



Informatica™

Informatica® Multidomain MDM

10.4 HotFix 3

Guía de servicios de entidad de negocio

Informatica Multidomain MDM Guía de servicios de entidad de negocio
10.4 HotFix 3
Octubre 2021

© Copyright Informatica LLC 2014, 2022

Este software y la documentación se proporcionan exclusivamente en virtud de un acuerdo de licencia independiente que contiene restricciones de uso y divulgación. Ninguna parte de este documento puede ser reproducida o transmitida de cualquier forma o manera (electrónica, fotocopia, grabación o mediante otros métodos) sin el consentimiento previo de Informatica LLC.

Las bases de datos, el software y los programas de DERECHOS DEL GOBIERNO DE LOS ESTADOS UNIDOS, y la documentación e información técnica relacionadas entregadas a los clientes del Gobierno de los Estados Unidos constituyen "software informático comercial" o "datos técnicos comerciales" de acuerdo con el Reglamento de Adquisición Federal y las regulaciones complementarias específicas del organismo que correspondan. Como tales, el uso, la duplicación, la divulgación, la modificación y la adaptación están sujetos a las restricciones y los términos de licencia establecidos en el contrato gubernamental aplicable, y hasta donde sea aplicable en función de los términos del contrato gubernamental, a los derechos adicionales establecidos en FAR 52.227-19, Licencia de Software Informático Comercial.

Informatica, el logotipo de Informatica y ActiveVOS son marcas comerciales o marcas comerciales registradas de Informatica LLC en Estados Unidos y en las diversas jurisdicciones de todo el mundo. La lista actual de marcas comerciales de Informatica está disponible en Internet en <https://www.informatica.com/trademarks.html>. Otros nombres de productos y empresas pueden ser nombres o marcas comerciales de sus respectivos titulares.

Las partes de este software o la documentación están sujetas a derechos de autor de terceros. Se incluyen con el producto los avisos obligatorios de terceros.

La información contenida en esta documentación está sujeta a cambios sin previo aviso. Si encuentra algún problema en esta documentación, escríbanos a infa_documentation@informatica.com para notificarnoslo.

Los productos de Informatica gozan de garantía en función de los términos y condiciones de los acuerdos conforme a los cuales se proporcionen. **INFORMATICA PROPORCIONA LA INFORMACIÓN DE ESTE DOCUMENTO "TAL CUAL" SIN GARANTÍA DE NINGÚN TIPO, EXPRESA O IMPLÍCITA, INCLUIDAS LAS GARANTÍAS DE COMERCIALIZACIÓN, ADAPTACIÓN A UN FIN PARTICULAR Y CUALQUIER GARANTÍA O CONDICIÓN DE NO INCUMPLIMIENTO.**

Fecha de publicación: 2022-05-05

Tabla de contenido

Prefacio	10
Recursos de Informatica	10
Informatica Network.	10
Base de conocimiento de Informatica.	10
Documentación de Informatica.	11
Matrices de disponibilidad de producto de Informatica.	11
Informatica Velocity.	11
Catálogo de soluciones de Informatica.	11
Servicio internacional de atención al cliente de Informatica.	11
Capítulo 1: Introducción a los servicios de entidad de negocio.....	12
Resumen de los servicios de entidad de negocio.	12
Servicios de entidad de negocio.	13
Servicio de entidad de negocio ReadBE.	13
Servicio de entidad de negocio WriteBE.	13
Servicio de entidad de negocio SearchBE.	14
Extremos del servicio de entidad de negocio.	14
Extremo EJB para servicios de entidad de negocio.	14
Extremo REST para servicios de entidad de negocio.	14
Llamadas del servicio de entidad de negocio de REST y EJB.	15
Extremo SOAP para servicios de entidad de negocio.	15
Identificar un registro raíz.	15
Filtros de datos y seguridad.	16
Autenticación basada en certificados.	16
Capítulo 2: Llamadas del servicio de entidad de negocio de EJB.....	18
Descripción de las llamadas del servicio de entidad de negocio de EJB.	18
Ejemplo de código Java con clases SDO estándares.	18
Ejemplo de código Java con clases SDO generadas.	22
Capítulo 3: Llamadas del servicio de entidad de negocio de REST.....	27
Resumen de las API de REST para servicios de entidad de negocio.	27
Métodos REST compatibles.	28
Método de autenticación.	28
Cookies de autenticación para el inicio de sesión desde aplicaciones de terceros.	29
Obtener la cookie de autenticación y el token CSRF de Informatica.	29
Crear un registro de persona.	30
Establecer el tiempo de espera de la sesión.	30
Obtener información de usuario en la solicitud de inicio de sesión.	31
Actualizar una sesión.	31

Validar una sesión.	31
Cerrar sesión en una sesión.	31
Swagger.	32
Localizador uniforme de recursos de REST.	32
Configuración de encabezado y cuerpo.	33
Encabezado de solicitud.	33
Cuerpo de la solicitud.	34
Parámetros de consulta estándar.	35
Formatos de fecha y hora UTC.	36
Configuración de WebLogic para ejecutar llamadas REST de servicios de entidad de negocio.	37
Visualización de parámetros de entrada y salida.	38
Plantilla de JavaScript.	38
Ejemplo de JavaScript.	39
Ejemplo de llamada AJAX con autenticación básica HTTP.	41
Referencia de API de REST para servicios de entidad de negocio	41
Obtener metadatos	42
Lista de metadatos.	45
Enumerar columnas de coincidencia.	50
Leer registros.	51
Crear registros.	58
Actualizar registros.	61
Eliminar registros.	65
Lista Registro.	66
Búsqueda Registro.	69
Proveedor de sugerencias.	75
SearchQuery.	76
SearchMatch.	81
Obtener metadatos de BPM.	86
Lista de tareas.	87
Leer tarea.	93
Crear tarea.	95
Actualizar tarea.	97
Tarea completada.	100
Ejecutar acción de tarea.	103
Lista de usuarios asignables.	105
Reclamación masiva de tareas.	106
Tareas de liberación masiva.	108
Asignación masiva de tareas.	109
Edición masiva de tareas.	111
Acción masiva de tareas.	112
Obtener acciones de tarea.	114
Enumerar posibles propietarios de tareas.	116

Enumerar posibles propietarios de una tarea.	117
Enumerar metadatos de archivo.	119
Crear metadatos de archivo.	120
Obtener metadatos de archivo.	121
Actualizar metadatos de archivo.	121
Cargar contenido de archivo.	122
Obtener contenido de archivo.	123
Suprimir archivo.	124
Vista previa de informes reemplazados.	125
Actualizar Buscar y reemplazar.	129
Importar archivo nuevo.	131
Importar archivo coincidente	132
Obtener propiedades de archivo.	133
Guardar propiedades de archivo.	134
Devolver propiedades de archivo.	136
Vista previa de archivos analizados.	137
Obtener archivos analizados.	138
Errores de análisis al importar archivos.	139
Errores de carga al importar archivos.	140
Vista previa de promoción.	141
Promover.	143
Eliminar pendiente.	143
Vista previa de fusión.	144
Actualizar fusión pendiente.	147
Fusión pendiente.	150
Promover fusión.	151
Transformación de archivos.	151
Sugerir asignación.	153
Fusionar registros.	155
Anular fusión de registros.	157
Crear asignación.	158
Vista previa de asignación.	160
Buscar asignación.	162
Leer asignación.	164
Actualizar asignación.	165
Transformación de archivos.	167
Leer una relación.	169
Crear una relación.	172
Actualizar una relación.	173
Eliminar una relación.	175
Obtener registros relacionados.	175
Exportar entidades de negocio relacionadas.	179

Lista Jerarquías.	180
Obtener metadatos de jerarquía.	181
Obtener ruta de jerarquía.	186
Obtener elementos principales.	188
Obtener elemento secundario.	190
Exportar jerarquía.	197
Exportar elementos principales y secundarios directos.	199
Obtener cambios de jerarquía.	200
Cambios masivos de relación.	206
Promoción masiva.	209
Rechazo masivo.	214
Iniciar coincidencias.	218
Leer registros coincidentes.	219
Actualizar registros coincidentes.	220
Eliminar registros coincidentes.	221
Obtener datos de coincidencia en CSV.	222
Obtener datos de coincidencia en JSON.	224
Importar archivo coincidente	225
Obtener metadatos de sistema de origen	226
Obtener eventos de historial de registros.	228
Obtener detalles de evento.	231
Lista Informes.	233
Obtener configuración y datos del informe.	235
Obtener configuración del informe e informes de detalle.	237
Registrar informe.	238
Actualizar la configuración del informe.	240
Agregar o actualizar datos de informe.	241
Eliminar informe.	242
Ejecutar el trabajo de actualización de informe.	243
Obtener estado de trabajo de actualización de informe.	243
Enumerar grupos de trabajos	244
Enumerar un grupo de trabajos.	246
Obtener metadatos de DaaS.	248
Búsqueda de DaaS.	249
Lectura de DaaS.	253
WriteMerge.	255
Importación de DaaS.	256
Actualización de DaaS.	259

Capítulo 4: API REST para Data Director 262

API REST para cambiar la marca de Data Director.	262
Cargar archivo de fondo de inicio de sesión	262
Delete Login BG File.	263

Upload Logo File.	264
Eliminar archivo de logotipo.	264
Obtener variables.	265
Actualizar variables.	266
Eliminar esquema de color.	271
API REST para configurar la exención de responsabilidad legal.	271
Parámetros de descargo de responsabilidad legal.	272
Read Legal Message Configuration.	272
Establecer la configuración de mensajes legales.	273
Eliminar la configuración de mensajes legales.	274

Capítulo 5: Llamadas del servicio de entidad de negocio de SOAP..... 275

Descripción de las llamadas SOAP para servicios de entidad de negocio.	275
Método de autenticación.	276
Cookies de autenticación para el inicio de sesión desde aplicaciones de terceros.	276
Obtener la cookie de autenticación y el token CSRF de Informatica.	277
Ejemplo de SOAP con cookie de autenticación y token CSRF de Informatica.	277
Archivo de lenguaje de descripción de servicios web.	278
URL de SOAP.	278
Respuestas y solicitudes SOAP.	279
Visualización de parámetros de entrada y salida.	280
Referencia de API de SOAP.	281
Respuestas y solicitudes SOAP de ejemplo.	282
Lista de usuarios asignables Muestra.	282
Entidad de negocio de lectura Muestra.	283
Entidad de negocio de creación Muestra.	284

Capítulo 6: Servicios de cálculos de BVT y registros de referencias cruzadas..... 285

Descripción general de los servicios de cálculos de BVT y registros de referencias cruzadas.	285
Obtención de datos de referencias cruzadas e investigación de cálculos de BVT.	285
Obtener registros de referencias cruzadas.	286
Determinar colaboradores del registro principal.	286
Obtener las puntuaciones de confianza de campos de registro de referencias cruzadas colaboradores.	287
Obtención de las puntuaciones de confianza de todos los campos de registro de referencias cruzadas.	288
Obtener información sobre los sistemas de origen.	289
Ejemplo de obtención de información sobre los sistemas de origen.	289
Filtrado y paginación de respuestas.	290
Ejemplos de solicitudes de filtrado.	290
Establecer la mejor versión de confianza.	290
Seleccionar el campo colaborador correcto.	290

Ejemplo de selección del campo colaborador correcto.	291
Escribir el valor correcto en el registro principal.	291
Ejemplo de escritura del valor correcto en el registro principal.	293
Eliminar los datos de origen no coincidentes.	294
Ejemplo de eliminación de datos de origen no coincidentes.	295
Respuesta de anulación de fusión.	296
Capítulo 7: Compatibilidad del servicio de vinculación empresarial.	297
Resumen.	297
Servicios de entidad de negocio para la importación y actualización de DaaS.	297
Configuración de la compatibilidad de la vinculación.	298
Aplicación personalizada para la división de datos de vinculación.	298
Capítulo 8: Llamadas externas para limpiar, analizar y transformar datos.	299
Resumen.	299
Fases de servicio.	300
Parámetros del método de proceso.	302
Prácticas recomendadas para llamadas externas.	303
Creación de llamadas externas.	304
Ejemplo de llamada externa.	304
Paso 1. Comprender e implementar el servicio web de ejemplo.	308
Paso 2. Iniciar sesión en la herramienta de aprovisionamiento.	308
Paso 3. Cargar el archivo WSDL de muestra.	309
Paso 4. Registrar el servicio de SOAP.	309
Paso 5. Configurar la llamada externa.	310
Paso 6. Publicar las configuraciones.	311
Probar la validación de direcciones.	311
Probar la lógica de fusión.	311
Obtención de ID de fila de origen con la llamada externa MergeCO.BeforeEverything.	316
Ejemplo: Validación y lógica personalizadas para servicios de entidad de negocio.	319
Requisitos previos.	319
Paso 1. Probar la validación personalizada.	319
Paso 2. Probar la lógica personalizada.	320
Ejemplo: realizar una llamada al marco de integración de servicios desde una llamada externa.	324
Apéndice A: Uso de las API de REST para añadir registros.	328
Resumen sobre el uso de las API de REST para añadir registros.	328
Estructura de una entidad de negocio de persona.	329
Paso 1. Obtener información sobre el esquema.	329
Obtener la respuesta de los metadatos.	330
Paso 2. Crear un registro.	335
Crear respuestas de registro.	336
Paso 3. Leer el registro.	337

