile to amazon S3_source",
        "to": "flatfile to amazon S3_target"
      }
    ]
}
```

Connection configuration for tasks with Azure Event Hubs as a target

When the target connection of the task target is Azure Event Hubs, you can include the following fields and key-value pairs in the config array of the target connection:

Key	Type	Required	Description
sasPolicyName	String	-	The name of the Event Hub Namespace Shared Access Policy. You can enter a string of up to 255 characters.
sasPolicyPrimaryKey	String	-	The primary key of the Event Hub Namespace Shared Access Policy. You can enter a string of up to 255 characters.
Event Hub	String	Yes	The name of the Azure Event Hubs. You can enter a string up to 255 characters. The name can contain lower case characters, upper case characters, numbers, and the special characters - and _.

POST request example

To update a streaming ingestion task with a flat file source and an Azure Event Hubs target, you might send a request similar to the following example:

```
{
  "name": "flatfile to azure event hub",
  "description": "flatfile to azure event hub",
  "runtimeId": "01000025000000000003",
  "locationId": "5sJ0JDyJyWLlrosS5qJjsQ",
  "currentVersion": "2",
  "messageFormat": "binary",
  "nodes": [
    {
      "name": "flatfile to azure event hub_source",
      "type": "source",
      "connectionId": "012MGS0B00000000002N",
      "transformationType": "",
      "config": [
        {
          "key": "File",
          "value": "logfile"
        },
        {
          "key": "initialPosition",
          "value": "Current Time"
        },
        {
          "key": "rolloverPattern",
          "value": "test"
        },
        {
          "key": "tailingMode",
          "value": "Single file"
        }
      ]
    },
    {
      "name": "flatfile to azure event hub_target",
      "type": "target",
      "connectionId": "012MGS0B00000000001S",
      "transformationType": "",
      "config": [
        {
          "key": "sasPolicyName",
          "value": "test"
        },
```

```
              {
                "key": "sasPolicyPrimaryKey",
                "value": "test"
              },
              {
                "key": "Event Hub",
                "value": "test"
              }
            ]
          }
        ],
        "edges": [
          {
            "from": "flatfile to azure event hub_source",
            "to": "flatfile to azure event hub_target"
          }
        ]
      }
```

Connection configuration for tasks with JDBC V2 as a target

When the target connection of the task target is JDBC V2, you can include the following fields and key-value pairs in the config array of the target connection:

Key	Type	Required	Description
Table Name	String	Yes	Name of the table to insert data to in JSON format. Enter a string of up to 988 characters.

POST request example

To update a streaming ingestion task with a flat file source and a JDBC V2 target, you might send a request similar to the following example:

```
{
  "name": "FileFile to jdbc",
  "description": "FileToFile to jdbc_target",
  "runtimeId": "01000025000000000003",
  "locationId": "5sJ0JDyJyWLlrosS5qJjsQ",
  "currentVersion": "2",
  "messageFormat": "binary",
  "nodes": [
    {
      "name": "flatfile to jdbc_source",
      "type": "source",
      "connectionId": "012MGS0B00000000002N",
      "transformationType": "",
      "config": [
        {
          "key": "initialPosition",
          "value": "Current Time"
        },
        {
          "key": "tailingMode",
          "value": "Single file"
        },
        {
          "key": "rolloverPattern",
          "value": "test"
        },
        {
          "key": "File",
          "value": "logfile"
        }
      ]
```

```
      },
      {
        "name": "flatfile to jdbc_target",
        "type": "target",
        "connectionId": "012MGS0B0000000000KF",
        "transformationType": "",
        "config": [
          {
            "key": "Table Name",
            "value": "table"
          }
        ]
      }
    ],
    "edges": [
      {
        "from": "flatfile to jdbc_source",
        "to": "flatfile to jdbc_target"
      }
    ]
  }
```

Connection configuration for tasks with Amazon Kinesis Streams as a source and as a target

When the source and target connection of the task is Amazon Kinesis Streams, you can include the following fields and key-value pairs in the config array of the source and target connection:

Key	Type	Required	Description
appendGUID	Boolean		Specifies whether or not to add a GUID as a suffix to the Amazon DynamoDB table name. Enter one of the following values: - true - false
dynamoDB	String		Amazon DynamoDB table name where to store the checkpoint details of the Kinesis source data. You can enter a string of up to 128 characters.
Stream	String	Yes	Name of the Kinesis Stream to read data from. Enter a string of up to 128 characters. Appears in the source node.
Stream Name/ Expression	String	Yes	Kinesis Stream name or a regular expression to write data to. Enter a string of up to 128 characters. Appears in the target node.

POST request example

To update a streaming ingestion task with an Amazon Kinesis Streams source and target, you might send a request similar to the following example:

```
  {
    "name": "kinesis to kinesis",
    "description": "kinesis to kinesis",
    "runtimeId": "01000025000000000003",
    "locationId": "5sJ0JDyJyWLlrosS5qJjsQ",
    "currentVersion": "2",
    "messageFormat": "binary",
    "nodes": [
```

```
    {
      "name": "kinesis to kinesis_source",
      "type": "source",
      "connectionId": "012MGS0B00000000000F",
      "transformationType": "",
      "config": [
        {
          "key": "appendGUID",
          "value": true
        },
        {
          "key": "dynamoDB",
          "value": "table"
        },
        {
          "key": "Stream",
          "value": "test"
        }
      ]
    },
    {
      "name": "kinesis to kinesis_target",
      "type": "target",
      "connectionId": "012MGS0B00000000000F",
      "transformationType": "",
      "config": [
        {
          "key": "Stream Name/Expression",
          "value": "trgt"
        }
      ]
    }
  ],
  "edges": [
    {
      "from": "kinesis to kinesis_source",
      "to": "kinesis to kinesis_target"
    }
  ]
}
```

Connection configuration for tasks with flat file as a source and as a target

When the source and target connection of the task is flat file, you can include the following fields and key-value pairs in the config array of the source and target connection:

Key	Type	Required	Description
File	String	Yes	Absolute path and name of the source file. Enter the base directory for multiple files mode.
initialPosition	String	Yes	Starting position to read data from the file to tail. Enter one of the following values: - Beginning of File. Read from the beginning of the file. Don't ingest any data that has already been rolled over. - Current Time. Read from the most recently updated part of the file. Don't ingest data that was rolled over or data in the file that was written.

Key	Type	Required	Description
rolloverPattern	String	-	File name pattern for the file that rolls over.
			If the file to tail rolls over, the Secure Agent uses the file name pattern to identify files that have rolled over. If the Secure Agent stops during a file rollover, when it restarts, it picks up the file where it was left off.
			You can use asterisk (*) and question mark (?) as wildcard characters to indicate that the files are rolled over in the same directory. For example,`${filename}.log.*`. Here, asterisk (*) represents the successive version numbers that would be appended to the file name.
tailingMode	String	Yes	Tail a file or multiple files based on the logging pattern. Enter one of the following values: - Single file. Tail one file. - Multiple files. Tail all the files indicated in the base directory. You can enter a regular expression to indicate the files to tail.
File Name	String	Yes	The name of the target file.
interimDirectory	String	Yes	Path to the staging directory on the Secure Agent.
rolloverSize	Integer	Yes	The file size, in KB, at which the task moves the file from the staging directory to the target. You can enter a value between 1 and 2147483647.
rolloverEvents	Integer	Yes	Number of events or messages to accumulate before a file rollover. You can enter a value between 1 and 2147483647.
rolloverTime	Integer	-	Length of time, in milliseconds, after which the target file rolls over. You can enter a value between 1 and 2147483647.
edges	Array	-	Sequence of dataflow execution.

POST request example

To update a streaming ingestion task with a flat file source and target, you might send a request similar to the following example:

```
{
  "name": "FileToFile",
  "description": "FileToFile_V2",
  "runtimeId": "01000025000000000003",
  "locationId": "5sJ0JDyJyWLlrosS5qJjsQ",
  "currentVersion": "2",
  "messageFormat": "binary",
  "nodes": [
    {
      "name": "FileToFile_source",
      "type": "source",
      "connectionId": "0100000B000000000002",
      "transformationType": "",
      "config": [
        {
          "key": "File",
          "value": "siagent.log"
        },
        {
          "key": "initialPosition",
          "value": "Current Time"
        },
        {
```

```
          "key": "rolloverPattern",
          "value": ""
        },
        {
          "key": "tailingMode",
          "value": "Single file"
        }
      ]
    },
    {
      "name": "FileToFile_target",
      "type": "target",
      "connectionId": "0100000B000000000002",
      "transformationType": "",
      "config": [
        {
          "key": "File Name",
          "value": "testing.log"
        },
        {
          "key": "interimDirectory",
          "value": "/home/agent/infa/test_file_target"
        },
        {
          "key": "rolloverSize",
          "value": 100
        },
        {
          "key": "rolloverEvents",
          "value": 100
        },
        {
          "key": "rolloverTime",
          "value": 100
        }
      ]
    }
  ],
  "edges": [
    {
      "from": "FileToFile_source",
      "to": "FileToFile_target"
    }
  ],
  "runtimeOptions": {
    "maxLogSize": {
      "value": 10,
      "unit": "MB"
    },
    "logLevel": "INFO"
  }
}
```

Connection configuration for tasks with Kafka as a source and as a target

When the source and target connection of the task is Kafka, you can include the following fields and key-value pairs in the config array of the source and target connection:

Key	Type	Required	Description
Topic	String	Yes	Kafka source topic name or a Java supported regular expression for the Kafka source topic name pattern to read the events from. Enter a string of up to 249 characters.
consumerProperties	String	-	Provide a comma-separated list of optional consumer configuration properties. Specify the values as key-value pairs. For example, `key1=value1, key2=value2`. You can enter a string of up to 4000 characters.
producerProperties	String	-	The configuration properties for the producer. Provide a comma-separated list and specify the values as key-value pairs. You can enter a string of up to 4000 characters.
mdFetchTimeout	Integer	-	The time after which the metadata is not fetched. Enter a value between 1 and 2147483647.
batchSize	Integer	-	The batch size of the events after which a streaming ingestion task writes data to the target. Enter a value between 1 and 2147483647.
Topic Name/ Expression	String	Yes	Kafka topic name or a Java supported regular expression for the Kafka topic name pattern. You can enter a string of up to 249 characters.

POST request example

To update a streaming ingestion task with a Kafka source and target, you might send a request similar to the following example:

```
{
  "name": "kafka to kafka",
  "description": "kafka to kafka",
  "runtimeId": "01000025000000000003",
  "locationId": "5sJ0JDyJyWLlrosS5qJjsQ",
  "currentVersion": "2",
  "messageFormat": "binary",
  "nodes": [
    {
      "name": "kafka to kafka_source",
      "type": "source",
      "connectionId": "012MGS0B000000000002",
      "transformationType": "",
      "config": [
        {
          "key": "consumerProperties",
          "value": "key=value"
        },
        {
          "key": "Topic",
          "value": "test"
        }
      ]
    },
```

```
            {
              "name": "kafka to kafka_target",
              "type": "target",
              "connectionId": "012MGS0B000000000002",
              "transformationType": "",
              "config": [
                {
                  "key": "producerProperties",
                  "value": "key=value"
                },
                {
                  "key": "mdFetchTimeout",
                  "value": 5000
                },
                {
                  "key": "batchSize",
                  "value": 1048576
                },
                {
                  "key": "Topic Name/Expression",
                  "value": "test"
                }
              ]
            }
          ],
          "edges": [
            {
              "from": "kafka to kafka_source",
              "to": "kafka to kafka_target"
            }
          ]
        }
```

POST response

When the REST API successfully performs an action, it returns a 200 or 201 success response. When the REST API encounters an error, it returns an appropriate error code.

If the request is successful, the response returns the following fields:

Field	Type	Description
name	String	Name of the task.
description	String	Description of the task, if available.
runtimeId	String	ID of the runtime environment.
currentVersion	String	The latest dataflow object version.
nodes	Array	Details of the task source and target connections.

Fields of the nodes array

The response includes the following fields in the nodes array:

Field	Type	Description
name	String	Name of the connection.
type	String	The connection type.

Field	Type	Description
connectionId	String	ID of the connection.
transformationType	String	The type of transformation.
config	String	Configuration of the source and target connections in key-value pairs. The keys in the array depend on the type of source and target connections.

If the request is unsuccessful, the response includes a reason for the failure.

Configuration information in the config array MQTT as a source

If the request is successful, the response returns the following fields:

Key	Type	Description
ClientID	String	Unique identifier that identifies the connection between the MQTT source and the MQTT broker. The client ID is the file-based persistence store that the MQTT source uses to store messages when they are being processed.
MaxQueueSize	Integer	The maximum number of messages that the processor can store in memory.
Topic	String	Name of the MQTT topic.

If the request is unsuccessful, the response includes a reason for the failure.

POST response example

If the request is successful, you might receive a response similar to the following example in a `Success` node:

```
{
  "Success": {
    "name": "mqtt to flatfile",
    "description": "mqtt to flatfile",
    "runtimeId": "01000025000000000003",
    "locationId": "5sJ0JDyJyWLlrosS5qJjsQ",
    "currentVersion": "2",
    "messageFormat": "binary",
    "nodes": [
      {
        "name": "mqtt to flatfile_source",
        "type": "source",
        "connectionId": "012MGS0B000000000001O",
        "transformationType": "",
        "config": [
          {
            "key": "ClientID",
            "value": "test"
          },
          {
            "key": "MaxQueueSize",
            "value": 1024
          },
          {
            "key": "Topic",
            "value": "test"
          }
        ]
      },
      {
```

```
            "name": "mqtt to flatfile_target",
            "type": "target",
            "connectionId": "012MGS0B00000000002N",
            "transformationType": "",
            "config": [
              {
                "key": "interimDirectory",
                "value": "/home/agent/test"
              },
              {
                "key": "rolloverSize",
                "value": 1024
              },
              {
                "key": "rolloverEvents",
                "value": 100
              },
              {
                "key": "rolloverTime",
                "value": 300000
              },
              {
                "key": "File Name",
                "value": "test"
              }
            ]
          }
        ],
        "edges": [
          {
            "from": "mqtt to flatfile_source",
            "to": "mqtt to flatfile_target"
          }
        ]
      }
    }
  }
```

Configuration information in the config array for JMS as a source

The response returns only the fields that you entered in the request.

If the request is successful, the response returns the following fields:

Key	Type	Description
destinationType	String	Type of destination that the source service sends JMS messages to.
clientId	String	Unique ID of the JMS connection.
sharedSubscription	String	Enables multiple consumers to access a single subscription. Applies to the TOPIC destination type.
durableSubscription	String	The JMS source service enables inactive subscribers to retain messages and then deliver them when the subscriber reconnects. Applies to the TOPIC destination type.
subscriptionName	String	Name of the subscription. Applies to the TOPIC destination type, when the topic subscription type is shared, durable, or both.
JMS Destination	String	Name of the queue or topic that the JMS provider delivers messages to.

If the request is unsuccessful, the response includes a reason for the failure.

POST response example

If the request is successful, you might receive a response similar to the following example in a `Success` node:

```json
{
  "Success": {
    "name": "crud",
    "description": "JMS to FileToFile",
    "runtimeId": "01000025000000000003",
    "locationId": "5sJ0JDyJyWLlrosS5qJjsQ",
    "currentVersion": "2",
    "messageFormat": "binary",
    "nodes": [
      {
        "name": "crud_source",
        "type": "source",
        "connectionId": "012MGS0B000000000003",
        "transformationType": "",
        "config": [
          {
            "key": "destinationType",
            "value": "QUEUE"
          },
          {
            "key": "clientId",
            "value": ""
          },
          {
            "key": "JMS Destination",
            "value": "test"
          }
        ]
      },
      {
        "name": "crud_target",
        "type": "target",
        "connectionId": "012MGS0B00000000000H",
        "transformationType": "",
        "config": [
          {
            "key": "interimDirectory",
            "value": "/home/agent/test"
          },
          {
            "key": "rolloverSize",
            "value": 1024
          },
          {
            "key": "rolloverEvents",
            "value": 100
          },
          {
            "key": "rolloverTime",
            "value": 300000
          },
          {
            "key": "File Name",
            "value": "test"
          }
        ]
      }
    ],
    "edges": [
      {
        "from": "crud_source",
        "to": "crud_target"
      }
    ]
  }
}
```

Configuration information in the config array for ADLS Gen2 as a target

The response returns only the fields that you entered in the request.

If the request is successful, the response returns the following fields:

Key	Type	Description
writeStrategy	String	The action to take when a file exists in the ADLS Gen2 storage.
rolloverSize *	Integer	Target file size, in KB, at which to trigger rollover. Applies to a Rollover write strategy.
rolloverEvents *	Integer	Number of events or messages to accumulate before a rollover. Applies to a Rollover write strategy.
rolloverTime *	Integer	Length of time, in milliseconds, after which to trigger a rollover. Applies to a Rollover write strategy.
filesystemNameOverride	String	Overrides the default file system name provided in the connection. This file system name is used write to a file at run time.
directoryOverride	String	Overrides the default directory path. The ADLS Gen2 directory path to write data to. If left blank, the default directory path is used.
compressionFormat	String	Compression format to use before the streaming ingestion task writes data to the target file.
File Name/Expression	String	ADLS Gen2 file name or a regular expression.
* Enter a value for at least one of the fields.		

If the request is unsuccessful, the response includes a reason for the failure.

POST response example

If the request is successful, you might receive a response similar to the following example in a `Success` node:

```
{
  "Success": {
    "name": "flatfile to adls",
    "description": "flatfile to adls",
    "runtimeId": "01000025000000000003",
    "locationId": "5sJ0JDyJyWLlrosS5qJjsQ",
    "currentVersion": "2",
    "messageFormat": "binary",
    "nodes": [
      {
        "name": "flatfile to adls_source",
        "type": "source",
        "connectionId": "012MGS0B00000000002N",
        "transformationType": "",
        "config": [
          {
            "key": "File",
            "value": "logfile"
          },
          {
            "key": "initialPosition",
            "value": "Current Time"
          },
          {
            "key": "rolloverPattern",
            "value": "test"
```

```
      },
      {
        "key": "tailingMode",
        "value": "Single file"
      }
    ]
  },
  {
    "name": "flatfile to adls_target",
    "type": "target",
    "connectionId": "012MGS0B00000000003D",
    "transformationType": "",
    "config": [
      {
        "key": "writeStrategy",
        "value": "Rollover"
      },
      {
        "key": "filesystemNameOverride",
        "value": "test"
      },
      {
        "key": "File Name/Expression",
        "value": "test"
      },
      {
        "key": "compressionFormat",
        "value": "NONE"
      },
      {
        "key": "directoryOverride",
        "value": "/test"
      },
      {
        "key": "interimDirectory",
        "value": "/home/agent/test"
      },
      {
        "key": "rolloverSize",
        "value": 1024
      },
      {
        "key": "rolloverEvents",
        "value": 100
      },
      {
        "key": "rolloverTime",
        "value": 300000
      }
    ]
  }
    ]
  }
}
```

Configuration information in the config array for Amazon S3 as a target

The response returns only the fields that you entered in the request.

If the request is successful, the response returns the following fields:

Key	Type	Description
partitionTime	String	The time interval according to which the streaming ingestion task creates partitions in the Amazon S3 bucket.
minUploadPartSize	Integer	Minimum part size when uploading a large file as a set of multiple independent parts, in megabytes. Use this property to tune the file load to Amazon S3.
multipartUploadThreshold	Integer	Multipart threshold when uploading objects in multiple parts in parallel.
Object Name/Expression	String	Amazon S3 target file name or a regular expression for the Amazon S3 file name pattern.

If the request is unsuccessful, the response includes a reason for the failure.

POST response example

If the request is successful, you might receive a response similar to the following example in the `Success` node:

```
{
    "Success": {
    "name": "flatfile to amazon S3",
    "description": "flatfile to amazon S3",
    "runtimeId": "01000025000000000003",
    "locationId": "5sJ0JDyJyWLlrosS5qJjsQ",
    "currentVersion": "2",
    "messageFormat": "binary",
    "nodes": [
      {
        "name": "flatfile to amazon S3_source",
        "type": "source",
        "connectionId": "012MGS0B00000000002N",
        "transformationType": "",
        "config": [
          {
            "key": "File",
            "value": "logfile"
          },
          {
            "key": "initialPosition",
            "value": "Current Time"
          },
          {
            "key": "rolloverPattern",
            "value": "test"
          },
          {
            "key": "tailingMode",
            "value": "Single file"
          }
        ]
      },
      {
        "name": "flatfile to amazon S3_target",
        "type": "target",
        "connectionId": "012MGS0B0000000000I7",
        "transformationType": "",
        "config": [
          {
            "key": "partitionTime",
            "value": "None"
          },
          {
            "key": "minUploadPartSize",
```

```
            "value": 5120
          },
          {
            "key": "multipartUploadThreshold",
            "value": 5120
          },
          {
            "key": "Object Name/Expression",
            "value": "test"
          }
        ]
      }
    ],
    "edges": [
      {
        "from": "flatfile to amazon S3_source",
        "to": "flatfile to amazon S3_target"
      }
    ]
  }
}
```

Configuration information in the config array for Azure Event Hubs as a target

The response returns only the fields that you entered in the request.

If the request is successful, the response returns the following fields:

Key	Type	Description
sasPolicyName	String	The name of the Event Hub Namespace Shared Access Policy.
sasPolicyPrimaryKey	String	The primary key of the Event Hub Namespace Shared Access Policy.
Event Hub	String	The name of the Azure Event Hubs.

If the request is unsuccessful, the response includes a reason for the failure.

POST response example

If the request is successful, you might receive a response similar to the following example in a `Success` node:

```
{
  "Success": {
    "name": "flatfile to azure event hub",
    "description": "flatfile to azure event hub",
    "runtimeId": "01000025000000000003",
    "locationId": "5sJ0JDyJyWLlrosS5qJjsQ",
    "currentVersion": "2",
    "messageFormat": "binary",
    "nodes": [
      {
        "name": "flatfile to azure event hub_source",
        "type": "source",
        "connectionId": "012MGS0B00000000002N",
        "transformationType": "",
        "config": [
          {
            "key": "File",
            "value": "logfile"
          },
          {
            "key": "initialPosition",
            "value": "Current Time"
          },
          {
```

```
                "key": "rolloverPattern",
                "value": "test"
              },
              {
                "key": "tailingMode",
                "value": "Single file"
              }
            ]
          },
          {
            "name": "flatfile to azure event hub_target",
            "type": "target",
            "connectionId": "012MGS0B000000000001S",
            "transformationType": "",
            "config": [
              {
                "key": "sasPolicyName",
                "value": "test"
              },
              {
                "key": "sasPolicyPrimaryKey",
                "value": "test"
              },
              {
                "key": "Event Hub",
                "value": "test"
              }
            ]
          }
        ],
        "edges": [
          {
            "from": "flatfile to azure event hub_source",
            "to": "flatfile to azure event hub_target"
          }
        ]
      }
    }
  }
```

Configuration information in the config array for JDBC as a target

The response returns only the fields that you entered in the request.

If the request is successful, the response returns the following field:

Key	Type	Description
Table Name	String	Name of the table to insert data to in JSON format.

If the request is unsuccessful, the response includes a reason for the failure.

POST response example

If the request is successful, you might receive a response similar to the following example in a `Success` node:

```
{
  "Success": {
    "name": "FileFile to jdbc",
    "description": "FileToFile to jdbc_target",
    "runtimeId": "01000025000000000003",
    "locationId": "5sJ0JDyJyWLlrosS5qJjsQ",
    "currentVersion": "2",
    "messageFormat": "binary",
    "nodes": [
      {
        "name": "flatfile to jdbc_source",
        "type": "source",
```

```
            "connectionId": "012MGS0B00000000002N",
            "transformationType": "",
            "config": [
              {
                "key": "initialPosition",
                "value": "Current Time"
              },
              {
                "key": "tailingMode",
                "value": "Single file"
              },
              {
                "key": "rolloverPattern",
                "value": "test"
              },
              {
                "key": "File",
                "value": "logfile"
              }
            ]
          },
          {
            "name": "flatfile to jdbc_target",
            "type": "target",
            "connectionId": "012MGS0B0000000000KF",
            "transformationType": "",
            "config": [
              {
                "key": "Table Name",
                "value": "table"
              }
            ]
          }
        ],
        "edges": [
          {
            "from": "flatfile to jdbc_source",
            "to": "flatfile to jdbc_target"
          }
        ]
      }
    }
  }
```

Configuration information in the config array for Amazon Kinesis Streams as a source and as a target

The response returns only the fields that you entered in the request.

If the request is successful, the response returns the following fields:

Key	Type	Description
appendGUID	Boolean	Specifies whether or not to add a GUID as a suffix to the Amazon DynamoDB table name.
dynamoDB	String	Amazon DynamoDB table name where to store the checkpoint details of the Kinesis source data.

Key	Type	Description
Stream	String	Name of the Kinesis Stream from where to read data. Applies when you use Amazon Kinesis Streams as a source.
Stream Name/ Expression	String	Kinesis stream name or a regular expression for the Kinesis stream name pattern. Applies when you use Amazon Kinesis Streams as a target.

If the request is unsuccessful, the response includes a reason for the failure.

POST response example

If the request is successful, you might receive a response similar to the following example in a `Success` node:

```
{
    "Success": {
    "name": "kinesis to kinesis",
    "description": "kinesis to kinesis",
    "runtimeId": "01000025000000000003",
    "locationId": "5sJ0JDyJyWLlrosS5qJjsQ",
    "currentVersion": "2",
    "messageFormat": "binary",
    "nodes": [
      {
        "name": "kinesis to kinesis_source",
        "type": "source",
        "connectionId": "012MGS0B00000000000F",
        "transformationType": "",
        "config": [
          {
            "key": "appendGUID",
            "value": true
          },
          {
            "key": "dynamoDB",
            "value": "table"
          },
          {
            "key": "Stream",
            "value": "test"
          }
        ]
      },
      {
        "name": "kinesis to kinesis_target",
        "type": "target",
        "connectionId": "012MGS0B00000000000F",
        "transformationType": "",
        "config": [
          {
            "key": "Stream Name/Expression",
            "value": "trgt"
          }
        ]
      }
    ],
    "edges": [
      {
        "from": "kinesis to kinesis_source",
        "to": "kinesis to kinesis_target"
      }
    ]
  }
}
```

Configuration information in the config array for flat file as a source and as a target

The response returns only the fields that you entered in the request.

If the request is successful, the response returns the following fields:

Key	Type	Required	Description
File	String	Yes	Absolute path and name of the source file you want to read.
initialPosition	String	Yes	Starting position to read data from the file to tail.
rolloverPattern	String	-	File name pattern for the file that rolls over.
tailingMode	String	Yes	Tail a file or multiple files based on the logging pattern.
File Name	String	Yes	The name of the target file.
interimDirectory	String	Yes	Path to the staging directory on the Secure Agent.
rolloverSize	Integer	Yes	The file size, in KB, at which the task moves the file from the staging directory to the target.
rolloverEvents	Integer	Yes	Number of events or messages to accumulate before a file rollover.
rolloverTime	Integer	-	Length of time, in milliseconds, after which the target file rolls over.

If the request is unsuccessful, the response includes a reason for the failure.

POST response example

If the request is successful, you might receive a response similar to the following example:

```
{
  "Success": {
    "name": "FileToFile",
    "description": "FileToFile_V2",
    "runtimeId": "01000025000000000003",
    "locationId": "5sJ0JDyJyWLlrosS5qJjsQ",
    "currentVersion": "2",
    "messageFormat": "binary",
    "nodes": [
      {
        "name": "FileToFile_source",
        "type": "source",
        "connectionId": "0100000B000000000002",
        "transformationType": "",
        "config": [
          {
            "key": "File",
            "value": "siagent.log"
          },
          {
            "key": "initialPosition",
            "value": "Current Time"
          },
          {
            "key": "rolloverPattern",
            "value": ""
          },
          {
            "key": "tailingMode",
            "value": "Single file"
```

```
            }
          ]
        },
        {
          "name": "FileToFile_target",
          "type": "target",
          "connectionId": "0100000B000000000002",
          "transformationType": "",
          "config": [
            {
              "key": "File Name",
              "value": "testing.log"
            },
            {
              "key": "interimDirectory",
              "value": "/home/agent/infa/test_file_target"
            },
            {
              "key": "rolloverSize",
              "value": 100
            },
            {
              "key": "rolloverEvents",
              "value": 100
            },
            {
              "key": "rolloverTime",
              "value": 100
            }
          ]
        }
      ],
      "edges": [
        {
          "from": "FileToFile_source",
          "to": "FileToFile_target"
        }
      ],
      "runtimeOptions": {
        "maxLogSize": {
          "value": 10,
          "unit": "MB"
        },
        "logLevel": "INFO"
      }
    }
  }
}
```

Configuration information in the config array for Kafka as a source and as a target

The response returns only the fields that you entered in the request.

If the request is successful, the response returns the following fields:

Key	Type	Description
Topic	String	Kafka source topic name or a Java supported regular expression for the Kafka source topic name pattern to read the events from.
consumerProperties	String	A comma-separated list of optional consumer configuration properties.
producerProperties	String	The configuration properties for the producer.

Key	Type	Description
mdFetchTimeout	Integer	The time after which the metadata is not fetched.
batchSize	Integer	The batch size of the events after which a streaming ingestion task writes data to the target.
Topic Name/Expression	String	Kafka topic name or a Java supported regular expression for the Kafka topic name pattern.

If the request is unsuccessful, the response includes a reason for the failure.

POST response example

If the request is successful, you might receive a response similar to the following example in a `Success` node:

```
{
  "Success": {
    "name": "kafka to kafka",
    "description": "kafka to kafka",
    "runtimeId": "01000025000000000003",
    "locationId": "5sJ0JDyJyWLlrosS5qJjsQ",
    "currentVersion": "2",
    "messageFormat": "binary",
    "nodes": [
      {
        "name": "kafka to kafka_source",
        "type": "source",
        "connectionId": "012MGS0B000000000002",
        "transformationType": "",
        "config": [
          {
            "key": "consumerProperties",
            "value": "key=value"
          },
          {
            "key": "Topic",
            "value": "test"
          }
        ]
      },
      {
        "name": "kafka to kafka_target",
        "type": "target",
        "connectionId": "012MGS0B000000000002",
        "transformationType": "",
        "config": [
          {
            "key": "producerProperties",
            "value": "key=value"
          },
          {
            "key": "mdFetchTimeout",
            "value": 5000
          },
          {
            "key": "batchSize",
            "value": 1048576
          },
          {
            "key": "Topic Name/Expression",
            "value": "test"
          }
        ]
      }
    ],
    "edges": [
```

```
            {
              "from": "kafka to kafka_source",
              "to": "kafka to kafka_target"
            }
          ]
        }
      }
```

jobs resource

Use the jobs resource to get the details of a streaming ingestion job.

GET request

To request the details of a streaming ingestion job, use the following URL:

```
<server URI>/sisvc/monitor/v1/jobs/<dataflow ID>/<run ID of the job>
```

GET request example

To request the details of a streaming ingestion job, you might send a request similar to the following example:

```
POST https://usw1-ing.dm2-us.informaticacloud.com/sisvc/monitor/v1/jobs/
1948938e-3923-4602-aba8-f122e3d66faf/42559
Content-Type: application/json
Accept:application/json
IDS-SESSION-ID:2l0oeVx22Rujiej7yTokmT
```

GET response

Returns the jobs object if successful or an error object if an error occurs.

If successful, the response includes the following information about a streaming ingestion job:

Parameter	Type	Description
assetId	String	ID of the streaming ingestion job.
assetName	String	Name of the streaming ingestion job.
duration	Integer	The time it took to deploy the job.
endTime	Integer	End time of deploying the job, in UTC time.
startTime	Integer	Start time of deploying the job, in UTC time.
extraData	String	Additional information including the task ID, deployed version, and the Secure Agent group ID.
runId	Integer	Run ID of the streaming ingestion job. The ID changes for every deployment.
orgId	String	ID of the organization the logged in user belongs to.
runtimeEnv	String	ID of the Secure Agent that deployed the streaming ingestion job.

Parameter	Type	Description
startedBy	String	Name of the user who created the streaming ingestion task.
status	String	The status of the streaming ingestion job. A job can be in one of the following status: - Deploying. The job is being deployed. - Up and Running. The job is running. - Running with Warning. The job is running with warnings. - Running with Error. The job is running with error. - Undeployed. The job is undeployed. - Stopped. The job was intentionally stopped.

GET response example

If the request to get the details of a streaming ingestion job is successful, you might receive a response similar to the following example:

```
{
    "assetId": "1948938e-3923-4602-aba8-f122e3d66faf",
    "assetName": "testmonitor",
    "assetType": "SI_DATAFLOW",
    "correlationId": null,
    "duration": 1543,
    "endTime": "2022-02-14T04:04:13.000+0000",
    "extraData": "{\"id\":\"0RwiUUb9bVwjL67dWOKjoI\",\"version\":1,\"agentGroupId
\":null}",
    "location": "Default",
    "runId": 42559,
    "orgId": "2lFy0UUNnlnbjhaoT3TSqw",
    "runtimeEnv": "011ZFB2500000000000N",
    "startedBy": "siqa_new",
    "status": "Undeployed",
    "startTime": "2022-02-14T03:38:30.000+0000",
    "deployedVersion": 1
}
```

MIJobs resource

Use the MIJobs resource to get a list of the available streaming ingestion jobs.

GET request

To request a list of the available streaming ingestion jobs, use the following URL:

```
<server URI>/mijobmonitor/api/v1/MIJobs
```

You can include the following query parameters in the URI:

Parameter	Type	Required	Description
$count	Boolean	No	Displays the number of ingestion jobs in the database.
$filter	String	No	Filters the job based on the input. You can filter using one of the following fields: - assetName - assetType - startedBy - status You can filter jobs using single or multiple fields.
$orderby	String	No	Sorts the order of the jobs. You can sort the jobs using the following fields: - assetName - assetType - status - runtimeEnv - startTime You can sort jobs using single or multiple fields.
$skip	Integer	No	Skips the number of streaming ingestion jobs that you specify. For example, you might want to skip the first five streaming ingestion jobs. Consider the $filter and $orderby parameter values, if specified.
$top	Integer	No	Displays the number of top streaming ingestion jobs that you specify. For example, you might want to view the top ten streaming ingestion jobs. Consider the $filter and $orderby parameter values, if specified.

GET request example

To get a list of the available streaming ingestion jobs, you might send a request similar to the following example:

```
POST https://usw1-ing.dm2-us.informaticacloud.com/mijobmonitor/api/v1/MIJobs?$count=true&
$filter=(startedBy eq 'siqa_new')&$orderby=deployTime desc&$skip=0&$top=25
Content-Type: application/json
Accept:application/json
IDS-SESSION-ID:2l0oeVx22Rujiej7yTokmT
```

GET response

Returns the MIjobs object if successful or an error object if an error occurs.

If successful, the response includes the following information about the streaming ingestion job:

Parameter	Type	Description
assetName	String	Name of the streaming ingestion job.
runId	Integer	Run ID of the streaming ingestion job. The ID changes for every deployment.
orgId	String	ID of the organization the logged in user belongs to.
runtimeEnv	String	ID of the Secure Agent that deployed the streaming ingestion job.
startTime	Integer	Date and start time of deploying the job, in UTC time.

Parameter	Type	Description
endTime	Integer	Date and end time of deploying the job, in UTC time.
deployTime	Integer	Date and time of deploying the job, in UTC time.
undeployTime	Integer	Date and time of undeploying the job, in UTC time.
startedBy	Integer	Name of the user who created the streaming ingestion task.
status	String	The status of the streaming ingestion job. A job can be in one of the following status: - Deploying. The job is being deployed. - Up and Running. The job is running. - Running with Warning. The job is running with warnings. - Running with Error. The job is running with error. - Undeployed. The job is undeployed. - Stopped. The job was intentionally stopped.
extraData	String	Additional information including the task ID, the location of the streaming ingestion job, and the Secure Agent ID.

GET response example

If the request to get a list of available streaming ingestion jobs is successful, you might receive a response similar to the following example:

```
{
    "@odata.context": "$metadata#Collection(OData.MI.JobMonitor.MIJob)",
    "@odata.count": 421,
    "value": [
        {
            "assetId": "7ce6bbc7-f0e2-4278-bd6d-d1187f4a1420",
            "assetName": "SIdeployJms",
            "assetType": "SI_DATAFLOW",
            "runId": 33015,
            "duration": 300000,
            "orgId": "1Pm6cSfPcAqfgeV57Fn3u4",
            "runtimeEnv": "011U5M08000000000003",
            "startTime": "2021-04-29T13:09:48.000+0000",
            "endTime": "2021-04-29T13:14:48.000+0000",
            "deployTime": "2021-04-29T13:09:48.000+0000",
            "undeployTime": "2021-04-29T13:14:48.000+0000",
            "startedBy": "siqa_new",
            "status": "Undeployed",
            "outOfSync": true,
            "extraData": "{\"taskId\":\"7Z4ZZjXc9QViT4t2okiHuz\",\"runtimeEnv
\":\"011U5M25000000000002\",\"location\":\"RestAutomation\"}",
            "deployedVersion": 1,
            "replace": null,
            "lastUpdateTime": 0
        },
        {
            "assetId": "a03b9aa1-4a4a-47ee-808d-ddc0ee7b3a4a",
            "assetName": "kafka to kafka test",
            "assetType": "SI_DATAFLOW",
            "runId": 33527,
            "duration": 204988000,
            "orgId": "1Pm6cSfPcAqfgeV57Fn3u4",
            "runtimeEnv": "011U5M08000000000002",
            "startTime": "2021-05-04T05:41:39.000+0000",
            "endTime": "2021-05-06T14:38:07.000+0000",
            "deployTime": "2021-05-04T05:41:39.000+0000",
            "undeployTime": "2021-05-06T14:38:07.000+0000",
            "startedBy": "siqa_new",
            "status": "Undeployed",
```

```
            "outOfSync": true,
            "extraData": "{\"taskId\":\"8V21nib7Sqgiw3QoDRi5uK\",\"runtimeEnv
\":\"011U5M25000000000002\",\"location\":\"Default\"}",
            "deployedVersion": 1,
            "replace": null,
            "lastUpdateTime": 0
        }
    ]
}
```

status resource

Use the status resource to get the status of a streaming ingestion job.

GET request

To request the status of a streaming ingestion job, use the following URL:

```
<server URI>/sisvc/monitor/v1/status/dataflows/<dataflow ID>
```

GET request example

To get the status of a streaming ingestion job, you might send a request similar to the following example:

```
POST https://usw1-ing.dm2-us.informaticacloud.com/sisvc/monitor/v1/status/dataflows/
1948938e-3923-4602-aba8-f122e3d66faf
Content-Type: application/json
Accept:application/json
IDS-SESSION-ID:2l0oeVx22Rujiej7yTokmT
```

GET response

Returns the job status object if successful or an error object if an error occurs.

If successful, the response includes the following information about the status of a streaming ingestion job:

Parameter	Type	Description
dataflowName	String	Name of the streaming ingestion job.
dataflowId	Integer	ID of the streaming ingestion job.
status	String	The status of the streaming ingestion job. A job can be in one of the following status: - Deploying. The job is being deployed. - Up and Running. The job is running. - Running with Warning. The job is running with warnings. - Running with Error. The job is running with error. - Undeployed. The job is undeployed. - Stopped. The job was intentionally stopped.
timestamp	Integer	Time, in milliseconds, when the Secure Agent records the status of the streaming ingestion job.
reports	Array	Status details of each node.

Parameter	Type	Description
graph	String	The throughput information for the source and target of the job.
runId	Integer	Run ID of the streaming ingestion job. The ID changes for every deployment.

GET response example

If the request to get the status of a streaming ingestion job is successful, you might receive a response similar to the following example:

```
{
    "dataflowName": "testmonitor",
    "dataflowId": "1948938e-3923-4602-aba8-f122e3d66faf",
    "status": "Running",
    "timestamp": 1644839755000,
    "reports": [
        {
            "name": "testmonitor_testmonitor_source",
            "id": "a5684428-f41f-4d24-b73f-33c232314a91",
            "status": "Running",
            "timestamp": 1644839756000,
            "message": null
        },
        {
            "name": "testmonitor_testmonitor_target",
            "id": "4f59b5fb-b5b2-4b83-994b-0d3e56f67e22",
            "status": "Running",
            "timestamp": 1644839756000,
            "message": null
        }
    ],
    "graph": "{\"agentId\":\"011ZFB0800000000000N\",\"nodes\":[{\"id\":\"a5684428-
f41f-4d24-b73f-33c232314a91\",\"name\":\"testmonitor_source\",\"serviceType\":\"source
\",\"config\":[{\"key\":\"_nativeName\",\"value\":\"src\"},{\"key\":\"consumerProperties
\",\"value\":null}],\"connectionId\":\"011ZFB0B0000000000KJ\",\"type
\":\"\",\"metaMetadata\":\"\"},{\"id\":\"4f59b5fb-b5b2-4b83-994b-0d3e56f67e22\",\"name
\":\"testmonitor_target\",\"serviceType\":\"target\",\"config\":[{\"key\":\"_nativeName
\",\"value\":\"trgt\"},{\"key\":\"batchSize\",\"value\":\"1048576\"},{\"key
\":\"mdFetchTimeout\",\"value\":\"5000\"},{\"key\":\"producerProperties\",\"value
\":null}],\"connectionId\":\"011ZFB0B0000000000KJ\",\"type\":\"\",\"metaMetadata
\":\"\"}],\"edges\":[{\"id\":\"6ae185ea-7e6e-4bf6-bd9e-0be5ef3a8e78\",\"name
\":\"testmonitor_source_testmonitor_target\",\"from\":\"testmonitor_source\",\"to
\":\"testmonitor_target\",\"type\":\"success\",\"config\":[],\"metaMetadata
\":\"\"}],\"runtimeOptions\":null}",
    "version": 1,
    "runId": 42563
}
```

statistics resource

Use the statistics resource to get the statistics of a streaming ingestion job.

The streaming ingestion job should be in one of the following status before you can view its statistics:

- Deploying
- Up and Running
- Running with Warning

- Running with Error
- Stopped

GET request

To request the statistics of a streaming ingestion job, use the following URL:

```
<server URI>/sisvc/monitor/v1/statistics/dataflows/<dataflow ID>
```

You can include the following query parameters in the URI:

Parameter	Type	Required	Description
intervals	Integer	Yes	Time, in seconds, to display statistics for a streaming ingestion job. For example, if you specify 30 seconds, the response displays job statistics for the last 30 seconds.
overall	Boolean	No	Displays the statistics from the time the job is deployed.

GET request example

To request the statistics of a streaming ingestion job, you might send a request similar to the following example:

```
POST https://usw1-ing.dm2-us.informaticacloud.com/sisvc/monitor/v1/statistics/
dataflows/7f1daca9-3983-4677-930f-a9529802c56b?intervals=30&overall=true
Content-Type: application/json
Accept:application/json
IDS-SESSION-ID:2l0oeVx22Rujiej7yTokmT
```

GET response

Returns the statistics object if successful or an error object if an error occurs.

If successful, the response includes the following information about the statistics of a streaming ingestion job:

Parameter	Type	Description
dataflowId	String	ID of the streaming ingestion job.
dataflowRunId	Integer	Run ID of the streaming ingestion job.
startTime	Integer	Start time of the streaming ingestion job, in milliseconds.
stopTime	Integer	Stop time of the streaming ingestion job, in milliseconds.
inMessages	Integer	The number of messages that arrive at a node. A node is a source, transformation, or target, that is used in the streaming ingestion task. The value is zero for a source node.
outMessages	Integer	The number of messages that transfer from a node. The value is zero for a target node.
inBytes	Integer	The total size of incoming messages in bytes. The value is zero for a source node.

Parameter	Type	Description
outBytes	Integer	The total size of outgoing messages in bytes. The value is zero for a target node.
nodes	Array	Information about streaming data in the source and the target used in the task.
intervals	Integer	The statistics of the job for the time interval you specify in the request. Applies when you set an interval.

GET response example

If the request to get the statistics of a streaming ingestion job is successful, you might receive a response similar to the following example:

```
{
    "dataflowId": "7f1daca9-3983-4677-930f-a9529802c56b",
    "dataflowName": "newnew",
    "dataflowVersion": 1,
    "dataflowRunId": 54231,
    "snapshotCount": 171,
    "overall": {
        "dataflowId": "7f1daca9-3983-4677-930f-a9529802c56b",
        "dataflowName": "newnew",
        "dataflowVersion": 1,
        "dataflowRunId": 54231,
        "traits": {},
        "interval": 6007,
        "startTime": 1646649995000,
        "stopTime": 1646656000000,
        "nodes": [
            {
                "name": "newnew_newnew_source",
                "id": "17a51cdf-1f27-481e-81b8-d2e8ff60ec28",
                "inMessages": 0,
                "outMessages": 0,
                "inBytes": 0,
                "outBytes": 0,
                "nodeType": "Unknown"
            },
            {
                "name": "newnew_newnew_target",
                "id": "c30d6db4-6a3b-40d3-adfb-88779a972098",
                "inMessages": 0,
                "outMessages": 0,
                "inBytes": 0,
                "outBytes": 0,
                "nodeType": "Unknown"
            }
        ]
    },
    "intervals": {
        "30": {
            "dataflowId": "7f1daca9-3983-4677-930f-a9529802c56b",
            "dataflowName": "newnew",
            "dataflowVersion": null,
            "dataflowRunId": 54231,
            "traits": {},
            "interval": 30,
            "startTime": 1646655972683,
            "stopTime": 1646656002683,
            "nodes": []
        }
    }
}
```

history resource

Use the history resource to get the history of a streaming ingestion job.

GET request

To request the history of a streaming ingestion job, use the following URL:

```
<server URI>/sisvc/monitor/v1/history/dataflows/<dataflow ID>
```

GET request example

To get the history of a streaming ingestion job, you might send a request similar to the following example:

```
POST https://usw1-ing.dm2-us.informaticacloud.com/siscv/monitor/v1/history/dataflows/
1948938e-3923-4602-aba8-f122e3d66faf
Content-Type: application/json
Accept:application/json
IDS-SESSION-ID:2l0oeVx22Rujiej7yTokmT
```

GET response

Returns the job history object if successful or an error object if an error occurs.

If successful, the response includes the following information about the history of a streaming ingestion job:

Parameter	Type	Description
dataflowName	String	Streaming ingestion job name.
dataflowId	Integer	Streaming ingestion job ID.
deployedAt	Integer	The start time of deploying the job, in UTC time.
undeployedAt	Integer	The time when the job finished undeploying, in UTC.
runID	Integer	Run ID of the streaming ingestion job. The ID changes for every deployment.

GET response example

If the request to get the history of a streaming ingestion job is successful, you might receive a response similar to the following example:

```
[
    {
        "dataflowId": "1948938e-3923-4602-aba8-f122e3d66faf",
        "dataflowName": "testmonitor",
        "deployedAt": 1644809910000,
        "undeployedAt": 1644811453000,
        "dataflowVersion": 1,
        "runId": 42559,
        "overall": null,
        "intervals": {},
        "graph": null
    },
    {
        "dataflowId": "1948938e-3923-4602-aba8-f122e3d66faf",
        "dataflowName": "testmonitor",
        "deployedAt": 1644811513000,
        "undeployedAt": 1644838813000,
        "dataflowVersion": 1,
        "runId": 42561,
        "overall": null,
        "intervals": {},
        "graph": null
```

```
        }
    ]
```

CHAPTER 7

Model Serve REST API

Use the Model Serve REST API resources to create and deploy Model Serve assets.

When you use Model Serve resources, note the following rules:

- Use JSON or XML format.
- The base URL that you use depends on the request that you make. Use one of the following options:
  - To generate predictions from a quick start model, use the predictUrl value from the request for information about a single quick start model.
  - To generate predictions from a user-defined machine learning model, use the predictUrl value from the request to monitor a model deployment.
  - To make any other request, use the serverUrl value from the login response. For example:

    ```
    https://na4.dm-us.informaticacloud.com/saas
    ```
- Use the following request header format:

  ```
  <METHOD> <serverUrl>/<URI> HTTP/<HTTP version>
  Content-Type: application/json
  Accept: application/json
  icSessionId: <SessionId>
  ```

Note: If you use a tool such as Postman that automatically includes the HTTP version, do not enter the HTTP version in the URL. If the HTTP version appears twice in the URL, the request fails.

Quick start models

Use API resources to get descriptions and status information about quick start models.

Getting descriptions of quick start models

Use the modelhub resource to get a list of the names and descriptions of all quick start models.

GET request

To get information about the quick start models, use the following URI:

```
/mlops/api/v1/modelhub/models
```

GET response

Returns an overview of each quick start model.

The following table describes the fields in each quick start model overview:

Field	Type	Description
displayName	String	External name of the quick start model that displays in the user interface.
name	String	Internal name of the quick start model.
description	String	Description of the quick start model.
framework	String	Framework used to create the model. Quick start models use the framework `PYTHON`.
frameworkType	String	Internal type of the framework. Quick start models use the type `CUSTOM`.

Monitoring quick start models

Use the modelhub resource to get the statuses of all quick start models.

GET request

To monitor all quick start models, use the following URI:

```
/mlops/api/v1/modelhub/monitor
```

GET response

Returns a status summary of each quick start model.

The following table describes the fields in each status summary:

Field	Type	Description
displayName	String	External name of the quick start model that displays in the user interface.
name	String	Internal name of the quick start model.
deploymentStatus	String	Current status of the quick start model.
deploymentState	String	Requested state of the quick start model. When you start or restart a model, the requested state is `ENABLED`. When you stop a model, the requested state is `DISABLED`.
deploymentId	String	ID of the quick start model.
predictUrl	String	Model endpoint URL. Use the URL to generate predictions from the model. Applies when the quick start model is available.

Getting information about a quick start model

Use the modelhub resource to get information about a single quick start model.

GET request

To get information about a quick start model, use the following URI:

```
/mlops/api/v1/modelhub/preBuiltModel?name=<model name>
```

To get information about the image classification model, use `IMAGE_CLASSIFICATION` as the model name. To get information about the text translation model, use `TEXT_TRANSLATION` as the model name.

GET response

Returns a summary of the quick start model.

The following table describes the fields in each quick start model summary:

Field	Type	Description
displayName	String	External name of the quick start model that displays in the user interface.
name	String	Internal name of the quick start model.
description	String	Description of the quick start model.
id	String	ID of the quick start model.
status	String	Current status of the quick start model.
predictUrl	String	Model endpoint URL. Use the URL to generate predictions from the model. Applies when the quick start model is available.

Machine learning models

Use API resources to perform actions on user-defined machine learning models, such as downloading the template file, uploading and downloading a model file, and registering and editing a model.

Getting the code template file

Use the model resource to get the template file that you need to define a machine learning model.

GET request

To request the code template file, use the following URI:

```
/mlops/api/v1/model/template/custom
```

GET response

Returns the code template ZIP file as an octet-stream file.

Model file upload

Upload a model ZIP file that contains the files that define your machine learning algorithm.

Before you upload a new model file, send a request to generate a model file ID. Then, use the model file ID in the request to upload or update the file.

Generating a model file ID

Use the model resource to generate a model file ID before you upload the model file.

POST request

To generate a model file ID, use the following URI:

```
/mlops/api/v1/model/upload/generateID
```

Include the following field in the request:

Field	Type	Description
file_name	String	Name of the model file to upload, including the file extension.

POST response

Returns the new model file ID.

Uploading or updating a model file

Use the model resource and the model file ID to upload or update a model file.

POST request

To upload or update a model file, include the model file ID in the URI. Use the following URI:

```
/mlops/api/v1/model/upload/<model file ID>
```

Include the following fields in the request:

Field	Type	Description
file	String	Model file to upload as a multipart file.
framework	String	Use `PYTHON` or `TENSORFLOW` as the value.

POST response

Returns the model file ID.

Registering a machine learning model

Use the Documents resource to register a machine learning model after you upload the model file.

POST request

To register a machine learning model, include the project or folder ID in the URI. Use the following URI:

```
/frs/v1/Projects('<project ID>')/Documents
```

Include the following fields in the request:

Field	Type	Description
name	String	Machine learning model asset name.
description	String	Optional. Description of the machine learning model.
documentType	String	Use `MLOPS_MODEL`.
nativeData	Object	Blob object that defines the machine learning model.

Include the following fields in the nativeData object:

Field	Type	Description
name	String	Machine learning model asset name.
framework	String	Framework used to create the machine learning model. Use one of the following values: - `Python` - `PyTorch` - `TensorFlow`
framework_type	String	Internal type of the framework. For a Python model, use `Custom`. For a PyTorch or TensorFlow model, use `Standard`.
file_id	String	Model file ID generated when you upload the model file.
file_name	String	Name of the model file. The name must match the file name you used to generate the model file ID.
size	String	Size of the model file in GB.
file_type	String	Use `zip`.
inputField		Input fields that the machine learning model expects. Include at least one input field.
name	String	Name of the field.
dataType	String	Data type of the field.
shape	String	Required for array data types. Shape of the array. For example, `[2,3]`.
outputField		Output fields that the machine learning model returns. Include at least one output field.
name	String	Name of the field.
dataType	String	Data type of the field.
shape	String	Required for array data types. Shape of the array. For example, `[2,3]`.

POST response

Returns a summary of the machine learning model.

Editing a machine learning model

Use the Documents resource to edit a machine learning model.

PATCH request

To edit a machine learning model, include the machine learning model ID in the URI. Use the following URI:

```
/frs/v1/Documents('<model ID>')
```

Get the model ID from the response to register a machine learning model.

Include the following fields in the request:

Field	Type	Description
name	String	Machine learning model asset name.
description	String	Optional. Description of the machine learning model.
documentType	String	Use `MLOPS_MODEL`.
nativeData	Object	Blob object that defines the machine learning model.

Include the following fields in the nativeData object:

Field	Type	Description
name	String	Machine learning model asset name.
framework	String	Framework used to create the machine learning model. Use one of the following values: - `Python` - `PyTorch` - `TensorFlow`
framework_type	String	Internal type of the framework. For a Python model, use `Custom`. For a PyTorch or TensorFlow model, use `Standard`.
file_id	String	Model file ID generated when you upload the model file.
file_name	String	Name of the model file. The name must match the file name you used to generate the model file ID.
size	String	Size of the model file in GB.
file_type	String	Use `zip`.
inputField		Input fields that the machine learning model expects. Include at least one input field.
name	String	Name of the field.
dataType	String	Data type of the field.
shape	String	Required for array data types. Shape of the array. For example, `[2,3]`.

Field	Type	Description
outputField		Output fields that the machine learning model returns. Include at least one output field.
name	String	Name of the field.
dataType	String	Data type of the field.
shape	String	Required for array data types. Shape of the array. For example, `[2,3]`.

PATCH response

Returns a summary of the machine learning model.

Getting the model file for a machine learning model

Use the model resource to download the model ZIP file from a registered machine learning model.

GET request

To get the model file, include the machine learning model ID in the URI. Use the following URI:

```
/mlops/api/v1/model/download/<model ID>
```

Get the model ID from the response to register a machine learning model.

GET response

Returns an octet-stream of the model file.

Model deployments

Use API resources to create a model deployment for a machine learning model, edit the deployment, and monitor deployments in your organization.

Creating a model deployment

Use the Documents resource to create a model deployment.

POST request

To create a model deployment, include the project or folder ID in the URI. Use the following URI:

```
/frs/v1/Projects('<project ID>')/Documents
```

Include the following fields in the request:

Field	Type	Description
name	String	Model deployment name.
description	String	Optional. Description of the model deployment.
documentType	String	Use `MLOPS_DEPLOYMENT`.
documentState	String	Use an empty string.
nativeData	Object	Blob object that defines the model deployment.

Include the following fields in the nativeData object:

Field	Type	Description
name	String	Model deployment name.
modelId	String	ID of the machine learning model to associate with this deployment.
computeUnits	Integer	Maximum number of compute units that you want the model deployment to use. Enter a whole number that is a multiple of 4 from 4 to 40.

POST response

Returns the model deployment summary.

Editing a model deployment

Use the Documents resource to edit a model deployment.

PATCH request

To edit a model deployment, include the deployment ID in the URI. Use the following URI:

```
/frs/v1/Documents('<deployment ID>')
```

Get the deployment ID from the response to create a model deployment or to monitor model deployments.

Include the following fields in the request:

Field	Type	Description
name	String	Model deployment name.
description	String	Optional. Description of the model deployment.
documentType	String	Use `MLOPS_DEPLOYMENT`.
nativeData	Object	Blob object that defines the model deployment.

Include the following fields in the nativeData object:

Field	Type	Description
name	String	Model deployment name.
modelId	String	ID of the machine learning model to associate with this deployment.
computeUnits	Integer	Maximum number of compute units that you want the model deployment to use. Enter a whole number that is a multiple of 4 from 4 to 40.

POST response

Returns the model deployment summary.

Monitoring model deployments

Use the deployment resource to monitor the status of a model deployment.

GET request

You can request the status of all deployments in the organization or the status of a particular deployment.

Status of all deployments in the organization

To get the status of all deployments in the organization, use the following URI:

```
/mlops/api/v1/deployment/monitor
```

You can include parameters in the request to sort and filter the results. Use the following syntax to specify parameters:

```
/mlops/api/v1/deployment/monitor?
offset=<offset>&limit=<limit>&filter=<filter>&sortkey=<sort key>&sortdir=<sort
direction>
```

The following table describes the parameters you can use:

Parameter	Description
offset	Number of results to offset from the beginning. For example, if you want the request to skip the top 5 results, set the offset to 5.
limit	Number of results to return. For example, if you want only the top 10 results, set the limit to 10.
filter	Term to use to filter results. The results only include deployments with the filter term in the deployment name.

Parameter	Description
sortkey	Field to use to sort the results. Use one of the following values: - `NAME` - `DISPLAY_STATUS` - `LOCATION` - `STARTED_BY` - `START_TIME` - `STOP_TIME` - `DURATION`
sortdir	Direction to sort the results. Use either `ASC` or `DESC`.

Status of a particular deployment

To get the status of a particular deployment, include the deployment ID in the URI. Use the following URI:

```
/mlops/api/v1/deployment/monitor/<deployment ID>
```

Get the deployment ID from the response to create a model deployment or a previous response to monitor model deployments.

GET response

Returns details about the results and the deployment status object for every deployment within the query parameters. If you requested the details for a particular deployment, returns only the deployment status object.

The following table describes the fields in a deployment status object:

Field	Description
count	Number of deployments returned in this response.
offset	Offset used in the response. **Note:** This might differ from the offset in the request if the requested value was out of bounds.
limit	Limit used in the response. **Note:** This might differ from the limit in the request if the requested value was out of bounds.
deployments	Object that describes each deployment.
deploymentID	ID of the model deployment.
deploymentName	Name of the model deployment.
state	Requested state of the deployment. When you start or restart the deployment, the state is `ENABLED`. When you stop the deployment, the state is `DISABLED`.
status	Internal status of the deployment.
statusLabel	External status label for the current state and status.
message	Warning or error message string. Applies only if a warning or error occurs.

Field	Description
locationID	ID of the folder that stores the deployment.
locationName	Name of the folder that stores the deployment.
startedByUserID	ID of the user that last started the deployment.
startedbyUserName	Name of the user that last started the deployment.
startTime	Time that the deployment started, in milliseconds from the UNIX epoch, 00:00:00 UTC on January 1, 1970.
stopTime	Time that the deployment stopped, in milliseconds from the UNIX epoch. Applies only when you stop the deployment.
duration	Number of milliseconds since the deployment started running.
updateTime	Time that the deployment status was last updated, in milliseconds from the UNIX epoch.
monitorTime	Time that the deployment status was checked, in milliseconds from the UNIX epoch.
agentSaasid	ID of the agent that deploys the job.
communicationMode	Mode of communication to the REST API. The communication mode is `channel`.
predictUrl	Endpoint URL to use to generate predictions from the model.
framework	Framework used to create the machine learning model.
frameworkType	Type of framework used to register the machine learning model.

Starting and stopping a quick start model or model deployment

Use the control resource to start, restart, and stop a quick start model or a model deployment.

POST request

To start, restart, or stop a deployment, include the deployment ID in the URI.

To start a deployment, use the following URI:

```
/mlops/api/v1/deployment/control/<deployment ID>/start
```

To restart a deployment, use the following URI:

```
/mlops/api/v1/deployment/control/<deployment ID>/restart
```

To stop a deployment, use the following URI:

```
/mlops/api/v1/deployment/control/<deployment ID>/stop
```

Get the ID for a quick start model from the response to get information about a single quick start model. For more information, see "Getting information about a quick start model" on page 508.

Get the deployment ID for a model deployment from the response to monitor a model deployment. For more information, see "Monitoring model deployments" on page 515.

Note: To start a quick start model through an API request, you must first start it at least once through the user interface.

POST response

Returns the deployment object.

The following table describes the fields in a deployment object:

Field	Description
deploymentID	ID of the quick start model or model deployment.
state	Requested state of the deployment. When you start or restart the deployment, the state is ENABLED. When you stop the deployment, the state is DISABLED.
status	Current status of the deployment.
message	Warning or error message string. Applies only if a warning or error occurs.
startedByUserID	ID of the user that last started the deployment.
startedbyUserName	Name of the user that last started the deployment.
startTime	Time that the deployment started, in milliseconds from the UNIX epoch, 00:00:00 UTC on January 1, 1970.
stopTime	Time that the deployment stopped, in milliseconds from the UNIX epoch. Applies only when you stop the deployment.
updateTime	Time that the deployment status was last updated, in milliseconds from the UNIX epoch.

Generating predictions

Use the deployment resource to generate predictions from a quick start model or a user-defined model.

Before you generate predictions, make sure that the deployment is available. To make a deployment available, start the deployment and wait until its status is *Available*.

Note: To generate predictions from a quick start model, use the predictUrl value from the request for information about a single quick start model.To generate predictions from a user-defined machine learning model, use the predictUrl value from the request to monitor a model deployment.

POST request

To generate a prediction, include the deployment ID in the URI. Use the following URI:

```
/mlops/api/v1/deployment/request/<deployment ID>
```

Get the ID for a quick start model from the response to get information about a quick start model. For more information, see "Getting information about a quick start model" on page 508.

Get the deployment ID for a model deployment from the response to monitor a model deployment. For more information, see "Monitoring model deployments" on page 515.

Include the following fields in the request body:

Field	Type	Description
deployment_id	String	ID of the quick start model or model deployment.
request	String	Input fields needed to generate a prediction. The request should consist of key-value pairs in a serialized JSON string. To generate predictions from a quick start model, use the input fields defined for that model. To generate predictions from a user-defined model, use the input fields that you specified when you registered the machine learning model.

POST response

Returns the prediction from the machine learning model.

When you generate predictions from a user-defined model, you specify the output fields when you register the machine learning model. The model returns each output field as an attribute of the response.

Input and response for the image classification model

To generate predictions from the image classification quick start model, include a reference to the image file in the request. If successful, the request returns a list of the image descriptors.

Input fields

Include the following field in the request field of the POST request:

Field	Type	Description
inputImage	String	File path to the image file.

POST response

Returns an array that contains the five image descriptors and confidence scores, similar to the following example:

```
{
  "prediction": [
    {
      "name": "drum, membranophone, tympan",
      "confidence_score": 0.23505400121212006
    },
    {
      "name": "frying pan, frypan, skillet",
      "confidence_score": 0.1420985758304596
    },
```

```
      {
        "name": "stove",
        "confidence_score": 0.11630863696336746
      },
      {
        "name": "ladle",
        "confidence_score": 0.08813537657260895
      },
      {
        "name": "strainer",
        "confidence_score": 0.06395173072814941
      }
    ]
  }
```

Input and response for the text translation model

To generate predictions from the text translation quick start model, include the languages and input text in the request. If successful, the request returns the translated text.

Input fields

Include the following attributes in the request field of the POST request:

Attribute	Type	Description
source_language	String	Language code of the input text.
target_language	String	Language code to translate to.
input_text	String	Text to translate. Text can be a maximum of 3,000 characters.

To see the languages that the model supports and the language codes, refer to the Hugging Face documentation.

POST response

Returns the translated text as a string.

Deleting an asset

Use the Documents resource to delete a user-defined machine learning model or a model deployment.

DELETE request

To delete an asset, include the asset ID in the URI. Use the following URI:

```
/frs/v1/Documents('<ID>')
```

Get the ID of a machine learning model from the response to register or edit a machine learning model. Get the ID of a model deployment from the response to create or monitor a model deployment.

Note: You can't delete a model deployment while it's running, and you can't delete a machine learning model if a running model deployment uses it. To delete a model deployment, stop the deployment before you send a request to delete it. To delete a machine learning model, stop all deployments that use the model before you send a request to delete the model.

DELETE response

Returns a success message.

CHAPTER 8

RunAJob utility

You can use the RunAJob utility to run jobs or check job status instead of making calls directly through the Informatica Intelligent Cloud Services REST API.

The RunAJob utility runs a JAR file that calls an Informatica Intelligent Cloud Services REST API to run a job. After the job completes, the utility provides the following job details:

- The user who initiated the job.
- The time the job was initiated.
- The run ID for the job.

You can use the RunAJob utility for certain Data Integration and Mass Ingestion asset types.

For Data Integration, you can run jobs for the following tasks and taskflows:

- Mapping tasks
- Synchronization tasks
- Replication tasks
- Masking tasks
- PowerCenter tasks
- Linear taskflows
- Published taskflows

For Mass Ingestion, you can run file ingestion task jobs.

You must have the RunAJobCli package enabled in your Informatica Intelligent Cloud Services organization to use the RunAJob utility.

To see if your organization is licensed to use the utility, log in to your organization and click **Administrator > Licenses**, and then look for the RunAJobCli package toward the bottom of the page. If you do not see the package, contact Informatica Global Customer Support to enable it.

When the package is enabled, the utility can be found in the following location:

```
<Secure Agent installation directory>\apps\runAJobCli
```

To use the RunAJob utility, the Secure Agent host must have Java version 1.8 or higher installed.

RunAJob utility setup

To set up the RunAJob utility, create copies of the RunAJob properties template files that are included with the utility and configure the new files.

The RunAJob utility includes the following template files:

- `Restenv_default.properties`. Specifies login credentials and job polling behavior.
- `Log4j2_default.properties`. Specifies the level of detail to return in log files.

To customize the RunAJob properties, copy the template files to create a `restenv.properties` file and optionally a `log4j2.properties` file, and then configure the properties. Or, you can update existing `restenv.properties` and `log4j2.properties` files if you already have them. Use the template files that are included with the utility as a reference.

You can find the template files in the following location:

```
<Secure Agent installation directory>\apps\runAJobCli
```

Login properties

Specify Informatica Intelligent Cloud Services login credentials in the `restenv.properties` file. Or, you can include the login parameters as arguments in a task command.

Use login credentials for a native Informatica Intelligent Cloud Services user account. You can't use a SAML user account for the RunAJob utility.

You can use a password string or an encrypted password for the password parameter value.

To create an encrypted password, use one of the following commands:

```
./cli.sh encryptText -t <password>
./cli.sh encryptText -text <password>
```

Copy the encrypted password string and replace the password in the `restenv.properties` file with the encrypted string, and then set the `use.encryption` flag to true.

The following example shows the `restenv.properties` file with an encrypted password and the `use.encryption` flag set to true:

```
username=saki
password=:1xCGDTC0oD9B2Rmd8Sr4IZWaWWkcEmiK5fy+GkycA==
ACTIVITYMONITORWAIT=2000
TOTALWAIT=60000
PROXYHOST=
PROXYPORT=
RETRYCOUNT=30
use.encryption=true
```

Include the following parameters in the `restenv.properties` file or in task commands:

Parameter	Description
baseUrl	Base URL. Default is `https://dm-us.informaticacloud.com/ma`.
username	Informatica Intelligent Cloud Services user name.

Parameter	Description
password	Informatica Intelligent Cloud Services password or encrypted password string.
use.encryption	Enables use of an encrypted password. To use an encrypted password, set the value to true.

For more information, see "RunAJob utility arguments" on page 527

Job status

Specify the frequency at which the RunAJob utility polls for status in the `restenv.properties` file.

You can use the following parameters:

Parameter	Description
ACTIVITYMONITORWAIT	The amount of time the utility waits before retrying if an internal exception occurs, such as a login failure or network problem. Default is 5000 milliseconds.
TOTALWAIT	The maximum amount of time the utility waits for a job to complete before polling the activity monitor and activity log again for status. Default is 5000 milliseconds.
RETRYCOUNT	The number of times the utility polls for status. This parameter is used for polling the activity monitor and activity log for job status and for internal exceptions such as login failure or network problems. Default is 6. **Note:** Informatica Intelligent Cloud Services adds 10 seconds between each API call to prevent server issues.

When configuring the `restenv.properties` file for polling job status, consider the values you set for TOTALWAIT in conjunction with RETRYCOUNT, keeping in mind the amount of time you expect a job to run.

For example, if you expect a job to run for approximately 25 minutes, you might want to set the parameters as follows:

```
TOTALWAIT=60000
RETRYCOUNT=30
```

With these settings, the utility polls for the job status every 60 seconds up to 30 times with 10 seconds between each retry, which totals 35 minutes. If the job executes more than 35 minutes, the command will exit with return code 6 (which means the job is running) and the job will continue to run in Informatica Intelligent Cloud Services.

When configuring the `restenv.properties` file for internal exceptions, consider the values you set for ACTIVITYMONITORLOG in conjunction with RETRYCOUNT.

For example, you might set these parameters as follows:

```
ACTIVITYMONITORWAIT=5000
RETRYCOUNT=30
```

With these settings, if a user login fails, the utility retries the login every 5 seconds up to 30 times.

Log file detail

You can set the level of job detail to include in log files in the `log4j2.properties` file.

By default, the level of detail is set to INFO. At this level, log files include basic information such as user ID, job ID, and the time that the task was initiated. If you want log files to include job details for debugging purposes, you can change the level of detail to DEBUG. Or, for fine-grained details, you can change the level to TRACE.

To set the level of detail, change the value of the following parameter in the `log4j2.properties` file:

```
rootLogger.level=INFO
```

Using the RunAJob utility

To use the RunAJob utility, type the RunAJob utility command followed by arguments.

The following string is the RunAJob utility command:

```
cli.bat runAJobCli
```

For each job, you must specify the task or taskflow to run. The syntax that you use to run a Data Integration taskflow is slightly different from the syntax you use to run a task.

Running tasks

The following command is an example of the syntax you can use to run a task using the task name and location to specify the task:

```
cli.bat runAJobCli -t <tasktype> -n <task name> -fp <folder path to the task>
```

For example, to run a Mass Ingestion file ingestion task, you might use the following command:

```
cli.bat runAJobCli -t MI_TASK -n mitask_Arch_2308 -fp myproject/folder1
```

The following command is an example of the syntax you can use to run a job using the federated task ID to specify the task:

```
cli.bat runAJobCli -t <tasktype> -fi <federated task ID>
```

For example, to run a Data Integration synchronization task using the federated task ID, you might use the following command:

```
cli.bat runAJobCli -t DSS -fi kvOF40yLXyUihm7wYYskmh
```

Running Data Integration taskflows

To run a taskflow using the RunAJob utility, the taskflow must be published and you must include values for **Allowed Users** and **Allowed Groups** in the taskflow designer. For more information, see *Taskflows* in the Data Integration help.

For each job, you must specify the taskflow to run using the taskflow's name.

To run a taskflow, use the following syntax:

```
cli.bat runAJobCli -t TASKFLOW -un <taskflow name>
```

For example, you might use the following command:

```
cli.bat runAJobCli -t TASKFLOW -un myPublishedTaskflow
```

Running Data Integration taskflows using parameter sets

You can use a parameter set to provide values for taskflow input parameters and run the taskflow using the RunAJob utility.

For more information about running taskflows with parameter sets using the RunAJob utility, see https://knowledge.informatica.com/s/article/DOC-19232. This article also includes a link to download the utility.

When you run a taskflow that uses a parameter set, type the RunAJob utility command followed by the `-pun` or `--parameterSetUniqueName` argument.

To run a taskflow using the RunAJob utility, the taskflow must be published and you must include values for **Allowed Users** and **Allowed Groups** in the taskflow designer. For more information, see *Taskflows* in the Data Integration help.

To run a taskflow that uses a parameter set, use one of the following syntaxes:

```
cli.bat runAJobCli -t TASKFLOW -un <taskflow_name> -fp <folder_path> -pun
<unique_param_set_name>
```

```
cli.bat runAJobCli -t TASKFLOW -un <taskflow_name> -fp <folder_path> --
parameterSetUniqueName <unique_param_set_name>
```

For example:

```
cli.bat runAJobCli -t TASKFLOW -un myTaskflow -fp myproject/folder1 -pun sampleParamSet
```

```
cli.bat runAJobCli -t TASKFLOW -un myTaskflow -fp myproject/folder1 --
parameterSetUniqueName sampleParamSet
```

All taskflows run by using the `/tf` endpoint from the RunAJob utility.

Note: If you use the utility on Linux and you use another script or wrapper script to call cli.sh, be sure to comment the following line in the cli.sh file:

```
cd "$SCRIPT_DIR"
```

If you do not comment this line, you might receive the following error:

```
Could not find or load main class com.informatica.saas.utilities.plugins.RunAJobPlugin.
```

Task location

If you do not include a folder path or federated task ID in the command, the utility runs the task in the Default folder.

If the task is not located in the Default folder or you have multiple tasks with the same name located in different folders, be sure to include the folder path or federated task ID in the command.

To find the federated task ID, send a POST request using the REST API version 3 lookup resource.

RunAJob utility arguments

The RunAJob utility supports short and long options for arguments. Precede a short argument with a single hyphen. Precede a long argument with two hyphens.

You can use the following arguments in a RunAJob command:

Parameter	Short argument	Long argument	Description
username	-u	--user	Informatica Intelligent Cloud Services user name.
password	-p	--password	Informatica Intelligent Cloud Services password.
baseUrl	-bu	--baseUrl	Base URL. Default is `https://dm-us.informaticacloud.com/ma`. Required.
taskId	-i	--taskId	Task ID. Required when the command does not include the task name or federated task ID. Not applicable for taskflows.
folderPath	-fp	--folderPath	Folder path to the location of the task such as `myproject/folder1`. Required when the task isn't in the Default folder and the command doesn't include the federated task ID. Not applicable for taskflows.
frsId	-fi	--frsId	Federated task ID, which is a global unique identifier. Required when the task is not in the Default folder and the command does not include the folder path. Not applicable for taskflows.
taskflowUniqueName	-un	--taskflowUniqueName	Taskflow unique name. Required for Data Integration taskflows. Use instead of taskName.
taskName	-n	--taskName	Task name. Not applicable for taskflows.
taskType	-t	--taskType	Task type. Required. Use one of the following values: - DMASK. Masking task. - DRS. Replication task. - DSS. Synchronization task. - MTT. Mapping task. - PCS. PowerCenter task. - Workflow. Linear taskflow. - MI_TASK. File ingestion task. - TASKFLOW. Taskflow.

Parameter	Short argument	Long argument	Description
parameterSetUniqueName	-pun	--parameterSetUniqueName	Unique name of the parameter set. Can be used for taskflow.
parameterDir	-pd	--parameterDir	Parameter file directory. Include with parameterFile. Default is: <SecureAgentInstallDir>/apps/ Data_Integration_Server/data/ userparameters
parameterFile	-pf	--parameterFile	Parameter file. Can be used for mapping tasks.
waitFlag	-w	--waitFlag	Wait flag. Determines whether to wait for the job to complete or run the job in the background.
insecure	-k	--insecure	Insecure mode.
instanceName	-in	--instanceName	Custom name to be added to a taskflow name.

Job status codes

If a job is successful, the RunAJob utility returns a SUCCESS value of 0. If the job fails, the utility returns errors.

The utility can return the following status codes:

Code	Description
-1	Exception
0	Success
1	Warning
2	No wait
3	Failure
4	Timeout
5	Error
6	Running
7	Failure to start

If any required parameters are missing or are not valid in a command, an error message displays and the REST API call does not run.

CHAPTER 9

ParamSetCli utility

ParamSetCli is Informatica's command line interface utility that is used only for taskflows that use parameter sets.

ParamSetCli enables you to perform the following actions on a parameter set:

1. Upload a parameter set to the cloud-hosted repository.

2. Download a parameter set from the cloud-hosted repository.

3. Delete a parameter set from the cloud-hosted repository.

4. List all the parameter sets available in the cloud-hosted repository.

The following video shows you how to upload, download, delete, and list the parameter set within the cloud-hosted repository:

https://www.youtube.com/watch?v=zDPYS9e0ryM

ParamSetCli utility requirements

Before you run the ParamSetCli utility, you must have a Java Runtime Environment.

To use the ParamSetCli utility with proxy settings, you must have the Secure Agent installed on the same machine as the ParamSetCli utility.

You can download the ParamSetCli utility from the following community article:

https://network.informatica.com/docs/DOC-19232

ParamSetCli utility setup

To set up the ParamSetCli utility, unzip the downloaded `paramsetcli.zip` to a directory other than the Secure Agent installation directory.

The ParamSetCli utility contains the following files:

- `restenv.properties`. Specifies login credentials.

- `log4j.properties`. Specifies the level of detail to return in log files.

To customize the ParamSetCli properties, you can update the existing `restenv.properties` and `log4j.properties` files.

Login properties

Specify the Informatica Intelligent Cloud Services login credentials in the `restenv.properties` file. Or, you can pass the login parameters through cli using the `-u` and `-p` arguments.

You can use a password string or an encrypted password for the password parameter and the PROXYPWD parameter.

To create an encrypted password, use one of the following commands:

`./paramsetcli.sh encryptText -t <password>`

`./paramsetcli.sh encryptText -text <password>`

You can use `paramsetcli.bat` for Windows and `paramsetcli.sh` for Unix-like operating systems.

Copy the encrypted password string and replace the password or proxy password in the `restenv.properties` file with the encrypted string, and then set the `use.encryption` flag to `true`.

The following example shows the `restenv.properties` file with an encrypted password, an encrypted proxy password, and the `use.encryption` flag set to true:

```
baseUrl=https://dm-us.informaticacloud.com/ma
username=testuser
password=:1xCGDTC0oD9B2Rmd8Sr4IZWaWWkcEmiK5fy+GkycA==
ACTIVITYMONITORWAIT=2000
PROXYHOST=
PROXYPORT=
PROXYUSERNAME=
PROXYPWD=x2UNmQ1nxXbKvKs1+ZOkvQ==
RETRYCOUNT=
use.encryption=true
paramSetBaseUrl=https://na1.dm-us.informaticacloud.com/active-bpel
```

Include the following parameters in the `restenv.properties` file or in action commands:

Parameter	Description
baseUrl	Informatica Intelligent Cloud Services MA service URL. Default is `https://dm-us.informaticacloud.com/ma`.
username	Informatica Intelligent Cloud Services user name.
password	Informatica Intelligent Cloud Services password or encrypted password string.
ACTIVITYMONITORWAIT	The amount of time the utility waits before retrying if an internal exception occurs, such as a login failure or network problem. Default is **5000** milliseconds.
PROXYHOST	Host name or an IP address of the HTTP(s) proxy server.
PROXYPORT	Port number of the HTTP(s) proxy server.
PROXYUSERNAME	Authenticated user name for the HTTP(s) proxy server. Required if the proxy server requires authentication.
PROXYPWD	Password for the authenticated user or an encrypted password string for the authenticated user. Required if the proxy server requires authentication.

Parameter	Description
RETRYCOUNT	The number of times the utility polls for status. This parameter is used for polling the activity monitor and activity log for job status and for internal exceptions such as login failure or network problems. Default is **6**. **Note:** Informatica Intelligent Cloud Services adds 10 seconds between each API call to prevent server issues.
use.encryption	Enables the use of an encrypted password. To use an encrypted password or an encrypted proxy password, set the value to **true**.
paramSetBaseUrl	Base URL of Informatica Cloud Data Integration suffixed with active-bpel. Default is `https://na1.dm-us.informaticacloud.com/active-bpel`.

Using the ParamSetCli utility

To use the ParamSetCli utility, type the `runParamSetCli` command followed by the arguments.

Use `paramsetcli.bat runParamSetCli` for Windows. Use `paramsetcli.sh runParamSetCli` for Unix-like operating systems.

For each action that you want to perform, you must specify the action type.

ParamSetCli utility arguments

You can use the following arguments in a ParamSetCli command:

Parameter	Argument	Description
username	-u	Required. The user name to log in to Informatica Intelligent Cloud Services.
password	-p	Required. The password or encrypted password string to log in to Informatica Intelligent Cloud Services.
baseUrl	-bu	Required. Informatica Intelligent Cloud Services MA service URL. Default is https://dm-us.informaticacloud.com/ma.
paramsetbaseUrl	-psbu	Base URL of Informatica Cloud Data Integration suffixed with active-bpel. For example, https://na1.dm-us.informaticacloud.com/active-bpel You can provide the paramset base URL in the `restenv.properties` file or you can pass it through the CLI using the `-psbu` argument. If it is available in both locations, the `-psbu` value in the action command takes precedence.
debug	-d	Optional. Displays debugging information.
insecure	-k	Optional. Enables the insecure mode.

Parameter	Argument	Description
action	-a	Required. Defines the action to be performed on the parameter set. You can specify one of the following actions: upload- Upload a parameter set to the cloud-hosted repository by using the unique name. Optionally, you can specify the -f force argument to override the existing parameter set. download- Download a parameter set from the cloud-hosted repository by using the unique name of an uploaded parameter set. If the specified file already exists on your local machine, the CLI does not download the parameter set. To forcefully download and overwrite the existing file, you can use the -f force argument. list- List the parameter sets with basic information such as created by, creation date, parameter set name, parameter file name, and parameter directory available in the cloud-hosted repository. delete- Permanently delete the specified parameter set from the cloud-hosted repository.
Force	-f	Optional. Enables you to force upload or download the parameter set.
Parameter Set Unique Name	-un	Required. A name used to identify a parameter set when performing different operations such as upload, download, or delete a parameter set, or use it in a taskflow. The name must be unique within the IICS organization.
Parameter directory	-pd	Specifies the directory where the parameter files are located.
Parameter file name	-pf	Parameter file name.
Parameter Set List Page	-page	Optional. Page number. Specifies the page number that you want to list.
Page size	-ps	Optional. Specifies the number of entries per page. The default and the maximum number of entries that you can list is 50 per page.

Running the ParamSetCli commands

You can use the following command to run an action using the action name.

```
paramsetcli.bat runParamSetCli -a <actionName>
```

You can perform different actions such as upload, download, and delete a parameter set and list all of the uploaded parameter sets. For each action, you must provide the -a argument and appropriate arguments specific to the action as shown below:

Upload a parameter set

You can use the upload action to upload a parameter set to the cloud-hosted repository. You can upload a file with a size up to 5 MB.

To upload a parameter set, use the following syntax:

```
paramsetcli.bat runParamSetCli -un <unique_param_set_name> -pf <parameter_file_name> -pd <parameter_directory> -a <action>
```

For example:

```
paramsetcli.bat runParamSetCli -un Paramset.params -pf Paramsetfile.params -pd C:\files -a upload
```

Here, `-un`, `-pf`, and `-pd` are mandatory arguments. If a parameter set already exists in the cloud-hosted repository, you cannot upload a parameter set with the same name. However, if you want to force an update of the content in the existing parameter set, you can use the `-f` force argument.

If you update the parameter set after uploading it to the cloud-hosted repository, you must reupload the parameter set for the changes to take effect.

After you run this command, the parameter set is uploaded to the cloud-hosted repository. The unique name of the parameter set is used in a taskflow to read the inputs.

Download a parameter set

You can use the unique name of an uploaded parameter set to download it from the cloud-hosted repository.

To download a parameter set, use the following syntax:

```
paramsetcli.bat runParamSetCli -un <unique_param_set_name> -pf <parameter_file_name> -pd
<parameter_directory> -a <action>
```

For example:

```
paramsetcli.bat runParamSetCli -un Paramset.params -pf Paramsetfile.paramsDownload -pd
C:\files -a download
```

Here, `-un`, `-pf`, and `-pd` are mandatory arguments. `-pf` specifies the file name with which you want to download the parameter set, and `-pd` specifies the directory where the parameter set is to be downloaded. If the specified file already exists on the local machine, the ParamSetCli utility does not download the parameter set. To forcefully download and overwrite the existing file, you can use the `-f` force argument.

After you run this command, the parameter set is downloaded to the specified directory.

Delete a parameter set

You can use the unique name of the parameter set to delete it from the cloud-hosted repository.

To delete a parameter set, use the following syntax:

```
paramsetcli.bat runParamSetCli -un <unique_param_set_name> -a <action>
```

For example:

```
paramsetcli.bat runParamSetCli -un Paramset.params -a delete
```

Here, `-un` is a mandatory argument.

After you run this command, the parameter set is permanently deleted from the cloud-hosted repository.

List the parameter sets

You can list the parameter sets that you have uploaded to the cloud-hosted repository.

To list the parameter sets, use the following syntax:

```
paramsetcli.bat runParamSetCli -a list
```

When you run the above command, the first page with 50 entries is listed. It displays basic information such as created by, creation date, parameter set name, parameter file name, and parameter directory for the parameter sets available in the cloud-hosted repository.

The default and the maximum number of entries that you can list is 50 per page. If you want a customized list, you can specify the page number and page size as per the requirement.

To customize the parameter set list, use the following syntax:

```
paramsetcli.bat runParamSetCli -a list -page <page_number> -ps <page_size>
```

For example: `paramsetcli.bat runParamSetCli -a list -page <5> -ps <10>`

Here, `-page` is the page number that you want to list and `-ps` is the number of entries you want to display on that page.

After you run this command, the fifth page with 10 entries are listed.

CHAPTER 10

REST API codes

The Informatica Intelligent Cloud Services REST API uses codes to represent data such as country names. Use the codes to pass information to the REST API and to interpret the data returned by the REST API.

The Informatica Intelligent Cloud Services REST API uses codes for the following information:

- State codes. Represent the names of the United States.
- Country codes. Represent country names.
- Time zone codes. Represent time zones.

State codes

The Informatica Intelligent Cloud Services REST API uses the following codes to represent the names of the United States.

- AL. Alabama.
- AK. Alaska.
- AZ. Arizona.
- AR. Arkansas.
- CA. California.
- CO. Colorado.
- CT. Connecticut.
- DE. Delaware.
- FL. Florida.
- GA. Georgia.
- HI. Hawaii.
- ID. Idaho.
- IL. Illinois.
- IN. Indiana.
- IA. Iowa.
- KS. Kansas.
- KY. Kentucky.
- LA. Louisiana.

- ME. Maine.
- MD. Maryland.
- MA. Massachusetts.
- MI. Michigan.
- MN. Minnesota.
- MS. Mississippi.
- MO. Missouri.
- MT. Montana.
- NE. Nebraska.
- NV. Nevada.
- NH. New Hampshire.
- NJ. New Jersey.
- NM. New Mexico.
- NY. New York.
- NC. North Carolina.
- ND. North Dakota.
- OH. Ohio.
- OK. Oklahoma.
- OR. Oregon.
- PA. Pennsylvania.
- RI. Rhode Island.
- SC. South Carolina.
- SD. South Dakota.
- TN. Tennessee.
- TX. Texas.
- UT. Utah.
- VT. Vermont.
- VA. Virginia.
- WA. Washington.
- WV. West Virginia.
- WI. Wisconsin.
- WY. Wyoming.

Country codes

The Informatica Cloud REST API uses the following codes to represent country names.

- AF. Afghanistan.
- AX. Aland Islands.

- AL. Albania.
- DZ. Algeria.
- AS. American Samoa.
- AD. Andorra.
- AO. Angola.
- AI. Anguilla.
- AQ. Antarctica.
- AG. Antigua and Barbuda.
- AR. Argentina.
- AM. Armenia.
- AW. Aruba.
- AU. Australia.
- AT. Austria.
- AZ. Azerbaijan.
- BS. Bahamas.
- BH. Bahrain.
- BD. Bangladesh.
- BB. Barbados.
- BY. Belarus.
- BZ. Belize.
- BE. Belgium.
- BJ. Benin.
- BM. Bermuda.
- BT. Bhutan.
- BO. Bolivia.
- BA. Bosnia and Herzegovina.
- BW. Botswana.
- BV. Bouvet Island.
- BR. Brazil.
- IO. British Indian Ocean Territory.
- BN. Brunei Darussalam.
- BG. Bulgaria.
- BF. Burkina Faso.
- BI. Burundi.
- KH. Cambodia.
- CM. Cameroon.
- CA. Canada.
- CV. Cape Verde.
- KY. Cayman Islands.

- CF. Central African Republic.
- TD. Chad.
- CL. Chile.
- CN. China.
- CX. Christmas Island.
- CC. Cocos (Keeling) Islands.
- CO. Colombia.
- KM. Comoros.
- CG. Congo.
- CD. Congo, the Democratic Republic of the.
- CK. Cook Islands.
- CR. Costa Rica.
- CI. Cote d'Ivoire.
- HR. Croatia.
- CU. Cuba.
- CY. Cyprus.
- CZ. Czech Republic.
- DK. Denmark.
- DM. Dominica.
- DO. Dominican Republic.
- DJ. Djibouti.
- EC. Ecuador.
- EG. Egypt.
- SV. El Salvador.
- GQ. Equatorial Guinea.
- ER. Eritrea.
- EE. Estonia.
- ET. Ethiopia.
- FK. Falkland Islands (Malvinas).
- FO. Faroe Islands.
- FJ. Fiji.
- FI. Finland.
- FR. France.
- GF. French Guiana.
- PF. French Polynesia.
- TF. French Southern Territories.
- GA. Gabon.
- GM. Gambia.
- GE. Georgia.

- DE. Germany.
- GH. Ghana.
- GI. Gibraltar.
- GR. Greece.
- GL. Greenland.
- GD. Grenada.
- GP. Guadeloupe.
- GU. Guam.
- GT. Guatemala.
- GG. Guernsey.
- GN. Guinea.
- GW. Guinea-Bissau.
- GY. Guyana.
- HT. Haiti.
- HM. Heard Island and McDonald Islands.
- HN. Honduras.
- HK. Hong Kong.
- HU. Hungary.
- IS. Iceland.
- IN. India.
- ID. Indonesia.
- IR. Iran, Islamic Republic of.
- IQ. Iraq.
- IE. Ireland.
- IL. Israel.
- IM. Isle of Man.
- IT. Italy.
- JM. Jamaica.
- JP. Japan.
- JE. Jersey.
- JO. Jordan.
- KZ. Kazakhstan.
- KE. Kenya.
- KI. Kiribati.
- KP. Korea, Democratic People's Republic of.
- KR. Korea, Republic of.
- KW. Kuwait.
- KG. Kyrgyzstan.
- LA. Lao People's Democratic Republic.

- LV. Latvia.
- LB. Lebanon.
- LS. Lesotho.
- LR. Liberia.
- LY. Libyan Arab Jamahiriya.
- LI. Liechtenstein.
- LT. Lithuania.
- LU. Luxembourg.
- MO. Macao.
- MK. Macedonia, the former Yugoslav Republic of.
- MG. Madagascar.
- MW. Malawi.
- MY. Malaysia.
- MV. Maldives.
- ML. Mali.
- MT. Malta.
- MH. Marshall Islands.
- MR. Mauritania.
- MU. Mauritius.
- MQ. Martinique.
- YT. Mayotte.
- MX. Mexico.
- FM. Micronesia, Federated States of.
- MD. Moldova, Republic of.
- MC. Monaco.
- MN. Mongolia.
- ME. Montenegro.
- MS. Montserrat.
- MA. Morocco.
- MZ. Mozambique.
- MM. Myanmar.
- NA. Namibia.
- NR. Nauru.
- NP. Nepal.
- NL. Netherlands.
- NC. New Caledonia.
- NZ. New Zealand.
- NI. Nicaragua.
- NE. Niger.

- NG. Nigeria.
- NU. Niue.
- NF. Norfolk Island.
- MP. Northern Mariana Islands.
- OM. Oman.
- PK. Pakistan.
- PW. Palau.
- PS. Palestinian Territory, Occupied.
- PA. Panama.
- PG. Papua New Guinea.
- PY. Paraguay.
- PE. Peru.
- PH. Philippines.
- PN. Pitcairn.
- PL. Poland.
- PT. Portugal.
- PR. Puerto Rico.
- QA. Qatar.
- RE. Reunion.
- RO. Romania.
- RU. Russian Federation.
- RW. Rwanda.
- BL. Saint Barthelemy.
- SH. Saint Helena.
- KN. Saint Kitts and Nevis.
- LC. Saint Lucia.
- MF. Saint Martin (French part).
- PM. Saint Pierre and Miquelon.
- VC. Saint Vincent and the Grenadines.
- WS. Samoa.
- SM. San Marino.
- ST. Sao Tome and Principe.
- SA. Saudi Arabia.
- SN. Senegal.
- RS. Serbia.
- SC. Seychelles.
- SL. Sierra Leone.
- SG. Singapore.
- SK. Slovakia.

- SI. Slovenia.
- SB. Solomon Islands.
- SO. Somalia.
- ZA. South Africa.
- GS. South Georgia and the South Sandwich Islands.
- ES. Spain.
- LK. Sri Lanka.
- SD. Sudan.
- SR. Suriname.
- SJ. Svalbard and Jan Mayen.
- SZ. Swaziland.
- SY. Syrian Arab Republic.
- SE. Sweden.
- CH. Switzerland.
- TW. Taiwan.
- TJ. Tajikistan.
- TZ. Tanzania, United Republic of.
- TH. Thailand.
- TL. Timor-Leste.
- TG. Togo.
- TK. Tokelau.
- TO. Tonga.
- TT. Trinidad and Tobago.
- TN. Tunisia.
- TR. Turkey.
- TC. Turks and Caicos Islands.
- TM. Turkmenistan.
- TV. Tuvalu.
- UG. Uganda.
- UA. Ukraine.
- AE. United Arab Emirates.
- GB. United Kingdom.
- US. United States.
- UM. United States Minor Outlying Islands.
- UY. Uruguay.
- UZ. Uzbekistan.
- VU. Vanuatu.
- VA. Holy See (Vatican City State).
- VE. Venezuela.

- VN. Viet Nam.
- VG. Virgin Islands, British.
- VI. Virgin Islands, U.S.
- WF. Wallis and Futuna.
- EH. Western Sahara.
- YE. Yemen.
- ZM. Zambia.
- ZW. Zimbabwe.

Time zone codes

The Informatica Intelligent Cloud Services REST API uses the following time zone codes:

- ACT
- AET
- Africa/Cairo
- Africa/Casablanca
- Africa/Johannesburg
- Africa/Nairobi
- America/Barbados
- America/Bogota
- America/Buenos_Aires
- America/Caracas
- America/Chicago
- America/Costa_Rica
- America/Dawson_Creek
- America/Denver
- America/Dominica
- America/El_Salvador
- America/Guadeloupe
- America/Halifax
- America/Havana
- America/Jamaica
- America/La_Paz
- America/Los_Angeles
- America/Mexico_City
- America/Montreal
- America/New_York
- America/Panama

- America/Phoenix
- America/Puerto_Rico
- America/Tijuana
- America/Vancouver
- Asia/Baghdad
- Asia/Bahrain
- Asia/Dubai
- Asia/Hong_Kong
- Asia/Jerusalem
- Asia/Karachi
- Asia/Katmandu
- Asia/Kuala_Lumpur
- Asia/Kuwait
- Asia/Magadan
- Asia/Muscat
- Asia/Qatar
- Asia/Rangoon
- Asia/Riyadh
- Asia/Seoul
- Asia/Singapore
- AST
- Atlantic/Cape_Verde
- Atlantic/South_Georgia
- Australia/Lord_Howe
- Australia/Perth
- Brazil/Acre
- Brazil/DeNoronha
- Brazil/East
- Brazil/West
- BST
- CNT
- CTT
- Europe/Amsterdam
- Europe/Athens
- Europe/Belgrade
- Europe/Berlin
- Europe/Brussels
- Europe/Bucharest
- Europe/Budapest

- Europe/Copenhagen
- Europe/Istanbul
- Europe/London
- Europe/Luxembourg
- Europe/Madrid
- Europe/Moscow
- Europe/Paris
- Europe/Prague
- Europe/Rome
- Europe/Stockholm
- Europe/Vienna
- Europe/Warsaw
- Europe/Zurich
- GMT
- HST
- Indian/Mauritius
- IST
- JST
- Pacific/Apia
- Pacific/Auckland
- Pacific/Chatham
- Pacific/Enderbury
- Pacific/Fiji
- Pacific/Gambier
- Pacific/Kiritimati
- Pacific/Norfolk
- Pacific/Tahiti
- UTC
- VST

CHAPTER 11

REST API resource quick references

Use the resource quick reference lists to find quick descriptions of REST API resources used by the Informatica Intelligent Cloud Services platform and services.

Platform resource quick reference

The following list contains the syntax and a brief description of the Informatica Intelligent Cloud Services REST API platform resources:

activityLog GET

Version 2 resource.

Returns information from the Monitor service.

Use the serverUrl from the login response as the base URL for one of the following URIs:

```
/api/v2/activity/activityLog/<id>
/api/v2/activity/activityLog?runId=<runId>
/api/v2/activity/activityLog?taskId=<taskId>
/api/v2/activity/activityLog?offset=<offset>
/api/v2/activity/activityLog?rowLimit=<row limit>
```

You can also use the activityLog to download error logs and session logs from the server.

Use the serverUrl from the login response for one of the following URIs:

```
/api/v2/activity/errorLog/<id>
/api/v2/activity/activityLog/<Top_Level_Log_Entry_Id>/sessionLog?itemId=<child-log-
entry-item-id>&childItemId=<child-log-entry-item-id>
```

activityMonitor GET

Version 2 resource.

Returns information from the Monitor service.

Use the serverUrl from the login response as the base URL for the following URI:

```
/api/v2/activity/activityMonitor?details=<true|false>
```

agent GET

Version 2 resource.

Returns the details of a Secure Agent or of all Secure Agents in the organization. Also returns a Secure Agent install token and checksum download URL.

Use the serverUrl from the login response as the base URL for one of the following URIs:

- To get Secure Agent details, use one of the following URIs:

```
/api/v2/agent
/api/v2/agent/<id>
/api/v2/agent/name/<name>
```

- To get a Secure Agent install token and checksum download URL, use the following URI:

```
/api/v2/agent/installerInfo/<install platform>
```

agent DELETE

Version 2 resource.

Deletes a Secure Agent.

Use the serverUrl from the login response as the base URL for the following URI:

```
/api/v2/agent/<id>
```

agentservice POST

Version 3 resource

Starts or stops a Secure Agent service.

Use the baseApiUrl from the login response as the base URL for the following URI:

```
public/core/v3/agent/service
```

auditlog GET

Version 2 resource.

Returns audit log entries.

Use the serverUrl from the login response as the base URL for one of the following URIs:

```
/api/v2/auditlog
/api/v2/auditlog?batchId=<batchId>&batchSize=<batchSize>
```

bundleObject GET

Version 2 resource.

Returns the details of a bundle or the details of all published or installed bundles in the organization.

Use the serverUrl from the login response as the base URL for one of the following URIs:

```
/api/v2/bundleObject
/api/v2/bundleObject/<id>
/api/v2/bundleObject/name/<name>
/api/v2/bundleObject/?published=true
/api/v2/bundleObject/?published=true&installed=false
/api/v2/bundleObject/?installed=true
/api/v2/bundleObject/?published=false&installed=true
```

bundleObject POST

Version 2 resource.

Pushes a published private bundle to sub-organizations.

Use the serverUrl from the login response as the base URL in the following URI:

```
/api/v2/bundleObject/push/<bundleId>
```

bundleObjectLicense GET

Version 2 resource.

Returns the details of all bundles available to or installed on the organization.

Use the serverUrl from the login response as the base URL in the following URI:

```
/api/v2/bundleObjectLicense/<bundleObjectId>
```

bundleObjectLicense POST

Version 2 resource.

Installs a bundle.

Use the serverUrl from the login response as the base URL in the following URI:

```
/api/v2/bundleObjectLicense/
```

Use a bundleObjectLicense object to define attributes. Include the following required attribute: bundleId.

bundleObjectLicense DELETE

Version 2 resource.

Uninstalls a bundle.

Use the serverUrl from the login response as the base URL in the following URI:

```
/api/v2/bundleObjectLicense?bundleObjectId=<bundleId>&updateOption=<updateOption>
```

ChangePassword POST

Version 3 resource.

Changes the password for the user who initiated the session or for a specified user.

Use the baseApiUrl from the login response as the base URL for the following URI:

```
/public/core/v3/Users/ChangePassword
```

checkin POST

Version 3 resource.

Updates the repository with latest version of an object.

Use the baseApiUrl from the login response as the base URL for the following URI:

```
/public/core/v3/checkin
```

checkout POST

Version 3 resource.

Checks out an object from the repository.

Use the baseApiUrl from the login response as the base URL for the following URI:

```
/public/core/v3/checkout
```

commit GET

Version 3 resource.

Returns details for a commit.

Use the baseApiUrl from the login response as the base URL for the following URI:

```
/public/core/v3/commit/<commit hash>
```

commitHistory GET

Version 3 resource.

Returns commit history for source-controlled objects with the latest commit listed first.

Use the baseApiUrl from the login response as the base URL for the following URI:

```
/public/core/v3/commitHistory<parameters>
```

export POST

Version 3 resource.

Starts an export job and returns an export job ID.

Use the baseApiUrl from the login response as the base URL for the following URI:

```
/public/core/v3/export
```

export GET

Version 3 resource.

Returns export status or the export package.

Use the baseApiUrl from the login response as the base URL for one of the following URIs:

- To receive status of an export job, use one of the following URIs:

```
/public/core/v3/export/<id>
/public/core/v3/export/<id>?expand=objects
```

- To download the export job log, use the following URI:

```
/public/core/v3/export/<id>/log
```

- To receive a ZIP stream of the export package, use the following URI:

```
/public/core/v3/export/<id>/package
```

fetchState POST

Version 3 resource.

Creates an object states package and returns a fetchState job ID.

Use the baseApiUrl from the login response as the base URL for the following URI:

```
/public/core/v3/fetchState
```

fetchState GET

Version 3 resource.

Returns the status of the fetchState job or the object states package.

Use the baseApiUrl from the login response as the base URL for one of the following URIs:

- To receive status of a fetchState job, use one of the following URIs:

```
/public/core/v3/fetchState/<id>
/public/core/v3/fetchState/<id>?expand=objects
```

- To receive a ZIP stream of the object states package, use the following URI:

```
/public/core/v3/fetchState/<id>/package
```

folders POST

Version 3 resource.

Creates a folder in the Default project.

Use the baseApiUrl from the login response as the base URL in the following URI:

```
/public/core/v3/folders
```

folders PATCH

Version 3 resource.

Updates a folder that's in the Default project.

Use the baseApiUrl from the login response as the base URL in the following URI:

```
/public/core/v3/folders/<folder ID>
```

Note: To create or update a folder that is not in the Default project and to delete a folder, use the projects resource.

import POST

Version 3 resource.

Uploads an import package or starts an import job.

Use the baseApiUrl from the login response as the base URL for one of the following URIs:

- To upload an import package, use the following URI:

  ```
  /public/core/v3/import/package
  ```

  For Content-Type, use multipart/form-data.

- To specify details for an import job and start the job, use the following URI:

  ```
  /public/core/v3/import/<id>
  ```

import GET

Version 3 resource.

Returns status of an import job.

Use the baseApiUrl from the login response as the base URL for one of the following URIs:

- To receive status of an import job, use one of the following URIs:

  ```
  /public/core/v3/import/<id>
  /public/core/v3/import/<id>?expand=objects
  ```

- To download the import job log, use the following URI:

  ```
  /public/core/v3/import/<id>/log
  ```

job POST

Version 2 resource.

Starts or stops a task and optionally provides job status.

Use the serverUrl from the login response as the base URL for one of the following URIs:

- To start a task, use the following URI:

  ```
  /api/v2/job
  ```

- To stop a task, use the following URI:

  ```
  /api/v2/job/stop
  ```

Do not use this resource for a file ingestion task. Instead, use the file ingestion job resource. For more information, see .

key GET

Version 3 resource.

Returns key rotation interval details for an organization.

Use the baseApiUrl from the login response as the base URL in the following URI:

```
/public/core/v3/key/rotationSettings
```

key PATCH

Version 3 resource.

Changes the key rotation interval for an organization.

Use the baseApiUrl from the login response as the base URL in the following URI:

```
/public/core/v3/key/rotationSettings
```

license GET

Version 3 resource.

Returns the license details for the organization that you are logged in to or a specified sub-organization.

Use the baseApiUrl from the login response as the base URL in the following URI:

```
/public/core/v3/license/org/<id>
```

license PUT

Version 3 resource.

Updates license information for a sub-organization.

Use the baseApiUrl from the login response as the base URL in the following URI:

```
/public/core/v3/license/org/<id>
```

Use the orgLicenseAssignment object to update license information.

loadState POST

Version 3 resource.

Uploads an object states package ZIP file or loads the object states.

Use the baseApiUrl from the login response as the base URL for one of the following URIs:

- To upload an object states package, use the following URI:

  ```
  /public/core/v3/loadState/package
  ```

  For Content-Type, use multipart/form-data.

- To specify details for a loadState job and start the job, use the following URI:

  ```
  /public/core/v3/loadState/<id>
  ```

loadState GET

Version 3 resource.

Returns status of a loadState job.

Use the baseApiUrl from the login response as the base URL and include the job ID in one of the following URIs:

```
/public/core/v3/loadState/<id>
/public/core/v3/import/<id>?expand=objects
```

login POST

Version 3 resource.

Logs into an organization and returns a session ID that you use for API calls that you make during the session.

Use the following URL:

```
https://<cloud provider>-<region>.informaticacloud.com/saas/public/core/v3/login
```

For more information about the cloud provider and region, see .

Don't include INFA-SESSION-ID in the request header.

Use a login object and include the following fields: username, password.

logout POST

Version 3 resource.

Logs out of an organization and ends the REST API session included in the request header.

Use the same URL used for the login POST except for the API name. Use the following URI:

```
https://<cloud provider>-<region>.informaticacloud.com/saas/public/core/v3/logout
```

login POST

Version 2 resource.

Logs into an organization and returns a session ID that you use for API calls that you make during the session.

To log in with your Informatica Intelligent Cloud Services account, use the following URL:

```
https://<cloud provider>-<region>.informaticacloud.com/ma/api/v2/user/login
```

For more information about the cloud provider and region, see "login" on page 54.

Don't include icSessionId in the request header.

Use a login object and include the following fields: username, password.

loginSAML POST

Version 2 resource.

For SAML single sign-on users, logs into an organization and returns a session ID that you can use for other API calls.

To log in to an organization, use the following URL:

```
https://<cloud provider>-<region>.informaticacloud.com/ma/api/v2/user/loginSaml
```

For more information about the cloud provider and region, see "login" on page 54.

Don't include icSessionId in the request header.

Include the following required attributes in the login object: orgId, samlToken.

loginSf POST

Version 2 resource.

Logs into an organization using Salesforce credentials and returns a session ID that you can use for other API calls.

To log in to an organization, use the following URL:

```
https://<cloud provider>-<region>.informaticacloud.com/ma/api/v2/user/loginSf
```

For more information about the cloud provider and region, see "login" on page 54.

Don't include icSessionId in the request header.

Include the following required attributes in the login object: sfSessionId, sfServerUrl.

logout POST

Version 2 resource.

Logs out of an organization and ends the REST API session included in the request header.

Use the serverUrl from the login response as the base URL for the following URI:

```
/api/v2/user/logout
```

logoutall POST

Version 2 resource.

Logs out of an organization and ends all version 2 REST API sessions for the organization.

Use the following URL:

```
https://<cloud provider>-<region>.informaticacloud.com/ma/api/v2/user/logoutall
```

Use a logout object to define attributes. Include the following required attributes: username, password.

Don't include icSessionId in the request header.

lookup POST

Version 3 resource.

Looks up an object's ID, name, path, or type attributes.

Use the baseApiUrl from the login response as the base URL for the following URI:

```
/public/core/v3/lookup
```

objects GET

Version 3 resource.

Returns a list of an organization's assets based on query parameters and returns a list of object dependencies for a specified asset. Also returns permission details for an organization's assets, projects, and folders, and the access rights you have for an object.

Use the baseApiUrl from the login response as the base URL for one of the following URIs:

- To get a list of an organization's assets, use the following URI:

  ```
  /public/core/v3/objects?<parameters>
  ```

- To get a list of object dependencies for an asset, use the following URI:

  ```
  /public/core/v3/objects/<id>/references?<parameters>
  ```

- To get the details for a particular ACL, use the following URI:

  ```
  /public/core/v3/objects/<id>/permissions/<ACL ID>
  ```

- To get the details for all permissions on an object, use the following URI:

  ```
  /public/core/v3/objects/<id>/permissions
  ```

- To get the access rights that the logged-in user has for an object, use the following URI:

  ```
  /public/core/v3/objects/<id>/permissions/checkAccess
  ```

- To find out if you can create a particular type of asset in a project or folder, use the following URI:

  ```
  /public/core/v3/objects/<id>/permissions/checkAccess?type=<type>
  ```

objects POST

Version 3 resource.

Creates or updates an access control list (ACL). ACLs define user or user group permissions to assets, folders, and projects.

Use the baseApiUrl from the login response as the base URL for one of the following URIs:

- To create an ACL for an object, use the following URI:

  ```
  /public/core/v3/objects/<id>/permissions
  ```

- To update an ACL for an object, use the following URI:

  ```
  /public/core/v3/objects/<id>/permissions/<ACL ID>
  ```

objects DELETE

Version 3 resource.

Deletes a custom role.

Use the baseApiUrl from the login response as the base URL one of the following URIs:

- To delete the permissions that a user or user group has for an object, use the following URI:

  ```
  /public/core/v3/objects/<id>/permissions/<ACL ID>
  ```

- To delete all of the permissions configured for an object, use the following URI:

```
/public/core/v3/objects/<id>/permissions
```

org GET

Version 2 resource.

Returns the details of your Informatica Intelligent Cloud Services organization or a related sub-organization.

Use the serverUrl from the login response as the base URL for one of the following URIs:

```
/api/v2/org/<id>
/api/v2/org/name/<name>
```

org POST

Version 2 resource.

Updates the details of an Informatica Intelligent Cloud Services organization or a related sub-organization.

Use the serverUrl from the login response as the base URL for the following URI:

```
/api/v2/org/<id>
```

Use an org object to define attributes.

org DELETE

Version 2 resource.

Deletes a related sub-organization.

Use the serverUrl from the login response as the base URL for the following URI:

```
/api/v2/org/<id>
```

Orgs GET

Version 3 resource.

Returns a list of trusted IP address ranges for an Informatica Intelligent Cloud Services organization or sub-organization.

Use the baseApiUrl from the login response as the base URL for one of the following URI:

```
/public/core/v3/Orgs/<orgId>/TrustedIP
```

Orgs PUT

Version 3 resource.

Enables or disables trusted IP ranges and adds values of trusted IP ranges for an Informatica Intelligent Cloud Services organization or sub-organization. Also creates or removes mappings between SAML groups and roles and Informatica Intelligent Cloud Services roles.

Use the baseApiUrl from the login response as the base URL for one of the following URIs:

- To enable, disable, or add trusted IP ranges, use the following URI:

```
/public/core/v3/Orgs/<orgId>/TrustedIP
```

- To add a mapping between SAML groups and a Informatica Intelligent Cloud Services role, use the following URI:

```
/public/core/v3/Orgs/<organization ID>/addSamlGroupMappings
```

- To remove a mapping between SAML groups and a Informatica Intelligent Cloud Services role, use the following URI:

```
/public/core/v3/Orgs/<organization ID>/removeSamlGroupMappings
```

- To add a mapping between SAML roles and aInformatica Intelligent Cloud Services role, use the following URI:

  ```
  /public/core/v3/Orgs/<organization ID>/addSamlRoleMappings
  ```

- To remove a mapping between SAML roles and a Informatica Intelligent Cloud Services role, use the following URI:

  ```
  /public/core/v3/Orgs/<organization ID>/removeSamlRoleMappings
  ```

privileges GET

Version 3 resource.

Returns a list of privileges that can be used in custom roles.

Use the baseApiUrl from the login response as the base URL for one of the following URIs:

- To get a list of enabled and default privileges, use the following URI:

  ```
  /public/core/v3/privileges
  ```

- To get a complete list of enabled, disabled, default, and unassigned privileges, use the following URI:

  ```
  /public/core/v3/privileges?<parameters>
  ```

projects POST

Version 3 resource.

Creates a project. Also creates a folder in a project other than the Default project.

Use the baseApiUrl from the login response as the base URL in one of the following URIs:

- To create a project, use the following URI:

  ```
  /public/core/v3/projects
  ```

- To create a folder in a project other than the Default project, use one of the followng URIs:

  ```
  /public/core/v3/projects/<id>/folders
  /public/core/v3/projects/name/<name>/folders
  ```

projects PATCH

Version 3 resource.

Updates a project. Also updates a folder that's in a project other than the Default project.

Use the baseApiUrl from the login response as the base URL in one of the following URIs:

- To update a project, use one of the following URIs:

  ```
  /public/core/v3/projects/<id>
  /public/core/v3/projects/name/<name>
  ```

- To update a folder that's in a project other than the Default project, use one of the following URIs:

  ```
  /public/core/v3/projects/<id>/folders/<id>
  /public/core/v3/projects/name/<name>/folders/name/<name>
  ```

Note: To create or update a folder that's in the Default project, use the folders resource.

projects DELETE

Version 3 resource.

Deletes a project or folder.

Use the baseApiUrl from the login response as the base URL in one of the following URIs:

- To delete a project, use one of the following URIs:

  ```
  /public/core/v3/projects/<id>
  /public/core/v3/projects/name/<name>
  ```

- To delete a folder that's in the Default project, use the following URI:

    ```
    /public/core/v3/projects/<folder ID>
    ```

- To delete a folder that's in a project other than the Default project, use one of the following URIs:

    ```
    /public/core/v3/projects/<id>/folders/<id>
    /public/core/v3/projects/name/<name>/folders/name/<name>
    ```

pull GET

Version 3 resource.

Note: The pull GET request that was used to get the status of a pull operation is deprecated. Use a sourceControlAction GET request to receive the status for a source control operation.

pull POST

Version 3 resource.

Retrieves objects from your repository and loads them into your organization.

Use the baseApiUrl from the login response as the base URL for the following URI:

    ```
    /public/core/v3/pull
    ```

pullByCommitHash POST

Version 3 resource.

Retrieves objects from your repository that were modified by a particular commit and loads them into your organization.

Use the baseApiUrl from the login response as the base URL for the following URI:

    ```
    /public/core/v3/pullByCommitHash
    ```

register POST

Version 2.

Creates an Informatica Intelligent Cloud Services sub-organization based on an Informatica Intelligent Cloud Services user account. For Informatica Intelligent Cloud Services partners only.

Use the serverUrl from the login response as the base URL for the following URI:

    ```
    /api/v2/user/register
    ```

Use a registration object to define attributes.

ResetPassword POST

Version 3 resource.

Resets the password for the user who initiated the session.

Use the baseApiUrl from the login response as the base URL for the following URI:

    ```
    /public/core/v3/Users/ResetPassword
    ```

roles GET

Version 3 resource.

Returns details for an organization's roles.

Use the baseApiUrl from the login response as the base URL for the following URI:

    ```
    /public/core/v3/roles
    ```

roles POST

Version 3 resource.

Creates a custom role.

Use the baseApiUrl from the login response as the base URL in the following URI:

```
/public/core/v3/roles
```

roles PUT

Version 3 resource.

Updates a custom role.

Use the baseApiUrl from the login response as the base URL in one of the following URIs:

- To add privileges to a custom role, use one of the following URIs:

```
/public/core/v3/roles/<id>/addPrivileges
/public/core/v3/roles/name/<name>/addPrivileges
```

- To remove privileges from a custom role, use one of the following URIs:

```
/public/core/v3/roles/<id>/removePrivileges
/public/core/v3/roles/name/<name>/removePrivileges
```

roles DELETE

Version 3 resource.

Deletes a custom role.

Use the baseApiUrl from the login response as the base URL for the following URI:

```
/public/core/v3/roles/<id>
```

runtimeEnvironment GET

Version 2 resource.

Returns the details of the runtime environments used by the organization. You can also get details about Secure Agent service properties for Secure Agent groups.

Use the serverUrl from the login response as the base URL for one of the following URIs:

- To get the details of the runtime environments used by the organization, use one of the following URIs:

```
/api/v2/runtimeEnvironment
/api/v2/runtimeEnvironment/<id>
/api/v2/runtimeEnvironment/<name>
```

- To find out which services and connectors are enabled for a Secure Agent group, use the following URI:

```
/api/v2/runtimeEnvironment/<id>/selections
```

- To get the enabled and disabled services and connectors for a Secure Agent group, use the following URI:

```
/api/v2/runtimeEnvironment/<id>/selections/details
```

- To get a list of editable Secure Agent service properties that you can configure for a Secure Agent group, use the following URI:

```
/api/v2/runtimeEnvironment/<id>/configs/details/<platform>
```

- To get a list of group-level properties that have overridden Secure Agent service default property settings, use the following URI:

```
/api/v2/runtimeEnvironment/<id>/configs/<platform>
```

runtimeEnvironment POST

Version 2 resource.

Creates or updates a Secure Agent group.

Use the serverUrl from the login response as the base URL for one of the following URIs:

```
/api/v2/runtimeEnvironment
/api/v2/runtimeEnvironment/<id>
```

runtimeEnvironment PUT

Version 2 resource.

Enables services and connectors and sets Secure Agent service properties for a Secure Agent group.

Use the serverUrl from the login response as the base URL for one of the following URIs:

- To enable services and connectors for a Secure Agent group, use the following URI:

```
/api/v2/runtimeEnvironment/<id>/selections
```

- To configure Secure Agent service properties for a Secure Agent group, use the following URI:

```
/api/v2/runtimeEnvironment/<id>/configs/<platform>
```

runtimeEnvironment DELETE

Version 2 resource.

Deletes a Secure Agent group or deletes Secure Agent service property settings for a Secure Agent group.

Use the serverUrl from the login response as the base URL for one of the following URIs:

- To delete a Secure Agent group, use the following URI:

```
/api/v2/runtimeEnvironment/<id>
```

- To delete property settings for a Secure Agent group, use the following URI:

```
/api/v2/runtimeEnvironment/<id>/configs
```

schedule GET

Version 3 resource.

Returns the details of a schedule or of all schedules in the organization.

Use the baseApiUrl from the login response as the base URL for one of the following URIs:

- To get the details for all schedules in the organization, use the following URI:

```
/public/core/v3/schedule
```

- To get the details for a specific schedule, use one of the following URIs:

```
/public/core/v3/schedule/<id>
/public/core/v3/schedule/<parameter>
```

schedule POST

Version 3 resource.

Creates a schedule.

Use the baseApiUrl from the login response as the base URL in the following URI:

```
/public/core/v3/schedule
```

schedule PATCH

Version 3 resource.

Updates a schedule.

Use the baseApiUrl from the login response as the base URL in the following URI:

```
/public/core/v3/schedule/<id>
```

schedule DELETE

Version 3 resource.

Deletes a schedule.

Use the baseApiUrl from the login response as the base URL in the following URI:

```
/public/core/v3/schedule/<id>
```

schedule GET

Version 2 resource.

Returns the details of a schedule or of all schedules in the organization.

Use the serverUrl from the login response as the base URL for one of the following URIs:

```
/api/v2/schedule/<id>
/api/v2/schedule/name/<name>
```

schedule POST

Version 2 resource.

Creates or updates a schedule.

Use the serverUrl from the login response as the base URL for the following URI:

```
/api/v2/schedule/<id>
```

Note: To leverage full scheduling capabilities, use the version 3 schedule resource instead of the version 2 schedule resource.

schedule DELETE

Version 2 resource.

Deletes a schedule.

Use the serverUrl from the login response as the base URL for the following URI:

```
/api/v2/schedule/<id>
```

securityLog GET

Version 3 resource.

Returns security log entries that include events such as login actions and permission changes.

Use the baseApiUrl from the login response as the base URL for the following URI:

```
/public/core/v3/securityLog
```

serverTime GET

Version 2 resource.

Returns the local time for the Informatica Intelligent Cloud Services server.

Use the serverUrl from the login response as the base URL for the following URI:

```
/api/v2/server/serverTime
```

sourceControlAction GET

Version 3 resource.

Returns the status of a source control operation for the specified object.

Use the baseApiUrl from the login response as the base URL for one of the following URIs:

- To get the status of a source control operation, use the following URI:

    ```
    /public/core/v3/sourceControlAction/<actionId>
    ```

- To get the status for each object in a source control operation, use the following URI:

    ```
    /public/core/v3/sourceControlAction/<actionId>?expand=objects
    ```

TagObjects POST

Version 3 resource.

Assigns tags to an asset.

Use the baseApiUrl from the login response as the base URL for the following URI:

```
/public/core/v3/TagObjects
```

task GET

Version 2 resource.

Returns a list of tasks of the specified type.

Use the serverUrl from the login response as the base URL for the following URI:

```
/api/v2/task?type=<type>
```

undoCheckout POST

Version 3 resource.

Reverses a checkout and reverts the object to its original state.

Use the baseApiUrl from the login response as the base URL for the following URI:

```
/public/core/v3/undoCheckout
```

UntagObjects POST

Version 3 resource.

Removes tags that were assigned to an asset.

Use the baseApiUrl from the login response as the base URL for the following URI:

```
/public/core/v3/UntagObjects
```

user GET

Version 2 resource.

Returns the details of an Informatica Intelligent Cloud Services user account or of all user accounts in the organization.

Use the serverUrl from the login response as the base URL for one of the following URIs:

```
/api/v2/user/<id>
/api/v2/user/name/<name>
```

Note: To leverage full user management capabilities, use the version 3 users resource instead of the version 2 user resource. The version 3 users resource supports users, user groups, and roles. The version 2 user resource does not support user groups and roles, and a GET request might not return all users in the organization.

user POST

Version 2 resource.

Creates or updates an Informatica Intelligent Cloud Services user account.

Use the serverUrl from the login response as the base URL for the following URI:

```
/api/v2/user/<id>
```

Note: We recommend that you use the version 3 users resource, instead of using the version 2 user resource. The version 2 user resource doesn't support user groups or user roles.

user DELETE

Version 2 resource.

Deletes an Informatica Intelligent Cloud Services user account.

Use the serverUrl from the login response as the base URL for the following URI:

```
/api/v2/user/<id>
```

userGroups GET

Version 3 resource.

Returns details for all user groups in the organization or the details for a particular user group.

Use the baseApiUrl from the login response as the base URL for the following URI:

```
/public/core/v3/userGroups
```

userGroups POST

Version 3 resource.

Creates an Informatica Intelligent Cloud Services user group.

Use the baseApiUrl from the login response as the base URL for one of the following URIs:

```
/public/core/v3/userGroups
/public/core/v3/userGroups/<id>
```

userGroups PUT

Version 3 resource.

Updates an Informatica Intelligent Cloud Services user group.

Use the baseApiUrl from the login response as the base URL for one of the following URIs:

- To add a role to a user group, use one of the following URIs:

```
/public/core/v3/userGroups/<id>/addRoles
/public/core/v3/userGroups/name/<name>/addRoles
```

- To remove a role from a user group, use one of the following URIs:

```
/public/core/v3/userGroups/<id>/removeRoles
/public/core/v3/userGroups/name/<name>/removeRoles
```

- To add a user to a user group, use one of the following URIs:

```
/public/core/v3/userGroups/<id>/addUsers
/public/core/v3/users/name/<name>/addGroups
```

- To remove a user from a user group, use one of the following URIs:

```
/public/core/v3/userGroups/<id>/removeUsers
/public/core/v3/users/name/<name>/removeGroups
```

userGroups DELETE

Version 3 resource.

Deletes an Informatica Intelligent Cloud Services user group.

Use the baseApiUrl from the login response as the base URL for the following URI:

```
/public/core/v3/userGroups/<id>
```

users GET

Version 3 resource.

Returns details for all users in the organization or the details for a particular user.

Use the baseApiUrl from the login response as the base URL for one of the following URIs:

```
/public/core/v3/users
/public/core/v3/users?<parameter>
```

users POST

Version 3 resource.

Creates an Informatica Intelligent Cloud Services user account.

Use the baseApiUrl from the login response as the base URL for the following URI:

```
/public/core/v3/users
```

users PUT

Version 3 resource.

Updates role and user group assignments for a user.

Use the baseApiUrl from the login response as the base URL for one of the following URIs:

- To assign a role to a user, use one of the following URIs:

```
/public/core/v3/users/<id>/addRoles
/public/core/v3/users/name/<name>/addRoles
```

- To remove a role from a user, use one of the following URIs:

```
/public/core/v3/users/<id>/removeRoles
/public/core/v3/users/name/<name>/removeRoles
```

- To assign a user group to a user, use one of the following URIs:

```
/public/core/v3/users/<id>/addGroups
/public/core/v3/users/name/<name>/addRGroups
```

- To remove a user group assignment, use one of the following URIs:

```
/public/core/v3/users/<id>/removeGroups
/public/core/v3/users/name/<name>/removeGroups
```

users DELETE

Version 3 resource.

Deletes an Informatica Intelligent Cloud Services user account.

Use the baseApiUrl from the login response as the base URL for the following URI:

```
/public/core/v3/users/<id>
```

Data Integration resource quick reference

The following list contains the syntax and a brief description of the Data Integration resources:

CodeTask POST

disnext version 1 resource.

Creates or starts a code task.

Use the server URL from the login response as the base URL for one of the following URIs:

- To create a code task, use the following URI:

```
/disnext/api/v1/CodeTask
```

- To start a code task, use the following URI:

```
/disnext/api/v1/CodeTask/Start
```

CodeTask GET

disnext version 1 resource.

Gets the details or the status of a code task.

Use the server URL from the login response as the base URL for one of the following URIs:

- To get the details of a code task, use the following URI:

    ```
    /disnext/api/v1/CodeTask/<codeTaskId>
    ```

- To get the status of a code task, use the following URI:

    ```
    /disnext/api/v1/CodeTask/JobStatus/<jobId>
    ```

CodeTask PUT

disnext version 1 resource.

Cancels a code task job.

Use the server URL from the login response as the base URL for the following URI:

    ```
    /disnext/api/v1/CodeTask/Cancel
    ```

connection GET

Version 2 resource.

Returns information related to connections in the organization.

Use the serverUrl from the login response as the base URL.

You can request the following information:

- Connection details. To request the details of a connection or of all connections in the organization, use one of the following URIs:

    ```
    /api/v2/connection
    /api/v2/connection/<id>
    /api/v2/connection/name/<name>
    ```

- Connection objects. To request a list of objects that you can use as a source or target for the specified connection, use one of the following URIs:

    ```
    /api/v2/connection/source/<id>
    /api/v2/connection/target/<id>
    ```

- Connection details by runtime environment. To request a list of all connections in the organization that use a particular runtime environment, use the following URI:

    ```
    /api/v2/connection/<runtimeEnvironmentId>
    ```

- Connections by Secure Agent and connection type. To request a list of connections by Secure Agent ID and connection type, use the following URI:

    ```
    /api/v2/connection/search?agentId=<id>&uiType=<uiType>
    ```

- Metadata details. To request metadata details for a connection, use the following URI:

    ```
    /api/v2/connection/<source or target>/<id>/metadata
    ```

- Test connection. To test a connection, use the following URI:

    ```
    /api/v2/connection/test/<id>
    ```

connection POST

Version 2 resource.

Creates or updates a connection.

Use the serverUrl from the login response as the base URL for the following URI:

    ```
    /api/v2/connection/<id>
    ```

Use a connection object to define attributes.

connection DELETE

Version 2 resource.

Deletes a connection.

Use the serverUrl from the login response as the base URL for the following URI:

```
/api/v2/connection/<id>
```

connector GET

Version 2 resource.

Returns a list of connectors available to the organization or attribute values for a specified connector type.

Use the serverUrl from the login response as the base URL for the following URI:

```
/api/v2/connector
/api/v2/connector/metadata?connectorType=<type>
```

customFunc GET

Version 2 resource.

Returns the details of a PowerCenter mapplet or of all PowerCenter mapplets in the organization.

Use the serverUrl from the login response as the base URL for one of the following URIs:

```
/api/v2/customFunc
/api/v2/customFunc/<id>
/api/v2/customFunc/name/<name>
```

customFunc POST

Version 2 resource.

Uploads a PowerCenter mapplet.

Use the serverUrl from the login response as the base URL for the following URI:

```
/api/v2/customFunc/<id>
```

Define attributes in the request body and encode the request body as multipart/form-data. Include the following required attributes: file, name.

customFunc DELETE

Version 2 resource.

Deletes a PowerCenter mapplet.

Use the serverUrl from the login response as the base URL for the following URI:

```
/api/v2/customFunc/<id>
```

dataPreview GET

Version 2 resource.

Use to preview data during mapping design. Specify the number of rows to return of source or target data for a specified object.

Use the serverUrl from the login response as the base URL for one of the following URIs:

```
/api/v2/connection/<source or target>/<connection ID>/datapreview/<object name>
/api/v2/connection/<source or target>/name/<name>/datapreview/<object name>
```

dynamictask GET

Batch-mapping version 1 resource.

Returns the details of a dynamic mapping task.

Use the serverUrl from the login response as the base URL for the following URI:

```
/batch-mapping/api/v1/dynamictask/<id>
```

dynamictask POST

Batch-mapping version 1 resource.

Creates a dynamic mapping task.

Use the serverUrl from the login response as the base URL for the following URI:

```
/batch-mapping/api/v1/dynamictask
```

dynamictask PUT

Batch-mapping version 1 resource.

Updates a dynamic mapping task.

Use the serverUrl from the login response as the base URL for the following URI:

```
/batch-mapping/api/v1/dynamictask/<id>
```

dynamictask DELETE

Batch-mapping version 1 resource.

Deletes a dynamic mapping task.

Use the serverUrl from the login response as the base URL for the following URI:

```
/batch-mapping/api/v1/dynamictask/<id>
```

expressionValidation POST

Version 2 resource.

Validates expressions and returns a success or error response.

Use the serverUrl from the login response as the base URL for the following URI:

```
/api/v2/expression/validate
```

field GET

Version 2 resource.

Returns the field details for a source or target object.

Use the serverUrl from the login response as the base URL for one of the following URIs:

```
/api/v2/connection/<source or target>/<id>/field/<name>
/api/v2/connection/<source or target>/name/<name>/field/<name>
/api/v2/connection/<source or target>/<id>/fields?objectName=<name>
```

field POST

Version 2 resource.

Updates the flat file attributes for a source or target object.

Use the serverUrl from the login response as the base URL for one of the following URIs:

```
/api/v2/connection/<source or target>/<id>/field/<name>
```

The flat file attributes provided in the request override the default attributes specified in the connection object.

filelisteners GET

mftsaas version 1 resource.

Gets file listener details and job status.

To get file listener details, use the serverUrl from the login response as the base URL for the following URI:

```
api/v1/filelisteners/<id>
```

To get the status of a file listener job, use the serverUrl from the login response as the base URL for the following URI:

```
mftsaas/api/v1/filelisteners/job/<id>/status
```

filelisteners POST

mftsaas version 1 resource.

Creates a file listener.

To create a file listener, use the serverUrl from the login response as the base URL for the following URI:

```
api/v1/filelisteners
```

To start a file listener, use the serverUrl from the login response as the base URL for the following URI:

```
mftsaas/api/v1/filelisteners/<id>/start
```

To stop a file listener , use the serverUrl from the login response as the base URL for the following URI:

```
mftsaas/api/v1/filelisteners/<id>/stop
```

filelisteners PUT

mftsaas version 1 resource.

Updates an existing file listener.

Use the serverUrl from the login response as the base URL for the following URI:

```
mftsaas/api/v1/filelisteners/<id>
```

filelisteners DELETE

mftsaas version 1 resource.

Deletes an existing file listener.

Use the serverUrl from the login response as the base URL for the following URI:

```
mftsaas/api/v1/filelisteners/<id>
```

fileRecord POST

Version 2 resource.

Upload a Visio template XML file or image file.

Use the serverUrl from the login response as the base URL for the following URI:

```
/api/v2/fileRecord
```

Define attributes in the request body and encode the request body as multipart/form-data. Include the following required attributes: file, name.

fileRecord DELETE

Version 2 resource.

Delete a Visio template file or image file.

Use the serverUrl from the login response as the base URL for the following URI:

```
/api/v2/fileRecord/<id>
```

fwConfig GET

Version 2 resource.

Returns the details of a fixed-width format.

Use the serverUrl from the login response as the base URL for one of the following URIs:

```
/api/v2/fwConfig/<id>
/api/v2/fwConfig/name/<name>
```

fwConfig POST

Version 2 resource.

Uploads a fixed-width format.

Use the serverUrl from the login response as the base URL for the following URI:

```
/api/v2/fwConfig/<id>
```

fwConfig DELETE

Version 2 resource.

Delete a fixed-width format.

Use the serverUrl from the login response as the base URL for the following URI:

```
/api/v2/fwConfig/<id>
```

mapping GET

Version 2 resource.

Returns the details of a mapping or of all mappings in the organization. Can also return an image of a mapping.

Use the serverUrl from the login response as the base URL for one of the following URIs:

```
/api/v2/mapping/<id>
/api/v2/mapping/name/<name>
/api/v2/mapping/search?name=<name>
/api/v2/mapping/<id>/image?deployed=<true|false>
```

masterTemplate GET

Version 2 resource.

Returns information about Visio templates. You can request the following information:

- Visio templates. You can request the details of a Visio template or of all Visio templates in the organization. Use the serverUrl from the login response as the base URL for one of the following URIs:

  ```
  /api/v2/masterTemplate/<id>
  /api/v2/masterTemplate/name/<name>
  ```

- Mapping tasks. You can request a list of mapping tasks that use a Visio template. Use the serverUrl from the login response as the base URL for the following URI:

  ```
  /api/v2/masterTemplate/<id>/tasks
  ```

masterTemplate POST

Version 2 resource.

Creates or updates a Visio template.

Use the serverUrl from the login response as the base URL for the following URI:

```
/api/v2/masterTemplate/<id>
```

Use a masterTemplate object to define attributes.

masterTemplate DELETE

Version 2 resource.

Deletes a Visio template.

Use the serverUrl from the login response as the base URL for the following URI:

```
/api/v2/masterTemplate/<id>
```

mttask GET

Version 2 resource.

Returns the details of a mapping task.

Use the serverUrl from the login response as the base URL for one of the following URIs:

```
/api/v2/mttask/<id>
/api/v2/mttask/frs/<federated task ID>
/api/v2/mttask/name/<name>
```

mttask POST

Version 2 resource.

Creates or updates a mapping task.

Use the serverUrl from the login response as the base URL for one of the following URIs:

```
/api/v2/mttask/<id>
/api/v2/mttask/frs/<federated task ID>
```

Use an mttask object to define attributes.

mttask DELETE

Version 2 resource.

Deletes a mapping task.

Use the serverUrl from the login response as the base URL for the following URI:

```
/api/v2/mttask/<id>
```

sendfiles POST

AS2 file transfer version 1 API.

Uses an AS2 connection to transfer files to a remote AS2 server.

Use the serverUrl from the login response as the base URL for the following URI:

```
mftsaas/api/v1/sendfiles/<connection name>
```

workflow GET

Version 2 resource.

Returns the details of a linear taskflow or of all linear taskflows in the organization.

Use the serverUrl from the login response as the base URL for one of the following URIs:

```
/api/v2/workflow/<id>
/api/v2/workflow/name/<name>
/api/v2/workflow/?simpleMode=true
```

workflow POST

Version 2 resource.

Creates or updates a linear taskflow.

Use the serverUrl from the login response as the base URL for the following URI:

```
/api/v2/workflow/<id>
```

Use a workflow object to define attributes.

workflow DELETE

Version 2 resource.

Deletes a linear taskflow.

Use the serverUrl from the login response as the base URL for the following URI:

```
/api/v2/workflow/<id>
```

Model Serve resource quick reference

The following list contains the syntax and a brief description of the actions you can perform with the Model Serve resources:

Get descriptions of all quick start models

To get a list of the names and descriptions of all of the quick start models, use the following URI:

```
/mlops/api/v1/modelhub/models
```

Monitor all quick start models

To get the statuses of and information about all of the quick start models, use the following URI:

```
/mlops/api/v1/modelhub/monitor
```

Get information about a quick start model

To get the status of and information about a single quick start model, use the following URI:

```
/mlops/api/v1/modelhub/preBuiltModel?name=<model name>
```

Use `TEXT_TRANSLATION` or `IMAGE_CLASSIFICATION` as the model name.

Get the code template file

To get the code template file that you need to define a machine learning model, use the following URI:

```
/mlops/api/v1/model/template/custom
```

Upload a model file

When you upload a model file, first generate a model file ID and then use the ID to upload the file.

To generate a model file ID, use the following URI:

```
/mlops/api/v1/model/upload/generateId
```

To upload or update a model file, use the following URI:

```
/mlops/api/v1/model/upload/<model file ID>
```

Register a machine learning model

Register a machine learning model after you upload the model file that defines the machine learning algorithm.

To register a machine learning model, use the following URI:

```
/frs/v1/Projects('<model ID>')/Documents
```

Edit a machine learning model

To edit a registered machine learning model, use the following URI:

```
/frs/v1/Documents('<ID>')
```

Download a model file

To download the model file from a registered machine learning model, use the following URI:

```
/mlops/api/v1/model/download/<ID>
```

Create a model deployment

To create a model deployment based on a registered machine learning model, use the following URI:

```
/frs/v1/Projects('<deployment ID>')/Documents
```

Edit a model deployment

To edit a model deployment, use the following URI:

```
/frs/v1/Documents('<deployment ID>')
```

Monitor deployments

To monitor the status of all model deployments in your organization, use the following URI:

```
/mlops/api/v1/deployment/monitor
```

To monitor the status of one model deployment, use the following URI:

```
/mlops/api/v1/deployment/monitor/<deployment ID>
```

Control a quick start model or model deployment

You can start a quick start model or model deployment to make it available for predictions, restart it while it's running, and stop it to release the resources.

To start a deployment, use the following URI:

```
/mlops/api/v1/deployment/control/<deployment ID>/start
```

To restart a deployment, use the following URI:

```
/mlops/api/v1/deployment/control/<deployment ID>/restart
```

To stop a deployment, use the following URI:

```
/mlops/api/v1/deployment/control/<deployment ID>/stop
```

Generate predictions

To generate predictions from an available quick start model or model deployment, use the following URI:

```
/mlops/api/v1/deployment/request/<deployment ID>
```

Delete assets

To delete a machine learning model or model deployment, use the following URI:

```
/frs/api/v1/Documents('<ID>')
```

Index