Leer la respuesta del registro.	338
Apéndice B: Uso de las API de REST para cargar archivos.	342
Resumen sobre el uso de las API de REST para cargar archivos.	342
API de REST para archivos.	343
Componentes de archivo.	343
Tipos de almacenamiento.	344
Adjuntar archivos a registros.	345
Adjuntar archivos a tareas.	346
Cargar archivos de paquete de recursos.	349
Apéndice C: Uso de las API de REST para administrar informes.	350
Resumen sobre el uso de las API de REST para administrar informes.	350
API de REST para informes.	351
Configuración del informe.	351
Datos del informe.	352
Informes de detalle.	353
Informes de fábrica.	354
Administrar informes de fábrica.	355
Informes personalizados.	357
Administrar informes personalizados.	357
Administrar informes personalizados con informes de detalle.	359
Solución de problemas con API de informes.	363
Índice.	364

Prefacio

Consulte Informatica® *Guía de servicios de entidad de negocio de Multidomain MDM* para obtener información acerca de los servicios de entidad de negocio disponibles como servicios web. Use la guía para aprender a usar Enterprise JavaBeans (EJB), Representational State Transfer (REST), Simple Object Access Protocol (SOAP) y llamadas de servicios web externos para realizar operaciones sobre los datos de la entidad de negocio. También puede aprender a configurar interfaces de usuario personalizadas para realizar llamadas de servicio de la entidad de negocio.

Recursos de Informatica

Informatica proporciona una variedad de recursos de productos a través de Informatica Network y otros portales en línea. Use los recursos para sacar el mayor provecho de los productos y las soluciones de Informatica y aprender de otros expertos en la materia y usuarios de Informatica.

Informatica Network

Informatica Network es la puerta de entrada a muchos recursos, entre ellos, la base de conocimientos de Informatica y el servicio internacional de atención al cliente de Informatica. Para entrar en Informatica Network, visite <https://network.informatica.com>.

Como miembro de Informatica Network, tiene las siguientes opciones:

- Buscar recursos de productos en la base de conocimientos
- Ver la información de disponibilidad del producto
- Crear y revisar casos de soporte
- Buscar su red de grupos de usuarios de Informatica locales y colaborar con sus pares

Base de conocimiento de Informatica

Use la base de conocimientos de Informatica para encontrar recursos de productos como artículos prácticos, procedimientos recomendados, tutoriales de video y respuestas a preguntas frecuentes.

Para buscar en la base de conocimiento, visite <https://search.informatica.com>. Si tiene preguntas, comentarios o ideas relacionadas con la base de conocimiento de Informatica, póngase en contacto con el equipo de la base de conocimiento de Informatica en KB_Feedback@informatica.com.

Documentación de Informatica

Use el portal de documentación de Informatica para recorrer una extensa biblioteca de documentación para las versiones de productos actuales y recientes. Para recorrer el portal de documentación, visite <https://docs.informatica.com>.

Si tiene preguntas, comentarios o ideas acerca de la documentación de los productos, póngase en contacto con el equipo de la documentación de Informatica en infa_documentation@informatica.com.

Matrices de disponibilidad de producto de Informatica

Las matrices de disponibilidad de producto (PAM, Product Availability Matrixes) indican las versiones de sistemas operativos, bases de datos y otros tipos de orígenes y destinos de datos admitidos por la versión de un producto. Puede recorrer las PAM de Informatica en <https://network.informatica.com/community/informatica-network/product-availability-matrices>.

Informatica Velocity

Informatica Velocity es una colección de consejos y procedimientos recomendados desarrollados por los servicios profesionales de Informatica que se basan en experiencias reales de cientos de proyectos de administración de datos. Informatica Velocity representa el conocimiento colectivo de los consultores de Informatica que trabajan con organizaciones de todo el mundo para planificar, desarrollar, implementar y dar mantenimiento a soluciones de administración de datos exitosas.

Puede encontrar recursos de Informatica Velocity en <http://velocity.informatica.com>. Si tiene alguna pregunta, comentario o idea acerca de Informatica Velocity, póngase en contacto con los servicios profesionales de Informatica en ips@informatica.com.

Catálogo de soluciones de Informatica

El catálogo de soluciones de Informatica es un foro donde puede buscar soluciones que aumenten, amplíen o mejoren sus implementaciones de Informatica. Aproveche cualquiera de los cientos de soluciones de socios y desarrolladores de Informatica que se encuentran en el catálogo para mejorar su productividad y acelerar la implementación de los proyectos. Puede encontrar el catálogo de soluciones de Informatica en <https://marketplace.informatica.com>.

Servicio internacional de atención al cliente de Informatica

Puede ponerse en contacto con un centro de atención global por teléfono o a través de Informatica Network.

Para encontrar el número de teléfono local del servicio internacional de atención al cliente de Informatica, visite el sitio web de Informatica en el siguiente vínculo:

<https://www.informatica.com/services-and-training/customer-success-services/contact-us.html>.

Para buscar recursos de asistencia en línea en Informatica Network, visite <https://network.informatica.com> y seleccione la opción eSupport.

CAPÍTULO 1

Introducción a los servicios de entidad de negocio

Este capítulo incluye los siguientes temas:

- [Resumen de los servicios de entidad de negocio, 12](#)
- [Servicios de entidad de negocio, 13](#)
- [Extremos del servicio de entidad de negocio, 14](#)
- [Identificar un registro raíz, 15](#)
- [Filtros de datos y seguridad , 16](#)
- [Autenticación basada en certificados, 16](#)

Resumen de los servicios de entidad de negocio

Un servicio de entidad de negocio es un conjunto de operaciones que ejecutan código de MDM Hub para crear, actualizar, eliminar y buscar registros de objeto base en una entidad de negocio. Puede desarrollar una interfaz de usuario personalizada que pueda ejecutar código Java o JavaScript para realizar llamadas de servicios de entidad de negocio.

Por ejemplo, puede crear un servicio de entidad de negocio para aumentar un registro de proveedor con datos de Dun and Bradstreet. Configure el servicio de entidad de negocio para tomar el registro de proveedor como una entrada, recuperar alguna información de Dun and Bradstreet, actualizar el registro y, después, generar el registro de proveedor actualizado.

Los objetos base de una entidad de negocio tienen los siguientes servicios de entidad de negocio:

Lectura

Cada entidad de negocio tiene un servicio de entidad de negocio para realizar una operación de lectura.

Escritura

Cada entidad de negocio tiene un servicio de entidad de negocio para realizar una operación de escritura.

Búsqueda

Cualquier entidad de negocio que tenga campos que admiten búsquedas tiene un servicio de entidad de negocio para realizar una operación de búsqueda.

Por ejemplo, una entidad de negocio de persona tiene campos que admiten búsquedas. MDM Hub genera un servicio de entidad de negocio ReadPerson, WritePerson y SearchPerson. Los pasos de los servicios de

entidad de negocio de lectura, escritura y búsqueda le permiten leer, crear, actualizar, eliminar y buscar registros en un objeto compuesto.

Servicios de entidad de negocio

Un servicio de entidad de negocio realiza una operación. Puede usar los servicios de entidad de negocio ReadBE, WriteBE y SearchBE.

Un servicio de entidad de negocio cuenta con pasos de servicio. Una solicitud entrante pasa a través de cada paso de servicio. La salida de un paso es una entrada del paso siguiente. La salida de un paso puede pasar información a la entrada del siguiente paso. Todos los pasos de servicio de entidad de negocio se ejecutan como una llamada Enterprise JavaBean (EJB) en una sola transacción. MDM Hub gestiona las excepciones.

Nota: Antes de usar los servicios de entidad de negocio, debe validar el Almacén de referencias operativas.

Servicio de entidad de negocio ReadBE

El servicio de entidad de negocio ReadBE lee los datos de un registro de objeto base en una entidad de negocio.

Puede especificar los parámetros de paginación con el paso ReadBE para establecer el número de registros que se van a devolver y la página de resultados que se van a ver.

Los resultados del servicio ReadBE no incluyen los registros que se hayan eliminado temporalmente.

Si no pasa el parámetro EffectiveDate en la solicitud de servicio de entidad de negocio, MDM Hub da por supuesto un valor NULL para la fecha de vigencia y el servicio de entidad de negocio lee los datos del objeto base. Si pasa el parámetro EffectiveDate, MDM Hub calculará la mejor versión de la verdad a partir de registros de referencias cruzadas y el servicio de entidad de negocio de lectura devuelve la mejor versión actualizada de la verdad.

Servicio de entidad de negocio WriteBE

El servicio de entidad de negocio WriteBE puede actualizar los datos de un elemento de entidad de negocio, crear un elemento secundario de entidad de negocio o eliminar elementos secundarios de entidad de negocio.

Nota: El servicio de entidad de negocio WriteBE utiliza la configuración de confianza existente para calcular la confianza en los objetos base. Con este servicio no puede realizar un reemplazo de confianza.

Parámetros opcionales

En la siguiente tabla se describen los parámetros opcionales que puede usar con el servicio de entidad de negocio WriteBE:

Parámetro	Descripción
recordState	Establezca el estado del registro en ACTIVE, PENDING o DELETED. Nota: Cuando se establece <code>recordState=ACTIVE</code> y se ejecuta el servicio en un registro eliminado temporalmente, el servicio devuelve el registro al estado activo.
EffectivePeriod	Especifique un período efectivo. Si no pasa el parámetro <code>EffectivePeriod</code> , MDM Hub asumirá que es un periodo sin enlazar. MDM Hub no comprueba que exista una alineación para los periodos efectivos entre los objetos raíz y los objetos secundarios. Al crear o actualizar los registros, asegúrese de los períodos efectivos del registro principal y los registros secundarios estén alineados.

Servicio de entidad de negocio SearchBE

Utilice el servicio de entidad de negocio SearchBE para buscar un registro raíz en una entidad de negocio.

Para obtener información sobre la configuración de entidades de negocio para Search, consulte la *Guía de configuración de Multidomain MDM*.

Extremos del servicio de entidad de negocio

Es posible acceder a los servicios de entidad de negocio a través de un extremo Enterprise JavaBeans (EJB), un extremo de transferencia de estado representacional (REST) o un extremo de Protocolo simple de acceso a objetos (SOAP).

Los extremos REST se basan en una perspectiva de EJB. La configuración de los servicios de entidad de negocio de REST define cómo se asignan las URL de REST a las llamadas de servicio de entidad de negocio EJB.

Extremo EJB para servicios de entidad de negocio

El extremo Enterprise JavaBeans (EJB) es el extremo que subyace para todos los tipos de llamadas de servicios de entidad de negocio. Todos los demás extremos se asignan al extremo EJB.

Los servicios de entidad de negocio se muestran como un EJB sin estado. Un contenedor EJB sin estado puede agrupar instancias, asignar instancias y aplicar estrategias de equilibrio de cargas para distribuir la carga en diferentes servidores dentro del dominio.

Un extremo EJB acepta un nombre de usuario y una contraseña para autenticación.

Extremo REST para servicios de entidad de negocio

Con las llamadas de extremo de transferencia de estado representacional (REST), todos los servicios de entidad de negocio estarán disponibles como servicios web.

MDM Hub usa Swagger para enumerar todos los servicios web de REST con URL y parámetros.

Puede acceder a Swagger con la siguiente URL:

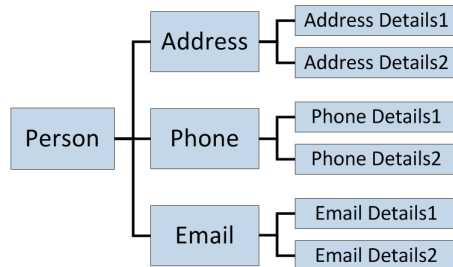
```
http://<host>:<puerto>/cmx/swagger-ui.html
```

Llamadas del servicio de entidad de negocio de REST y EJB

Cuando realice una llamada de servicio de entidad de negocio, puede especificar determinadas sucursales secundarias en lugar de solicitar la entidad de negocio completa.

Por ejemplo, desea realizar una operación de lectura en una entidad empresarial que tenga un nodo raíz de persona y varias sucursales secundarias. El objeto base de persona contiene los objetos base secundarios de dirección, teléfono, y correo electrónico. Cada objeto base secundario tiene dos objetos base secundarios de segundo nivel.

La siguiente imagen muestra la estructura de una entidad empresarial con varias sucursales:



Puede leer desde varias sucursales secundarias situadas en distintos niveles en una única solicitud. Por ejemplo, puede leer la persona, el teléfono, los detalles de teléfono 1, los detalles de teléfono 2, el correo electrónico y los detalles del correo electrónico 2 con una única solicitud.

El siguiente ejemplo de URL muestra cómo es posible hacer que una solicitud de lectura de REST obtenga el registro Persona con el ID de fila 1242, además de los registros secundarios Detalles de dirección 1 y Correo electrónico:

```
http://localhost:8080/cmx/cs/localhost-ORCL-DS_UI1/Person/1242?children=Address/  
Address_Details_1,Email
```

Extremo SOAP para servicios de entidad de negocio

Las llamadas de extremo de Protocolo simple de acceso a objetos (SOAP) hacen que los servicios de entidad de negocio estén disponibles como servicios web.

Los archivos de lenguaje de descripción de servicios web (WSDL) contienen las descripciones XML de los servicios web, los formatos de las respuestas y las solicitudes SOAP y todos los parámetros. MDM Hub genera un archivo WSDL para cada Almacén de referencias operativas.

Identificar un registro raíz

Puede usar uno de los siguientes métodos para identificar un registro raíz:

- rowid. Valor de la columna ROWID_OBJECT del registro.
- systemName y sourceKey. systemName es el nombre del sistema al que pertenece el registro. sourceKey es el valor de la columna PKEY_SRC_OBJECT del registro.

- Identificador de negocio global (GBID) de un objeto. Un GBID puede ser un valor compuesto, en cuyo caso debe pasar todos los valores.

Nota: El método GBID funciona únicamente con el servicio ReadBE.

En el siguiente código de ejemplo se usan `systemName` y `sourceKey` para identificar un registro:

```
String systemName = "SFA";

Properties config = new Properties();
config.put(SiperianClient.SIPERIANCLIENT_PROTOCOL, EjbSiperianClient.PROTOCOL_NAME);
CompositeServiceClient client = CompositeServiceClient.newCompositeServiceClient(config);
CallContext callContext = new CallContext(orsId, user, pass);
helperContext = client.getHelperContext(callContext);
DataFactory dataFactory = helperContext.getDataFactory();

//String personRowId = "1097";
String pkeySrcObject = "CST1379";

//Set custom key pkey
pkey = (Key) dataFactory.create(Key.class);
pkey.setSystemName(systemName);
pkey.setSourceKey(val);
writePerson.setKey(pkey);
```

Filtros de datos y seguridad

Cuando los objetos base y los recursos tienen privilegios de funciones de usuario, la entidad de negocio hereda esos privilegios. Para acceder a los registros de entidad de negocio, la función de usuario debe tener los privilegios necesarios para el objeto base raíz de la entidad de negocio, así como de otros recursos.

Los servicios de entidad de negocio también heredan los filtros de datos que se establecen en los campos de la entidad de negocio.

Si desea más información sobre los filtros de datos y la seguridad, consulte la *Guía de la herramienta de aprovisionamiento de Multidomain MDM*.

Autenticación basada en certificados

MDM Hub usa un mecanismo de autenticación basado en certificados para garantizar la autenticación entre los componentes de MDM Hub y las aplicaciones de confianza. El mecanismo de autenticación también se admite para las API de servicios de entidad de negocio.

Las aplicaciones cliente externas pueden realizar solicitudes de servicio de entidad de negocio a MDM Hub. Debe registrar a los usuarios de aplicaciones cliente externas como usuarios de aplicaciones de confianza en MDM Hub. También debe registrar el certificado público para los usuarios asociados con la aplicación cliente externa.

Después del registro, los usuarios de clientes externos pueden enviar solicitudes de autenticación cifradas a MDM Hub usando la clave privada. MDM Hub descifra la solicitud de autenticación usando la clave pública y responde con los datos solicitados en texto sin formato.

La autenticación de usuario solicita que la carga de trabajo de servicios de entidad de negocio se cifre usando la clave privada y se descifre usando la clave pública.

Nota: Para registrar la clave pública y el certificado con MDM Hub, póngase en contacto con el servicio internacional de atención al cliente de Informatica.

Para obtener más información acerca de cómo configurar la autenticación basada en certificados y aplicaciones de confianza, consulte el capítulo Autenticación basada en certificados en la Guía de Seguridad de Informatica Multidomain MDM 10.4.

CAPÍTULO 2

Llamadas del servicio de entidad de negocio de EJB

Este capítulo incluye los siguientes temas:

- [Descripción de las llamadas del servicio de entidad de negocio de EJB, 18](#)
- [Ejemplo de código Java con clases SDO estándares, 18](#)
- [Ejemplo de código Java con clases SDO generadas, 22](#)

Descripción de las llamadas del servicio de entidad de negocio de EJB

Puede realizar llamadas del servicio de entidad de negocio Enterprise JavaBean (EJB) para crear, actualizar, eliminar y buscar registros de objetos base en una entidad de negocio. Puede crear código Java para ejecutar llamadas de servicios de entidad de negocio EJB.

Puede crear el código Java en función de las clases estándar de Objetos de servicio de datos (SDO) o bien puede crearlo en función de las clases java que genera MDM Hub según la configuración de la entidad de negocio y de los servicios de entidad de negocio.

Ejemplo de código Java con clases SDO estándares

En el ejemplo se muestra el código Java para ejecutar llamadas Enterprise JavaBean (EJB) basadas en las clases de objetos de servicio de datos (SDO) estándar.

El ejemplo se encuentra en el siguiente archivo del kit de recursos: C:\<directorio de instalación de MDM Hub>\hub\resourcekit\samples\COS\source\java\com\informatica\mdm\sample\cs\DynamicSDO.java

El siguiente código Java se basa en las clases SDO estándar y ejecuta llamadas de servicio de entidad de negocio EJB para crear un registro de objeto base de persona, agregar varios registros secundarios, eliminar un registro secundario y, a continuación, eliminar el registro de persona y todos los registros secundarios:

```
package com.informatica.mdm.sample.cs;

import com.informatica.mdm.cs.CallContext;
import com.informatica.mdm.cs.api.CompositeServiceException;
import com.informatica.mdm.cs.client.CompositeServiceClient;
```

```

import com.siperian.sif.client.EjbSiperianClient;
import com.siperian.sif.client.SiperianClient;
import commonj.sdo.DataObject;
import commonj.sdo.Property;
import commonj.sdo.Type;
import commonj.sdo.helper.DataFactory;
import commonj.sdo.helper.HelperContext;

import java.io.PrintStream;
import java.util.Arrays;
import java.util.Properties;

public class DynamicSDO {

    public static void main(String[] args) throws CompositeServiceException {

        if(args.length != 3) {
            System.err.println("USAGE: DynamicSDO <ors> <user> <pass>");
            return;
        }

        new DynamicSDO(args[0], args[1], args[2]).execute();
    }

    private String orsId;
    private String user;
    private String pass;
    private HelperContext helperContext;
    private PrintStream out = System.out;

    public DynamicSDO(String orsId, String user, String pass) {
        this.orsId = orsId;
        this.user = user;
        this.pass = pass;
    }

    public void execute() throws CompositeServiceException {

        String systemName = "Admin";

        Properties config = new Properties();
        config.put(SiperianClient.SIPERIANCLIENT_PROTOCOL,
EjbSiperianClient.PROTOCOL_NAME);
        CompositeServiceClient client =
CompositeServiceClient.newCompositeServiceClient(config);

        CallContext callContext = new CallContext(orsId, user, pass);

        helperContext = client.getHelperContext(callContext);

        DataFactory dataFactory = helperContext.getDataFactory();

        // types for Read requests
        Type coFilterType = helperContext.getTypeHelper().getType("urn:cs-
base.informatica.mdm", "CoFilter");
        Type coFilterNodeType = helperContext.getTypeHelper().getType("urn:cs-
base.informatica.mdm", "CoFilterNode");
        Type keyType = helperContext.getTypeHelper().getType("urn:cs-
base.informatica.mdm", "Key");

        // ReadCO & WriteCO request types
        Type readPersonType = helperContext.getTypeHelper().getType("urn:cs-
ors.informatica.mdm", "ReadPerson");
        Type writePersonType = helperContext.getTypeHelper().getType("urn:cs-
ors.informatica.mdm", "WritePerson");

        // 1. Create new person
        DataObject createPerson = dataFactory.create(writePersonType);
        DataObject createPersonParameters = createPerson.createDataObject("parameters");
        createPersonParameters.setString("systemName", systemName);

```

```

DataObject person = createPerson.createDataObject("object");

person.getChangeSummary().beginLogging();

DataObject personRoot = person.createDataObject("Person");
personRoot.setString("firstName", "John");
personRoot.setString("lastName", "Smith");

person.getChangeSummary().endLogging();

dump("*** CREATE NEW PERSON ...", createPerson);

DataObject createPersonResponse = client.process(callContext, createPerson);

dump("*** PERSON CREATED:", createPersonResponse);

String personRowId = createPersonResponse.getString("object/Person/rowidObject");

DataObject readPerson = dataFactory.create(readPersonType);
DataObject readPersonParameters = readPerson.createDataObject("parameters");
DataObject coFilter = readPersonParameters.createDataObject("coFilter");
DataObject coFilterNode = coFilter.createDataObject("object");
coFilterNode.set("name", "Person");
DataObject key = coFilterNode.createDataObject("key");
key.set("rowid", personRowId);

dump("*** READ CREATED PERSON...", readPerson);

DataObject readPersonResponse = client.process(callContext, readPerson);

dump("*** READ RESULT:", readPersonResponse);

person = readPersonResponse.getDataObject("object");
person.detach();

person.getChangeSummary().beginLogging();

personRoot = person.getDataObject("Person");
// add new 'one' child
DataObject genderCd = personRoot.createDataObject("genderCd");
genderCd.setString("genderCode", "M");

// add two 'many' children
DataObject phonePager = personRoot.createDataObject("TelephoneNumbers");
Property item = phonePager.getInstanceProperty("item");
Type phoneType = item.getType();

DataObject phone1 = dataFactory.create(phoneType);
phone1.setString("phoneNumber", "111-11-11");
DataObject phone2 = dataFactory.create(phoneType);
phone2.setString("phoneNumber", "222-22-22");

phonePager.setList(item, Arrays.asList(phone1, phone2));

person.getChangeSummary().endLogging();

DataObject updatePerson = dataFactory.create(writePersonType);
updatePerson.setDataObject("object", person);
DataObject updatePersonParameters = updatePerson.createDataObject("parameters");
updatePersonParameters.setString("systemName", systemName);
updatePersonParameters.setString("interactionId", "");

dump("*** UPDATE PERSON...", updatePerson);

DataObject updatePersonResponse = client.process(callContext, updatePerson);

dump("*** PERSON UPDATED:", updatePersonResponse);

coFilterNode.set("depth", 3);

readPersonParameters.setBoolean("readSystemFields", true);

```

```

dump("*** READ UPDATED PERSON WITH CHILDREN...", readPerson);

readPersonResponse = client.process(callContext, readPerson);

dump("*** READ RESULT:", readPersonResponse);

person = readPersonResponse.getDataObject("object");
person.detach();

person.getChangeSummary().beginLogging();

genderCd = person.getDataObject("Person").createDataObject("genderCd");
genderCd.setString("genderCode", "F");

// delete one phone
DataObject phoneItem = person.getDataObject("Person/TelephoneNumbers/item[1]");
phoneItem.delete();

person.getChangeSummary().endLogging();

DataObject deletePhone = dataFactory.create(writePersonType);
deletePhone.setDataObject("object", person);
DataObject deletePhoneParameters = deletePhone.createDataObject("parameters");
deletePhoneParameters.setString("systemName", systemName);

dump("*** DELETE CHILD...", deletePhone);

DataObject deletePhoneResponse = client.process(callContext, deletePhone);

dump("*** CHILD DELETED:", deletePhoneResponse);

readPersonParameters.setBoolean("readSystemFields", false);

dump("*** READ PERSON AFTER CHILD WAS DELETED...", readPerson);

readPersonResponse = client.process(callContext, readPerson);

dump("*** READ RESULT:", readPersonResponse);

person = readPersonResponse.getDataObject("object");
person.detach();

person.getChangeSummary().beginLogging();

person.getDataObject("Person").detach();

person.getChangeSummary().endLogging();

DataObject deletePerson = dataFactory.create(writePersonType);
deletePerson.setDataObject("object", person);
DataObject deletePersonParameters = deletePerson.createDataObject("parameters");
deletePersonParameters.setString("systemName", systemName);

dump("*** DELETE PERSON...", deletePerson);

DataObject deletePersonResponse = client.process(callContext, deletePerson);

dump("*** PERSON DELETED:", deletePersonResponse);

dump("*** TRY TO READ PERSON AFTER DELETE", readPerson);

try {
    readPersonResponse = client.process(callContext, readPerson);

    dump("*** READ RESULT:", readPersonResponse);
} catch (CompositeServiceException e) {
    out.println("*** READ RESULT: " + e.getLocalizedMessage());
}

```

```

    }

    private void dump(String title, DataObject dataObject) {
        String xml = helperContext.getXMLHelper().save(
            dataObject,
            dataObject.getType().getURI(),
            dataObject.getType().getName());
        out.println(title);
        out.println(xml);
        out.println();
    }
}

```

Ejemplo de código Java con clases SDO generadas

En el ejemplo se muestra el código Java para ejecutar llamadas Enterprise JavaBean (EJB) basadas en las clases Java que genera MDM Hub en función de la configuración de la entidad de negocio y de los servicios de entidad de negocio.

El ejemplo se encuentra en el siguiente archivo del kit de recursos: C:\<directorio de instalación de MDM Hub>\hub\resourcekit\samples\COS\source\java\com\informatica\mdm\sample\cs\GeneratedSDO.java

El siguiente código Java se basa en las clases generadas y ejecuta llamadas de servicio de entidad de negocio EJB para crear un registro de objeto base de persona, agregar varios registros secundarios, eliminar un registro secundario y, después, eliminar el registro de persona y todos los registros secundarios:

```

package com.informatica.mdm.sample.cs;

import com.informatica.mdm.cs.CallContext;
import com.informatica.mdm.cs.api.CompositeServiceException;
import com.informatica.mdm.cs.client.CompositeServiceClient;
import com.informatica.mdm.sdo.cs.base.CoFilter;
import com.informatica.mdm.sdo.cs.base.CoFilterNode;
import com.informatica.mdm.sdo.cs.base.Key;
import com.siperian.sif.client.EjbSiperianClient;
import com.siperian.sif.client.SiperianClient;
import commonj.sdo.DataObject;
import commonj.sdo.helper.DataFactory;
import commonj.sdo.helper.HelperContext;
import mdm.informatica.co_ors.*;
import mdm.informatica.cs_ors.*;

import java.io.PrintStream;
import java.util.Arrays;
import java.util.Properties;

public class GeneratedSDO {

    public static void main(String[] args) throws CompositeServiceException {

        if(args.length != 3) {
            System.err.println("USAGE: GeneratedSDO <ors> <user> <pass>");
            return;
        }

        new GeneratedSDO(args[0], args[1], args[2]).execute();

    }

    private String orsId;
    private String user;
    private String pass;
    private HelperContext helperContext;

```

```

private PrintStream out = System.out;

public GeneratedSDO(String orsId, String user, String pass) {
    this.orsId = orsId;
    this.user = user;
    this.pass = pass;
}

public void execute() throws CompositeServiceException {

    String systemName = "Admin";

    Properties config = new Properties();
    config.put(SiperianClient.SIPERIANCLIENT_PROTOCOL,
EjbSiperianClient.PROTOCOL_NAME);
    CompositeServiceClient client =
CompositeServiceClient.newCompositeServiceClient(config);

    CallContext callContext = new CallContext(orsId, user, pass);

    helperContext = client.getHelperContext(callContext);

    DataFactory dataFactory = helperContext.getDataFactory();

    // 1. Create new person
    WritePerson createPerson = (WritePerson)dataFactory.create(WritePerson.class);
    WritePersonParameters createPersonParameters =
(WritePersonParameters)dataFactory.create(WritePersonParameters.class);
    createPersonParameters.setSystemName(systemName);
    createPerson.setParameters(createPersonParameters);

    Person person = (Person)dataFactory.create(Person.class);
    createPerson.setObject(person);

    person.getChangeSummary().beginLogging();

    PersonRoot personRoot = (PersonRoot)dataFactory.create(PersonRoot.class);
    personRoot.setFirstName("John");
    personRoot.setLastName("Smith");
    person.setPerson(personRoot);

    person.getChangeSummary().endLogging();

    dump("*** CREATE NEW PERSON ...", createPerson);

    WritePersonReturn createPersonResponse =
(WritePersonReturn)client.process(callContext, (DataObject)createPerson);

    dump("*** PERSON CREATED:", createPersonResponse);

    String personRowId =
createPersonResponse.getObject().getPerson().getRowidObject();

    Key key = (Key)dataFactory.create(Key.class);
    key.setRowid(personRowId);
    CoFilterNode coFilterNode = (CoFilterNode)dataFactory.create(CoFilterNode.class);
    coFilterNode.setName(Person.class.getSimpleName());
    coFilterNode.setKey(key);
    CoFilter coFilter = (CoFilter)dataFactory.create(CoFilter.class);
    coFilter.setObject(coFilterNode);
    ReadPersonParameters readPersonParameters =
(ReadPersonParameters)dataFactory.create(ReadPersonParameters.class);
    readPersonParameters.setCoFilter(coFilter);

    ReadPerson readPerson = (ReadPerson)dataFactory.create(ReadPerson.class);
    readPerson.setParameters(readPersonParameters);

    dump("*** READ CREATED PERSON...", readPerson);

    ReadPersonReturn readPersonResponse =
(ReadPersonReturn)client.process(callContext, (DataObject)readPerson);

```

```

dump("*** READ RESULT:", readPersonResponse);

person = readPersonResponse.getObject();
((DataObject)person).detach();

person.getChangeSummary().beginLogging();

personRoot = person.getPerson();
// add new 'one' child
LUGenderLookup genderCd =
(LUGenderLookup) dataFactory.create(LUGenderLookup.class);
genderCd.setGenderCode("M");
personRoot.setGenderCd(genderCd);

// add two 'many' children
PersonTelephoneNumbersPager phonePager =
(PersonTelephoneNumbersPager) dataFactory.create(PersonTelephoneNumbersPager.class);

PersonTelephoneNumbers phone1 =
(PersonTelephoneNumbers) dataFactory.create(PersonTelephoneNumbers.class);
phone1.setPhoneNumber("111-11-11");
PersonTelephoneNumbers phone2 =
(PersonTelephoneNumbers) dataFactory.create(PersonTelephoneNumbers.class);
phone2.setPhoneNumber("222-22-22");

phonePager.setItem(Arrays.asList(phone1, phone2));
personRoot.setTelephoneNumbers(phonePager);

person.getChangeSummary().endLogging();

WritePerson updatePerson = (WritePerson) dataFactory.create(WritePerson.class);
updatePerson.setObject(person);
WritePersonParameters updatePersonParameters =
(WritePersonParameters) dataFactory.create(WritePersonParameters.class);
updatePersonParameters.setSystemName(systemName);
updatePersonParameters.setInteractionId("");
updatePerson.setParameters(updatePersonParameters);

dump("*** UPDATE PERSON...", updatePerson);

WritePersonReturn updatePersonResponse =
(WritePersonReturn) client.process(callContext, (DataObject)updatePerson);

dump("*** PERSON UPDATED:", updatePersonResponse);

coFilterNode.setDepth(3);

readPersonParameters.setReadSystemFields(true);

dump("*** READ UPDATED PERSON WITH CHILDREN (with system fields)...",
readPerson);

readPersonResponse = (ReadPersonReturn) client.process(callContext,
(DataObject)readPerson);

dump("*** READ RESULT:", readPersonResponse);

person = readPersonResponse.getObject();
((DataObject)person).detach();

person.getChangeSummary().beginLogging();

// delete one phone
person.getPerson().getTelephoneNumbers().getItem().remove(0);

// change gender
genderCd = (LUGenderLookup) dataFactory.create(LUGenderLookup.class);
genderCd.setGenderCode("F");
personRoot.setGenderCd(genderCd);

```



```

        person.getChangeSummary().endLogging();

        WritePerson deletePhone = (WritePerson)dataFactory.create(WritePerson.class);
        deletePhone.setObject(person);
        WritePersonParameters deletePhoneParameters =
        (WritePersonParameters)dataFactory.create(WritePersonParameters.class);
        deletePhoneParameters.setSystemName(systemName);
        deletePhone.setParameters(deletePhoneParameters);

        dump("*** DELETE CHILD...", deletePhone);

        WritePersonReturn deletePhoneResponse =
        (WritePersonReturn)client.process(callContext, (DataObject)deletePhone);

        dump("*** CHILD DELETED:", deletePhoneResponse);

        readPersonParameters.setReadSystemFields(false);

        dump("*** READ PERSON AFTER CHILD WAS DELETED...", readPerson);

        readPersonResponse = (ReadPersonReturn)client.process(callContext,
        (DataObject)readPerson);

        dump("*** READ RESULT:", readPersonResponse);

        person = readPersonResponse.getObject();
        ((DataObject)person).detach();

        person.getChangeSummary().beginLogging();

        ((DataObject)person).getPerson().delete();

        person.getChangeSummary().endLogging();

        WritePerson deletePerson = (WritePerson)dataFactory.create(WritePerson.class);
        deletePerson.setObject(person);
        WritePersonParameters deletePersonParameters =
        (WritePersonParameters)dataFactory.create(WritePersonParameters.class);
        deletePersonParameters.setSystemName(systemName);
        deletePerson.setParameters(deletePersonParameters);

        dump("*** DELETE PERSON...", deletePerson);

        WritePersonReturn deletePersonResponse =
        (WritePersonReturn)client.process(callContext, (DataObject)deletePerson);

        dump("*** PERSON DELETED:", deletePersonResponse);

        dump("*** TRY TO READ PERSON AFTER DELETE", readPerson);

        try {
            readPersonResponse = (ReadPersonReturn)client.process(callContext,
            (DataObject)readPerson);

            dump("*** READ RESULT:", readPersonResponse);
        } catch (CompositeServiceException e) {
            out.println("*** READ RESULT: " + e.getLocalizedMessage());
        }
    }

    private void dump(String title, Object object) {
        DataObject dataObject = (DataObject)object;
        String xml = helperContext.getXMLHelper().save(
            dataObject,
            dataObject.getType().getURI(),
            dataObject.getType().getName());
        out.println(title);
        out.println(xml);
        out.println();
    }
}

```

} }

CAPÍTULO 3

Llamadas del servicio de entidad de negocio de REST

Este capítulo incluye los siguientes temas:

- [Resumen de las API de REST para servicios de entidad de negocio, 27](#)
- [Métodos REST compatibles, 28](#)
- [Método de autenticación, 28](#)
- [Cookies de autenticación para el inicio de sesión desde aplicaciones de terceros, 29](#)
- [Swagger, 32](#)
- [Localizador uniforme de recursos de REST, 32](#)
- [Configuración de encabezado y cuerpo, 33](#)
- [Parámetros de consulta estándar, 35](#)
- [Formatos de fecha y hora UTC, 36](#)
- [Configuración de WebLogic para ejecutar llamadas REST de servicios de entidad de negocio, 37](#)
- [Visualización de parámetros de entrada y salida, 38](#)
- [Plantilla de JavaScript, 38](#)
- [Ejemplo de JavaScript, 39](#)
- [Ejemplo de llamada AJAX con autenticación básica HTTP, 41](#)
- [Referencia de API de REST para servicios de entidad de negocio , 41](#)

Resumen de las API de REST para servicios de entidad de negocio

Con las llamadas de extremo de transferencia de estado representacional (REST), todos los servicios de entidad de negocio estarán disponibles como servicios web.

Puede realizar llamadas de REST para crear, actualizar, eliminar y buscar registros de objeto base y registros secundarios relacionados en una entidad de negocio. Puede realizar operaciones como fusionar, anular la fusión y hacer coincidir registros. Puede realizar llamadas de REST para crear, actualizar y buscar tareas, así como llevarlas a cabo. También puede realizar llamadas de REST para crear, actualizar y eliminar archivos, como adjuntos para tareas o registros. Antes de que utilice las API REST para llamar a los servicios de entidad de negocio, valide el Almacén de referencias operativas.

Una llamada de servicio de entidad de negocio de REST es una solicitud de servicio web con la forma de un localizador uniforme de recursos (URL). MDM Hub asigna una única URL a cada objeto base de una entidad de negocio. Puede utilizar la URL única para identificar el objeto base que desea actualizar o eliminar.

Puede especificar la entidad de negocio o la vista de la entidad de negocio en la URL de la mayoría de las llamadas a la API REST. Por ejemplo, la llamada a la API REST de Create Record puede utilizar los siguientes formatos:

```
http://<host>:<port>/<context>/<databaseID>/<business entity>?systemName=<name of the source system>
```

```
http://<host>:<port>/<context>/<databaseID>/<business entity view name>?systemName=<name of the source system>
```

Algunas API REST no admiten vistas de entidades de negocio. Por ejemplo, las API REST relacionadas con las tareas de fusión, como Vista previa de fusión, Fusión pendiente o Promocionar fusión, no admiten vistas de entidades de negocio.

Nota: Antes de que utilice las API REST para llamar a los servicios de entidad de negocio, valide el Almacén de referencias operativas.

Métodos REST compatibles

Las API de REST para servicios de entidad de negocio utilizan los métodos HTTP estándar para realizar operaciones en los recursos, como son los registros, las tareas y los archivos.

Las API de REST para servicios de entidad de negocio admiten los siguientes métodos de solicitud HTTP:

Método	Descripción
GET	Recupera información sobre un registro, una tarea o un archivo.
POST	Crea una tarea, un registro raíz, un registro secundario o un archivo. Nota: El nombre del Almacén de referencias operativas (ORS) es una solicitud POST con distinción de mayúsculas y minúsculas. Si el nombre del ORS no coincide con el nombre en MDM Hub, se produce un error.
PUT	Actualiza un registro raíz, un registro secundario, una tarea o un archivo.
PATCH	Actualiza una tarea parcialmente.
DELETE	Elimina un registro raíz, un registro secundario o un archivo.

Método de autenticación

Los extremos REST de los servicios de entidad de negocio utilizan el método de autenticación HTTP básico para autenticar usuarios. Cuando se conecte por primera vez a un servicio de entidad de negocio con su navegador, tendrá que proporcionar el nombre de usuario y la contraseña de MDM Hub. Cuando se haya

autenticado correctamente, podrá utilizar las API de REST para servicios de entidad de negocio para realizar operaciones.

El navegador almacena en la caché las credenciales de usuario y las emplea en cada solicitud para acceder a un servicio de entidad de negocio que se produzca con posterioridad.

Cookies de autenticación para el inicio de sesión desde aplicaciones de terceros

Utilice cookies de autenticación para autenticar a los usuarios de MDM Hub y llamar a servicios de entidad de negocio desde aplicaciones de terceros, como Postman. Con el uso de cookies de autenticación, no es necesario codificar el nombre de usuario y la contraseña.

El proceso para utilizar cookies de autenticación incluye los siguientes pasos:

1. Utilice las credenciales de un usuario autenticado para obtener una cookie de autenticación.
2. Guarde la cookie de autenticación.
3. Utilice la cookie de autenticación para llamar a las API REST.

Todas las sesiones de MDM utilizan un token CSRF de Informatica (ICT). Aparece un ICT en el encabezado HTTP.

Cuando un usuario solicita un ID de sesión, la respuesta devuelve un ICT como parte del encabezado HTTP. Para las aplicaciones externas, hay una URL para la autenticación que pueden usar las aplicaciones externas. Esta URL devuelve una cookie de autenticación y un ICT que se encuentra en el encabezado. La cookie de autenticación es un ID de sesión de MDM.

Toda llamada de servicios de entidad de negocio posterior debe incluir la cookie de autenticación y el ICT. Las solicitudes posteriores pueden utilizar el ID de sesión de MDM y el ICT hasta que expire la sesión.

Cuando envía una solicitud para obtener la cookie de autenticación y el ICT, la sección `userinfo` es opcional.

Para administrar sesiones, use llamadas REST.

Obtener la cookie de autenticación y el token CSRF de Informatica

El siguiente ejemplo muestra cómo obtener la cookie de autenticación y el token CSRF de Informatica (ICT):

```
POST: https://<host>:<port>/cmx/auth/<database ID>

Authorization: No Auth

Body:
{
  "username" : "admin",
  "password" : {
    "encrypted" : false,
    "password" : "admin"
  },
  "userInfo" : {
    "sessionTimeout":300,
    "role":"Manager"
  }
}
```

El siguiente ejemplo muestra cómo aparece el ICT en el encabezado:

```
ICT: 0ffa3b95c67e111b9c14c140a70273a15db266993d0c763d555135de4c4d6110
```

El siguiente ejemplo muestra cómo aparece la cookie de autenticación en la respuesta:

```
Name: mdmssessionid
Value: vppOTOhJLKMjYJf7Diil
Domain: 10.xx.xx.xx
Path: /cmx
Expires: Session
HttpOnly: true
Secure: false
```

Crear un registro de persona

El siguiente ejemplo muestra cómo crear un registro de persona con la cookie de autenticación y el token CSRF de Informatica (ICT):

```
POST: https://<host:port>/cmx/cs/<Database ID>/Person?systemName=Admin

Authorization: No Auth (or Inherit auth from the parent)

Headers:
Cookie: mdmssessionid: vppOTOhJLKMjYJf7Diil
ICT: 0ffa3b95c67e111b9c14c140a70273a15db266993d0c763d555135de4c4d6110

Body:
{
  "firstName": "John",
  "lastName": "Smith"
}
```

El ICT de este ejemplo procede de un ejemplo anterior.

El siguiente ejemplo muestra el error que aparece en la respuesta cuando no se pasa la cookie de autenticación y el ICT en la solicitud:

```
{
  "errorCode": "SIP-50201",
  "errorMessage": "SIP-50201: The MDM Hub did not process the business entity service request. The request did not contain credentials. Specify the appropriate credentials in the business entity service request."
}
```

Establecer el tiempo de espera de la sesión

Cuando envía una solicitud para obtener la cookie de autenticación y el token CSRF de Informatica (ICT), la sección `Información de usuario de la sesión` es opcional. Para establecer el tiempo de espera de la sesión, incluya una sección `userinfo` en la solicitud y especifique el parámetro de tiempo de espera de la sesión. El valor del parámetro está en segundos.

El siguiente código de ejemplo muestra la sección de información del usuario y el parámetro de tiempo de espera de la sesión:

```
"userinfo" : {
  "sessionTimeout":300,
  "role":"Manager"
}
```

Obtener información de usuario en la solicitud de inicio de sesión

Puede obtener los detalles del usuario que solicitó la cookie de autenticación y el token CSRF de Informatica (ICT).

La siguiente solicitud de ejemplo muestra cómo obtener información de usuario a partir de la solicitud de inicio de sesión:

```
GET: https://<host:port>/cmx/session/info

Headers:

Cookie: mdmsessionid=zjl4PDN8MEqiQWqAXwxX
ICT: 138d0a12d038323d7a436c7363370011846b97b0bb51006507dd0c435a983ce9
```

La siguiente salida de respuesta de ejemplo muestra información sobre el tiempo de espera de la sesión y el rol:

```
{
  "role": "Manager",
  "sessionTimeout": "300"
}
```

Actualizar una sesión

Puede ampliar la validez de una sesión en 30 minutos con cada solicitud de actualización. Puede especificar el intervalo de tiempo de espera de la sesión en la solicitud de inicio de sesión.

La siguiente solicitud de ejemplo muestra cómo extender o actualizar una sesión:

```
GET: https://<host:port>/cmx/session/refresh

Headers:

Cookie: mdmsessionid=zjl4PDN8MEqiQWqAXwxX
```

Validar una sesión

Puede validar si una sesión está activa y es válida.

La siguiente solicitud de ejemplo muestra cómo validar una sesión:

```
GET: https://<host:port>/cmx/session/validate

Headers:

Cookie: mdmsessionid=zjl4PDN8MEqiQWqAXwxX
```

El estado HTTP de OK indica que la sesión está activa y es válida.

Cerrar sesión en una sesión

Después de cerrar la sesión, la cookie de autenticación y el token CSRF de Informatica (ICT) relacionados con la sesión caducan. Si los usa en una solicitud, aparece un error en la respuesta.

La siguiente solicitud de ejemplo muestra cómo cerrar una sesión:

```
GET: https://<host:port>/cmx/session/logout

Headers:

Cookie: mdmsessionid=zjl4PDN8MEqiQWqAXwxX
ICT: 138d0a12d038323d7a436c7363370011846b97b0bb51006507dd0c435a983ce9
```

El estado HTTP de OK indica que el cierre de sesión fue exitoso.

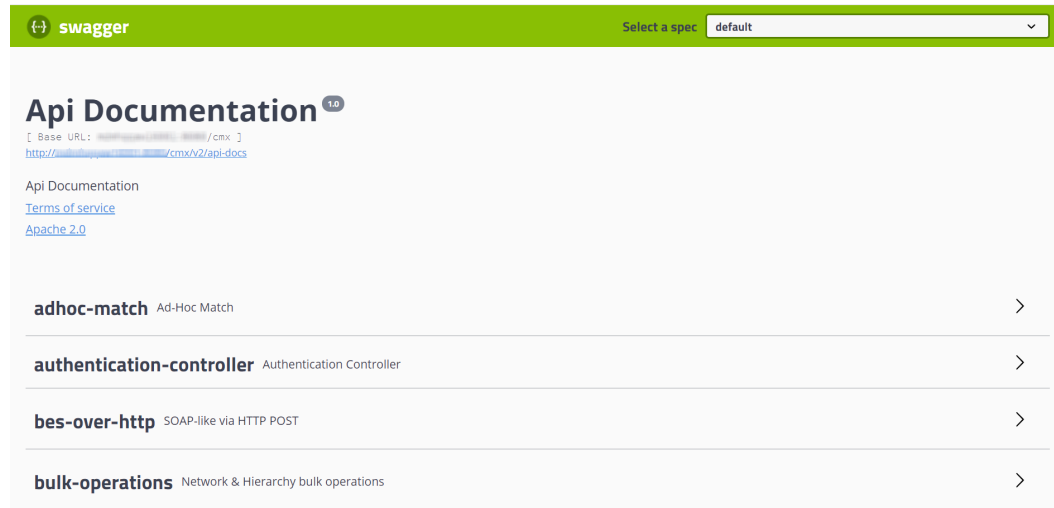
Swagger

MDM Hub usa Swagger para enumerar todos los servicios web de REST con URL y parámetros.

Puede acceder a Swagger con la siguiente URL:

```
http://<host>:<puerto>/cmx/swagger-ui.html
```

En la siguiente imagen, se muestra la interfaz de usuario de Swagger:



Localizador uniforme de recursos de REST

Utilice una URL de REST para realizar llamadas REST para servicios de entidad de negocio.

La URL de REST tiene la siguiente sintaxis:

```
http://<host>:<puerto>/<contexto>/<ID de base de datos>/<ruta de acceso>
```

La URL tiene los siguientes campos:

host

El host que está ejecutando la base de datos.

puerto

Número de puerto que utiliza el servicio de escucha de la base de datos.

contexto

El contexto para las API de entidad de negocio, búsqueda, consulta, coincidencia, tarea y jerarquía es `cmx/cs`.

El contexto para la API de columnas de coincidencia es `cmx`.

El contexto para las API de archivo es `cmx/file`.

El contexto para las API de tarea es `cmx/task`.

El contexto para las API de administración de tareas masiva es `cmx/task/operations`.

El contexto para las API de relación masiva es `cmx/bulk`.

El contexto para las API de informe de gráfico es `cmx/report`.

Nota: En un entorno de MDM alojado, incluya el nombre de inquilino en el contexto. Por ejemplo, el contexto puede ser <nombre de inquilino>/cmx/cs o <nombre de inquilino>/cmx/file.

ID de base de datos

El ID de ORS como está registrado en la herramienta de bases de datos de la Consola del concentrador.

ruta de acceso

El objeto sobre el que quiere usar la API, como registros, tareas o archivos.

Si la dirección URL corresponde a un registro raíz, la ruta de acceso será el nombre del objeto raíz seguido de un identificador único.

Un ejemplo de una ruta de acceso para un registro raíz de persona puede ser `Person/798243.json`.

Si la dirección URL corresponde a un registro que sea un elemento secundario directo del objeto raíz, la ruta también incluiría el nombre del registro secundario y un identificador único.

Un ejemplo de una ruta para un registro de dirección de facturación que sea un elemento secundario de un registro raíz de persona sería:

```
Person/798243/BillAddresses/121522.json.
```

Si la dirección URL es para un registro secundario que esté a un nivel de dos o más, la ruta también incluiría la profundidad.

La siguiente URL es un ejemplo de una URL de REST para un registro secundario con una profundidad de 2:

```
http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/Person/798243/BillAddresses/121522.json?depth=2
```

Nota: Los parámetros distinguen entre mayúsculas y minúsculas. Asegúrese de que el caso de los nombres de parámetro de la URL de REST coincida con el caso de los nombres de parámetro de la configuración de REST.

Configuración de encabezado y cuerpo

Una operación REST combina un método HTTP con la dirección URL completa al recurso. Para obtener una solicitud completa, combine la operación REST con los encabezados HTTP correspondientes y otros datos necesarios. Una solicitud REST posee un componente de encabezado y de cuerpo. Para definir una solicitud se puede utilizar el formato JSON o XML.

Encabezado de solicitud

Utilice un encabezado de solicitud para definir los parámetros operativos o los metadatos de la operación REST. El encabezado consta de una serie de pares campo-valor. La línea de solicitud de la API contiene el método y la dirección URL. Especifique los campos del encabezado después de la línea de la solicitud.

Para crear el encabezado de solicitud de la API de REST, agregue los campos de encabezado tras la línea de solicitud <METHOD> <<host>:<port>/<context>/<database ID>/<Path>, como se muestra en el siguiente ejemplo:

```
<METHOD> <<host>:<port>/<context>/<database ID>/<Path>
Content-Type: application/<json/xml>
Accept: application/<json/xml>
```

En la siguiente tabla se describen algunos de los campos de encabezado de solicitud más frecuentes:

Componente de solicitud	Descripción
Content-Type	El tipo de medio de los datos en la solicitud. Al incluir un cuerpo en la solicitud de REST, debe especificar el tipo de medio del cuerpo en el campo de encabezado Content-Type. Incluya el campo de encabezado Content-Type en las solicitudes PUT y POST.
Accept	El tipo de medio de los datos de la respuesta. Para especificar el formato de la solicitud, utilice <code>application/<json/xml></code> en el encabezado o agregue <code>.json</code> o <code>.xml</code> a la URL. El valor predeterminado es XML.

Cuerpo de la solicitud

Utilice el cuerpo de solicitud de la API de REST para enviar datos en la solicitud. El cuerpo de la solicitud se utiliza con un método en el que se puede anexar un cuerpo, como los métodos POST y PUT. El cuerpo de los datos se escribe después de las líneas de encabezado. Si el mensaje de la solicitud incluye un cuerpo, utilice el campo de encabezado Content-Type para especificar el formato del cuerpo en el encabezado de la solicitud.

Los archivos de definición de esquema XML (XSD) describen qué elementos y atributos se pueden utilizar. El contenido del cuerpo de la solicitud depende de los tipos de elementos que se definan en los archivos XSD.

Los archivos XSD se encuentran en la siguiente ubicación:

```
http://<host>:<puerto>/cmx/csfiles
```

Formato XML

Cuando utilice el formato de solicitud XML, defina el objeto de la solicitud como un conjunto delimitador de etiquetas.

Utilice el siguiente formato XML para definir un objeto de solicitud:

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

En el siguiente ejemplo se muestra la representación XML de un objeto de solicitud:

```
<task>
  <taskType>
    <name>UpdateWithApprovalWorkflow</name>
  </taskType>
  <taskId>urn:b4p2:5149</taskId>
  <owner>manager</owner>
  <title>Smoke test task 222</title>
  <comments>Smoke testing</comments>
  <dueDate>2015-06-15T00:00:00</dueDate>
  <status>OPEN</status>
  <priority>NORMAL</priority>
  <creator>admin</creator>
  <createDate>2015-06-15T00:00:00</createDate>
  <updatedBy>admin</updatedBy>
  <lastUpdateDate>2015-06-15T00:00:00</lastUpdateDate>
  <businessEntity>Person</businessEntity>
  <orsId>localhost-orcl-DS_UI1</orsId>
  <processId>IDDUpdateWithApprovalTask</processId>
  <taskRecord>
    <businessEntity>
      <name>Person</name>
    </businessEntity>
  </taskRecord>
</task>
```

```

        <key>
            <rowid>123</rowid>
            <systemName></systemName>
            <sourceKey></sourceKey>
        </key>
    </businessEntity>
</taskRecord>
</task>

```

Formato JSON

Cuando utilice el formato de solicitud JSON, defina el objeto de solicitud con el atributo de tipo.

Utilice el siguiente formato JSON para especificar un objeto de solicitud:

```

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

```

En el siguiente ejemplo se muestra la representación de JSON de un objeto de solicitud:

```

{
  "type"="task"
  taskType: {name:"UpdateWithApprovalWorkflow"},
  taskId: "urn:b4p2:5149",
  owner: "manager",
  title: "Smoke test task 222",
  comments: "Smoke testing",
  dueDate: "2015-06-15T00:00:00",
  status: "OPEN",
  priority: "NORMAL",
  creator: "admin",
  createDate: "2015-06-15T00:00:00",
  updatedBy: "admin",
  lastUpdateDate: "2015-06-15T00:00:00",
  businessEntity: "Person",
  orsId: "localhost-orcl-DS_UI1",
  processId: 'IDDUpdateWithApprovalTask',
  taskRecord: [{
    businessEntity:{
      name: 'Person',
      key:{
        rowid: '123',
        systemName: '',
        sourceKey: ''
      }
    }
  ]
}

```

Parámetros de consulta estándar

Las API de REST de servicios de entidad de negocio utilizan parámetros de consulta estándar para filtrar, paginar y ampliar los resultados.

Utilice un signo de interrogación (?) para separar los parámetros de consulta del resto de parámetros. Los parámetros de consulta son pares clave-valor separados por el signo igual. Utilice un ampersand (&) para separar una secuencia de parámetros de consulta.

La siguiente URL de solicitud REST muestra cómo se deben usar los parámetros de consulta:

`/Person/123/Phone/SFA:456/PhoneUse?recordsToReturn=100&recordStates=ACTIVE,PENDING`

Utilice los siguientes parámetros de consulta estándar:

Parámetro	Descripción
recordsToReturn	Especifica el número de filas que se van a devolver. El valor predeterminado es 10.
firstRecord	Especifica la primera fila del resultado. El valor predeterminado es 1. Se utiliza en las llamadas posteriores para leer más páginas.
searchToken	Especifica el token de búsqueda que se devuelve con la solicitud anterior. Puede utilizar el token de búsqueda para obtener las páginas posteriores de los resultados de búsqueda. Por ejemplo, la siguiente consulta enumera la primera página: <code>/Person/123/Phone</code> La siguiente consulta devuelve la segunda página: <code>/Person/123/Phone?searchToken=SVR1.AZAM5&firstRecord=10</code>
returnTotal	Si se establece en true, devuelve el número de registros en el resultado. El valor predeterminado es false.
depth	Especifica el número de niveles de elementos secundarios que se van a incluir en el resultado.

Formatos de fecha y hora UTC

En la solicitud y la respuesta, todas las fechas y horas se especifican en formato UTC (hora universal coordinada) con o sin ajuste para una zona horaria específica.

Cuando especifique una fecha y una hora en el cuerpo de una solicitud, use uno de los formatos definidos en [Date and Time Formats \(NOTE-datetime\)](#) de acuerdo con la norma ISO 8601.

Las siguientes directrices se han extraído del documento NOTE-datetime:

Tipo	Sintaxis	Ejemplo
Fecha: año	YYYY	1997
Fecha: año y mes	YYYY-MM	1997-07
Fecha: año, mes y día	YYYY-MM-DD	1997-07-16
Fecha más horas y minutos	YYYY-MM-DDThh:mmTZD	1997-07-16T19:20+01:00
Fecha más horas, minutos y segundos	YYYY-MM-DDThh:mm:ssTZD	1997-07-16T19:20:30+01:00
Fecha más horas, minutos, segundos y segundos fraccionarios	YYYY-MM-DDThh:mm:ss.sTZD	1997-07-16T19:20:30.45+01:00

Donde:

- YYYY = un año de cuatro dígitos
- MM = un mes de dos dígitos, de 01 a 12
- DD = un día del mes de dos dígitos, de 01 a 31
- T = un valor literal que sigue a la fecha y presenta la hora
- hh = dos dígitos para la hora, de 00 a 23
- mm = dos dígitos para los minutos, de 00 a 59
- ss = dos dígitos para los segundos, de 00 a 59
- s = uno o varios dígitos que representan una fracción decimal de un segundo
- TZD = designador de zona horaria (Z o +hh:mm o -hh:mm)
 - Z para hora UTC
 - +hh:mm para una zona horaria local anterior a UTC
 - -hh:mm para una zona horaria local posterior a UTC

Configuración de WebLogic para ejecutar llamadas REST de servicios de entidad de negocio

Dado que las llamadas REST de servicios de entidad de negocio utilizan autenticación HTTP básica, debe deshabilitar la autenticación del Servidor WebLogic para las llamadas REST. Para configurar WebLogic de manera que ejecute las llamadas REST de servicios de entidad de negocio, edite el archivo `config.xml` de WebLogic.

1. Desplácese hasta el siguiente directorio de WebLogic:

En UNIX.

```
<directorio de instalación de WebLogic>/user_projects/domains/base_domain/config
```

En Windows.

```
<directorio de instalación de WebLogic>\user_projects\domains\base_domain\config
```

2. Abra el siguiente archivo en un editor de texto:

```
config.xml
```

3. Antes de la etiqueta `</security-configuration>` de cierre, agregue el siguiente código XML:

```
<enforce-valid-basic-auth-credentials>  
  false  
</enforce-valid-basic-auth-credentials>
```

Visualización de parámetros de entrada y salida

Puede usar Swagger para ver los parámetros de entrada y salida de las API de REST.

En la siguiente imagen, se muestran los parámetros de entrada en Swagger para la API de REST de coincidencia ad hoc:

Parameters

Name	Description
jobGroupControlId string (path)	Batch Job Group ID <input type="text" value="jobGroupControlId - Batch Job Group ID"/>
orsId string (path)	ORS ID <input type="text" value="orsId - ORS ID"/>
withMatches boolean (query)	Return file records that matched to existing data <input type="text" value="true"/>
withoutMatches boolean (query)	Return file records that don't have matches in existing data <input type="text" value="false"/>

Plantilla de JavaScript

En el siguiente ejemplo de código se muestra una plantilla básica que se puede modificar para crear código JavaScript para las llamadas de servicio de entidad de negocio REST. Necesita la biblioteca de scripts java jQuery.

```
(function ($) {
    window.CSClient = window.CSClient || {
        baseUrl: "/cmx/cs/" + "[siperian-client.orsId]",
        user: "[siperian-client.username]",
        pass: "[siperian-client.password]",

        process: function (method, url, body, params) {
            var fullUrl = this.baseUrl + url + ".json?" + $.param(params);
            return $.ajax({
                method: method,
                contentType: "application/json",
                url: fullUrl,
                data: JSON.stringify(body),
                beforeSend: function (xhr) {
                    xhr.setRequestHeader("Authorization", "Basic " + btoa(CSClient.user
+ ":" + CSClient.pass));
                }
            });
        },

        readCo: function (url, params) {
            return this.process("GET", url, null, params);
        },
        createCo: function (url, body, params) {
            return this.process("POST", url, body, params);
        },
        updateCo: function (url, body, params) {
            return this.process("PUT", url, body, params);
        },
    },
},
```

```

        deleteCo: function (url, params) {
            return this.process("DELETE", url, null, params);
        }
    };
})(jQuery);

```

Ejemplo de JavaScript

El kit de recursos contiene ejemplos del código fuente de Java que muestran cómo realizar llamadas de servicio de entidad de negocio REST.

El código de ejemplo se muestra en el siguiente archivo:

```

<directorio de instalación de MDM Hub>\hub\resourcekit\samples\COS\source\resources\webapp
\rest-api.html

```

El siguiente código muestra las llamadas de las API de REST para crear un registro de raíz de persona, agregar varios registros secundarios, eliminar un registro secundario y, a continuación, eliminar el registro de persona y todos los registros secundarios:

```

<html>
<head>
    <script type="text/javascript" src="jquery-1.11.1.js"></script>
    <script type="text/javascript" src="cs-client.js"></script>
</head>
<body>

<script type="text/javascript" language="javascript">
    $(document).ready(function () {

        $("#run").click(function () {

            log = function(msg, json) {
                $('#log').before("<hr/><b>" + msg + "</b>");
                $('#log').before("<pre>" + JSON.stringify(json, undefined, 2) + "</
pre>");
            };

            CSClient.createCo(
                "/Person",
                {
                    firstName: "John",
                    lastName: "Smith"
                },
                {
                    systemName: "Admin"
                }
            ).then(
                function (result) {
                    log("PERSON CREATED:", result);
                    return CSClient.readCo(
                        "/Person/" + result.Person.rowidObject.trim(),
                        {
                            depth: 1
                        }
                    );
                }
            ).then(
                function (result) {
                    log("READ CREATED PERSON:", result);
                    return CSClient.updateCo(
                        "/Person/" + result.rowidObject.trim(),
                        {
                            genderCd: {

```

```

        genderCode: "M"
    },
    TelephoneNumbers: {
        item: [
            {
                phoneNumber: "111-11-11"
            },
            {
                phoneNumber: "222-22-22"
            }
        ]
    }
},
{
    systemName: "Admin"
}
);
}
).then(
    function (result) {
        log("PERSON UPDATED:", result);
        return CSClient.readCo(
            "/Person/" + result.Person.rowidObject.trim(),
            {
                depth: 3,
                readSystemFields: true
            }
        );
    }
).then(
    function (result) {
        log("READ UPDATED PERSON:", result);
        return CSClient.deleteCo(
            "/Person/" + result.rowidObject.trim() + "/"
            TelephoneNumbers/" + result.TelephoneNumbers.item[0].rowidObject.trim(),
            {
                systemName: "Admin"
            }
        );
    }
).then(
    function (result) {
        log("TELEPHONE DELETED:", result);
        return CSClient.readCo(
            "/Person/" + result.Person.rowidObject.trim(),
            {
                depth: 3
            }
        );
    }
).then(
    function (result) {
        log("READ PERSON AFTER TELEPHONE IS DELETED:", result);
        return CSClient.deleteCo(
            "/Person/" + result.rowidObject.trim(),
            {
                systemName: "Admin"
            }
        );
    }
).then(
    function (result) {
        log("PERSON DELETED:", result);
        return CSClient.readCo(
            "/Person/" + result.Person.rowidObject.trim(),
            {
                depth: 1,
                recordStates: "ACTIVE,PENDING,DELETED",
                readSystemFields: true
            }
        );
    }
);

```



```

        }
    ).then(
        function (result) {
            log("READ PERSON AFTER DELETE (HSI -1):", result);
        }
    );
});

});
</script>

<input type="button" id="run" value="Run..." />
<p/>
<div id="log"></div>
</body>
</html>

```

Ejemplo de llamada AJAX con autenticación básica HTTP

Una llamada a los servicios de entidad de negocio requiere autenticación básica HTTP.

El siguiente código muestra un ejemplo de una llamada AJAX con autenticación básica HTTP que usa jQuery:

```

<script type="text/javascript">
    function readMeta(user, pass){
        $.ajax({
            type: "GET",
            url: "/cmx/cs/localhost-mdm-DS_UI1/Person?action=meta",
            dataType: 'json',
            async: false,
            headers: {
                "Authorization": "Basic " + btoa(user + ":" + pass)
            },
            success: function (data) {
                alert('Data received! ' + data.object.label);
            }
        });
    }
</script>

```

En este ejemplo, `btoa` es una función estándar que codifica una cadena en formato Base64.

El desarrollador que implementa esta llamada puede decidir cómo pasar el nombre de usuario y la contraseña.

Nota: Para que las llamadas AJAX tengan éxito, aloje la aplicación JavaScript en el mismo dominio que MDM Hub.

Referencia de API de REST para servicios de entidad de negocio

La referencia de la API de REST para servicios de entidad de negocio enumera las API de REST y proporciona una descripción de cada una de ellas. La referencia de API también contiene información sobre las direcciones URL, los parámetros de consulta, las solicitudes de muestra y las respuestas de ejemplo.

Obtener metadatos

La API de REST de obtención de metadatos devuelve la estructura de datos de una entidad de negocio o una relación de entidad de negocio.

La API utiliza el método GET para devolver los siguientes metadatos de una entidad de negocio:

- Estructura de la entidad de negocio
- Lista de campos
- Tipos de campos, como el tipo y el tamaño de los datos
- Configuración de seguridad para operaciones, como crear, actualizar y fusionar
- Etiquetas localizadas para nodos o campos
- Nombre del código y campos de visualización para campos de búsqueda
- Valores predeterminados para campos y campos de búsqueda

Nota: Varios valores predeterminados no son compatibles con los campos de búsqueda dependientes.

La API devuelve los siguientes detalles de una relación de entidad de negocio:

- Nombre y etiqueta de la relación.
- Entidades de negocio de origen y destino.
- Dirección de la relación.

URL de solicitud

La dirección URL de obtención de metadatos tiene el siguiente formato para una entidad de negocio:

```
http://<host>:<port>/<context>/<database ID>/<business entity>?action=meta
```

La dirección URL de obtención de metadatos tiene el siguiente formato para una relación:

```
http://<host>:<port>/<context>/<database ID>/<relationship>?action=meta
```

Utilice el parámetro de consulta "action=meta" para recuperar la información de metadatos. Asegúrese de especificar correctamente el nombre de la entidad de negocio.

Haga una solicitud GET de HTTP a la URL para la obtención de metadatos:

```
GET http://<host>:<port>/<context>/<database ID>/<business entity/relationship>?  
action=meta
```

Puede agregar encabezados HTTP a la solicitud.

Ejemplo de solicitud de API

La siguiente solicitud de ejemplo recupera la información de metadatos para la entidad de negocio de persona y el nodo raíz:

```
GET http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/Person?action=meta
```

La siguiente solicitud de ejemplo incluye un encabezado para recuperar la información de metadatos para la entidad de negocio de persona en el formato JSON:

```
GET http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/Person?action=meta  
Accept: application/json
```

En la siguiente solicitud de ejemplo se recupera la información de los metadatos para la relación HouseholdContainsMemberPerson:

```
GET http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/  
ProductGroupProductGroupIsParentOfProductGroupProductGroup?action=meta
```

Ejemplo de respuesta de API

Las respuestas de ejemplo para una entidad y una relación contienen permisos de seguridad. La primera sección `operations` define los posibles permisos. La sección `object` enumera los permisos para toda la entidad de negocio o relación. La sección `fields` define los permisos en el nivel de campo.

El siguiente ejemplo muestra una estructura de datos parcial para la entidad de negocio de persona en el formato JSON.

```
{
  "operations": {
    "read": {
      "allowed": true
    },
    "search": {
      "allowed": true
    },
    "create": {
      "allowed": true,
      "task": {
        "template": {
          "title": "Review changes in {taskRecord[0].label}",
          "priority": "NORMAL",
          "dueDate": "2018-04-24T09:28:13.455-04:00",
          "taskType": "AVOSBeUpdate",
          "comment": "This is urgent. Please review ASAP"
        },
        "comment": "AS_REQUIRED",
        "attachment": "OPTIONAL"
      }
    },
    "update": {
      "allowed": true,
      "task": {
        "template": {
          "title": "Review changes in {taskRecord[0].label}",
          "priority": "NORMAL",
          "dueDate": "2018-04-24T09:28:13.455-04:00",
          "taskType": "AVOSBeUpdate",
          "comment": "This is urgent. Please review ASAP"
        },
        "comment": "AS_REQUIRED",
        "attachment": "OPTIONAL"
      }
    },
    "merge": {
      "allowed": true,
      "task": {
        "template": {
          "title": "Review changes in {taskRecord[0].label}",
          "priority": "NORMAL",
          "dueDate": "2018-04-24T09:28:13.455-04:00",
          "taskType": "AVOSBeMerge",
          "comment": "This is urgent. Please review ASAP"
        },
        "comment": "AS_REQUIRED",
        "attachment": "OPTIONAL"
      }
    },
    "delete": {
      "allowed": true
    },
    "unmerge": {
      "allowed": true,
      "task": {
        "template": {
          "title": "Review changes in {taskRecord[0].label}",
          "priority": "NORMAL",
          "dueDate": "2018-04-24T09:28:13.455-04:00",
          "taskType": "AVOSBeUnmerge",

```

```

        "comment": "This is urgent. Please review ASAP"
    },
    "comment": "AS REQUIRED",
    "attachment": "OPTIONAL"
}
}
},
"objectType": "ENTITY",
"timeline": true,
"object": {
    "operations": {
        "read": {
            "allowed": true
        },
        "create": {
            "allowed": true
        },
        "update": {
            "allowed": true
        },
        "merge": {
            "allowed": true
        },
        "delete": {
            "allowed": true
        },
        "unmerge": {
            "allowed": true
        }
    }
},
"field": [
    {
        "operations": {
            "read": {
                "allowed": true
            },
            "create": {
                "allowed": true
            },
            "update": {
                "allowed": true
            }
        },
        "allowedValues": [
            "Person"
        ],
        "searchable": {
            "filterable": true,
            "facet": true
        },
        "name": "partyType",
        "label": "Party Type",
        "dataType": "String",
        "length": 255
    },
    {
        "operations": {
            "read": {
                "allowed": true
            },
            "create": {
                "allowed": true
            },
            "update": {
                "allowed": true
            }
        },
        "name": "lastName",
        "label": "Last Name",
        "dataType": "String",
        "length": 50
    }
]
}

```

```

    },
    {
      "operations": {
        "read": {
          "allowed": true
        },
        "create": {
          "allowed": true
        },
        "update": {
          "allowed": true
        }
      },
      "searchable": {
        "filterable": true,
        "facet": true
      },
      "name": "displayName",
      "label": "Display Name",
      "dataType": "String",
      "length": 200
    },
    ...
  ],
  "name": "Person",
  "label": "Person",
  "many": false
}
}

```

Lista de metadatos

La API de REST de lista de metadatos devuelve una lista de las entidades de negocio o las relaciones que ha definido. La respuesta incluye información de la línea temporal e información de seguridad, si las entidades de negocio contienen esa información. Puede usar la API para recuperar la lista de relaciones que empiezan en una entidad, finalizan en una entidad o empiezan y finalizan en las entidades especificadas.

La API usa el método GET.

Dirección URL de lista de metadatos

La dirección URL de lista de metadatos tiene el siguiente formato para los metadatos de entidad:

```
http://<host>:<port>/<context>/<database ID>/meta/entity
```

La dirección URL de lista de metadatos tiene el siguiente formato para los metadatos de relación:

```
http://<host>:<port>/<context>/<database ID>/meta/relationship
```

Use la siguiente solicitud HTTP GET para la dirección URL de lista de metadatos:

```
GET http://<host>:<port>/<context>/<database ID>/meta/entity|relationship
```

Parámetros de consulta

Puede anexas los parámetros de consulta a la dirección URL de la solicitud para filtrar los resultados de búsqueda para las relaciones de entidad de negocio. Puede especificar la dirección y la búsqueda de relaciones en y desde una entidad de negocio.

En la tabla siguiente se muestran los parámetros de la consulta:

Parámetro	Descripción
start	Opcional. Especifica la entidad en la que se origina una relación. Por ejemplo, la consulta <code>meta/relationship?start=Organization</code> devuelve todas las relaciones que empiezan a partir de la entidad de negocio <code>Organization</code> .
finish	Opcional. Especifica la entidad en la que finaliza una relación. Por ejemplo, la consulta <code>meta/relationship?finish=Person</code> devuelve todas las relaciones que finalizan en la entidad de negocio <code>Person</code> .

Puede especificar ambos parámetros para recuperar todas las relaciones entre las dos entidades de negocio. Por ejemplo, la consulta `meta/relationship?start=Organization&finish=Person` devuelve todas las relaciones que empiezan por la entidad de negocio `Organization` y terminan por la entidad de negocio `Person`.

Ejemplo de solicitud de API

En la siguiente solicitud de ejemplo se recupera la lista de entidades de negocio configuradas:

```
GET http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/meta/entity
```

En la siguiente solicitud de ejemplo se recupera la lista de relaciones configuradas:

```
GET http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/meta/relationship
```

En la siguiente solicitud de ejemplo se recupera la lista de relaciones que se inician en la entidad de negocio `Organization` y finalizan en la entidad de negocio `Person`:

```
GET http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/meta/relationship?start=Organization&finish=Person
```

Ejemplo de respuesta de API

El siguiente ejemplo muestra un fragmento de la lista de relaciones configuradas:

```
{
  "link": [],
  "firstRecord": 1,
  "pageSize": 10,
  "item": [
    {
      "operations": {
        "read": {
          "allowed": true
        },
        "search": {
          "allowed": false
        },
        "create": {
          "allowed": true,
          "task": {
            "template": {
              "title": "Review changes in {taskRecord[0].label}",
            }
          }
        }
      }
    }
  ]
}
```

```

        "priority": "NORMAL",
        "dueDate": "2018-04-24T09:31:48.167-04:00",
        "taskType": "AVOSBeUpdate",
        "comment": "This is urgent. Please review ASAP"
    },
    "comment": "AS_REQUIRED",
    "attachment": "OPTIONAL"
}
},
"update": {
    "allowed": true,
    "task": {
        "template": {
            "title": "Review changes in {taskRecord[0].label}",
            "priority": "NORMAL",
            "dueDate": "2018-04-24T09:31:48.167-04:00",
            "taskType": "AVOSBeUpdate",
            "comment": "This is urgent. Please review ASAP"
        },
        "comment": "AS_REQUIRED",
        "attachment": "OPTIONAL"
    }
},
"merge": {
    "allowed": true,
    "task": {
        "template": {
            "title": "Review changes in {taskRecord[0].label}",
            "priority": "NORMAL",
            "dueDate": "2018-04-24T09:31:48.167-04:00",
            "taskType": "AVOSBeMerge",
            "comment": "This is urgent. Please review ASAP"
        },
        "comment": "AS_REQUIRED",
        "attachment": "OPTIONAL"
    }
},
"delete": {
    "allowed": true
},
"unmerge": {
    "allowed": true,
    "task": {
        "template": {
            "title": "Review changes in {taskRecord[0].label}",
            "priority": "NORMAL",
            "dueDate": "2018-04-24T09:31:48.167-04:00",
            "taskType": "AVOSBeUnmerge",
            "comment": "This is urgent. Please review ASAP"
        },
        "comment": "AS_REQUIRED",
        "attachment": "OPTIONAL"
    }
}
},
"objectType": "ENTITY",
"timeline": false,
"object": {
    "link": [
        {
            "href": "http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/
                CreditCard.json?action=meta",
            "rel": "entity"
        }
    ],
    "field": [
        {
            "name": "issuingCompany",
            "label": "Issuing Company",
            "dataType": "String",
            "length": 100
        }
    ]
}

```

```

},
{
  "name": "expirationYear",
  "label": "Expiration Year",
  "dataType": "String",
  "length": 4
},
{
  "allowedValues": [
    "Credit Card"
  ],
  "name": "accountType",
  "label": "Account Type",
  "dataType": "String",
  "length": 255
},
{
  "name": "accountNumber",
  "label": "Account Number",
  "dataType": "String",
  "length": 20
},
{
  "name": "securityCode",
  "label": "Security Code",
  "dataType": "String",
  "length": 4
},
{
  "name": "expirationMonth",
  "label": "Expiration Month",
  "dataType": "String",
  "length": 2
},
{
  "name": "cardholderName",
  "label": "Card Holder Name",
  "dataType": "String",
  "length": 100
},
{
  "name": "consolidationInd",
  "label": "Consolidation Ind",
  "dataType": "Integer",
  "length": 38,
  "readOnly": true,
  "system": true
},
{
  "name": "creator",
  "label": "Creator",
  "dataType": "String",
  "length": 50,
  "readOnly": true,
  "system": true
},
{
  "name": "interactionId",
  "label": "Interaction Id",
  "dataType": "Integer",
  "length": 38,
  "readOnly": true,
  "system": true
},
{
  "name": "updatedBy",
  "label": "Updated By",
  "dataType": "String",
  "length": 50,
  "readOnly": true,
  "system": true
}

```



```

},
{
  "name": "lastUpdateDate",
  "label": "Last Update Date",
  "dataType": "Date",
  "readOnly": true,
  "system": true
},
{
  "name": "lastRowidSystem",
  "label": "Last Rowid System",
  "dataType": "String",
  "length": 14,
  "readOnly": true,
  "system": true
},
{
  "name": "dirtyIndicator",
  "label": "Dirty Indicator",
  "dataType": "Integer",
  "length": 38,
  "readOnly": true,
  "system": true
},
{
  "name": "deletedBy",
  "label": "Deleted By",
  "dataType": "String",
  "length": 50,
  "readOnly": true,
  "system": true
},
{
  "name": "deletedInd",
  "label": "Deleted Indicator",
  "dataType": "Integer",
  "length": 38,
  "readOnly": true,
  "system": true
},
{
  "name": "hubStateInd",
  "label": "Hub State Ind",
  "dataType": "Integer",
  "length": 38,
  "readOnly": true,
  "system": true
},
{
  "name": "deletedDate",
  "label": "Deleted Date",
  "dataType": "Date",
  "readOnly": true,
  "system": true
},
{
  "name": "rowidObject",
  "label": "Rowid Object",
  "dataType": "String",
  "length": 14,
  "readOnly": true,
  "system": true
},
{
  "name": "cmDirtyInd",
  "label": "Content metadata dirty Ind",
  "dataType": "Integer",
  "length": 38,
  "readOnly": true,
  "system": true
},
},

```

```

        {
            "name": "createDate",
            "label": "Create Date",
            "dataType": "Date",
            "readOnly": true,
            "system": true
        }
    ],
    "name": "CreditCard",
    "label": "Credit Card",
    "many": false
}
...
]
}

```

Enumerar columnas de coincidencia

La API de REST de enumeración de columnas de coincidencia puede devolver una lista de conjuntos de reglas de coincidencia para una entidad de negocio especificada o una lista de columnas de coincidencia para un conjunto de reglas de coincidencia especificado. Puede generar una lista de las columnas de coincidencia y usar estas columnas con la API de REST SearchMatch.

Si desea información sobre cómo configurar columnas de coincidencia y conjuntos de reglas de coincidencia, consulte la *Guía de configuración de Multidomain MDM*.

La API usa el método GET.

URL de solicitud

El contexto de la URL para enumerar columnas de coincidencia es `cmx`. En un entorno de MDM alojado, incluya el nombre de inquilino en el contexto, como `<nombre de inquilino>/cmx`.

La URL para enumerar columnas de coincidencia tiene los siguientes formatos:

URL para devolver todos los conjuntos de reglas de coincidencia

Utilice la siguiente URL para enumerar todos los conjuntos de reglas de coincidencia para una entidad de negocio específica:

```
http://<host>:<port>/<context>/queryTemplate/<database ID>/<business entity>
```

Haga la siguiente solicitud HTTP GET a la URL para enumerar columnas de coincidencia:

```
GET http://<host>:<port>/<context>/queryTemplate/<database ID>/<business entity>
```

URL para devolver las columnas de coincidencia que se utilizan en un conjunto de reglas de coincidencia especificado

Utilice la siguiente URL para enumerar todas las columnas de coincidencia en un conjunto de reglas de coincidencia especificado:

```
http://<host>:<port>/<context>/queryTemplate/<database ID>/<business entity>/<match rule set>
```

Ejemplo de solicitud de API

Solicitud para los conjuntos de reglas de coincidencia

Con la siguiente solicitud de ejemplo se enumeran los conjuntos de reglas de coincidencia para la entidad de negocio de persona:

```
GET http://localhost:8080/cmx/queryTemplate/localhost-orcl-DS_UI1/Person
```

Solicitud para las columnas de coincidencia

Con la siguiente solicitud de ejemplo se enumeran las columnas de coincidencia que se utilizan en el conjunto de reglas de coincidencia IDL2:

```
GET http://localhost:8080/cmX/queryTemplate/localhost-orcl-DS_UI1/Person/IDL2
```

Ejemplo de respuesta de API

La siguiente respuesta de ejemplo contiene las columnas de coincidencia que se incluyen en el conjunto de reglas de coincidencia IDL2:

```
{
  "queryTemplates": [
    {
      "businessEntity": "Person",
      "matchRuleSet": "IDL2",
      "type": "extended",
      "searchFields": [
        {
          "name": "displayName",
          "mandatory": true
        },
        {
          "name": "BillAddresses.Address.addressLine1",
          "mandatory": false
        },
        {
          "name": "ShipAddresses.Address.addressLine2",
          "mandatory": false
        },
        {
          "name": "ShipAddresses.Address.addressLine1",
          "mandatory": false
        }
      ]
    }
  ]
}
```

Leer registros

La API de REST de lectura de registros devuelve los detalles de un registro raíz en la entidad de negocio. Puede utilizar la API para devolver los detalles de los registros secundarios de un registro raíz. Igualmente, la API también se puede utilizar para ver los metadatos del contenido de un registro.

La API utiliza el método GET.

Puede ordenar el conjunto de resultados para ver la información en orden ascendente o descendente. Utilice el método POST cuando necesite parámetros complejos y en mayor cantidad. Por ejemplo, si desea recuperar los datos en los que los elementos secundarios se ordenan por un conjunto de campos.

URL de solicitud

Utilice el ID de fila o el nombre del sistema de origen y la clave de fuente para especificar el registro en la URL de solicitud.

La URL de lectura de registros puede tener los siguientes formatos:

URL con rowId

Utilice el siguiente formato de URL al especificar el ID de fila:

```
http://<host>:<port>/<context>/<database ID>/<business entity>/<rowId of the root record>
```

Haga la siguiente solicitud HTTP GET a la URL:

```
GET http://<host>:<port>/<context>/<database ID>/<business entity>/<rowId of the root record>
```

URL con nombre del sistema de origen y clave de fuente

Utilice el siguiente formato de URL al especificar el nombre del sistema de origen y la clave de fuente:

```
http://<host>:<port>/<context>/<database ID>/<business entity>/<system name>:<source key>
```

URL con el nombre de sistema y el identificador de negocio global (GBID) de un objeto

Utilice el siguiente formato de URL al especificar el nombre del sistema de origen y el GBID:

```
http://<host>:<port>/<context>/<database ID>/<business entity>/<system name>:uid:<gbid>
```

URL con solo el GBID

Utilice el siguiente formato de URL al especificar solo el GBID:

```
http://<host>:<port>/<context>/<database ID>/<business entity>/:uid:<gbid>
```

URL con más de un GBID

Utilice el siguiente formato de URL al especificar más de un GBID:

```
http://<host>:<port>/<context>/<database ID>/<business entity>/:one:<gbid>,another:<gbid>
```

URL para devolver los detalles de los nodos secundarios

Utilice el siguiente formato de URL para devolver los detalles de los nodos secundarios:

```
http://<host>:<port>/<context>/<database ID>/<business entity>/<rowId of the record?>depth=n
```

URL para devolver los detalles de un nodo secundario

Utilice el siguiente formato de URL para devolver los detalles de los nodos secundarios:

```
http://<host>:<port>/<context>/<database ID>/<business entity>/<rowId of the record?>children=<comma separated list of child node names or paths>
```

Por ejemplo, children= BillAddresses/Address,Email

URL para devolver los detalles de un nodo concreto

Utilice el siguiente formato de URL para devolver los detalles de un nodo concreto:

```
http://<host>:<port>/<context>/<database ID>/<business entity>/<rowId of the record?>/<node name>
```

URL para devolver los detalles de los elementos secundarios de un nodo concreto

Utilice el siguiente formato de URL para devolver los detalles de los elementos secundarios de un nodo concreto:

```
http://<host>:<port>/<context>/<database ID>/<business entity>/<rowId of the record?>/<node name?>children=<child node name>
```

URL para devolver los metadatos de contenido de un registro

Utilice el siguiente formato de URL para devolver los metadatos de contenido de un registro:

```
http://<host>:<port>/<context>/<database ID>/<business entity>/<rowId of the root record?>contentMetadata=<content metadata type>
```

Por ejemplo, puede recuperar coincidencias para los registros secundarios con la siguiente solicitud GET:

```
GET http://<host>:<port>/<context>/<database ID>/<business entity>/<rowId of the root record>?contentMetadata=MATCH
```

URL para ordenar los elementos secundarios por campos

Utilice el siguiente formato de URL para ordenar los elementos secundarios por campos:

```
http://<host>:<port>/<context>/<database ID>/<business entity>/<rowId of the root record>/<node name>?order=-<field name>
```

Utilice el carácter - como sufijo para especificar el orden descendente.

Parámetros de consulta

Puede anexar los parámetros de consulta a la URL de solicitud para filtrar los detalles del registro.

En la tabla siguiente se muestran los parámetros de la consulta:

Parámetro	Descripción
depth	El número de niveles secundarios que se devolverán. Especifique 2 para devolver el nodo raíz y sus elementos secundarios directos, y 3 para devolver el nodo raíz, el elemento secundario directo y los elementos secundarios de segundo nivel. El valor predeterminado es 1.
effectiveDate	La fecha cuyos datos desea recuperar.
readSystemFields	Indica si desea devolver los campos del sistema en el resultado. El valor predeterminado es false.
recordStates	El estado del registro. Proporcione una lista de estados separados por comas. Los estados de registro admitidos son ACTIVE, PENDING y DELETED. El valor predeterminado es ACTIVE.
contentMetadata	Los metadatos del registro. Proporcione una lista de valores separados por comas. Por ejemplo, XREF, PENDING_XREF, DELETED_XREF, HISTORY, XREF_HISTORY y MATCH. Si selecciona MATCH, la respuesta contiene una lista de registros coincidentes recuperados de la tabla _MTCH.
historyDate	Fecha cuyos datos del historial desea recuperar. La respuesta contiene datos del registro para la fecha especificada que se han recuperado de la tabla _HIST. Puede usar historyDate con el parámetro contentMetadata para recuperar metadatos históricos. Establezca contentMetadata en XREF, BVT o TRUST. - XREF. La respuesta contiene datos de referencias cruzadas históricos de la tabla _HXRF. - BVT. La respuesta contiene la mejor versión de confianza histórica de la tabla _HCTL. - TRUST. La respuesta contiene valores de configuración de confianza históricos de las tablas _HCTL y _HVXR.
children	Lista de los nombres de nodos secundarios o rutas separados por comas.

Parámetro	Descripción
suppressLinks	<p>Indica si los vínculos entre elementos primarios y secundarios se pueden ver en la respuesta de la API. Establezca el parámetro en true para suprimir todos los vínculos entre elementos primarios y secundarios en la respuesta. El valor predeterminado es false.</p> <p>Por ejemplo, la consulta <code>Person/1242?depth=10&suppressLinks=true</code> mostrará los detalles del registro de hasta 10 niveles secundarios, sin incluir vínculos entre elementos primarios y secundarios en la respuesta.</p>
order	<p>Lista de nombres de campo separados por comas con el prefijo opcional + o -. El prefijo + indica la ordenación ascendente de los resultados y el prefijo - indica la ordenación descendente de los resultados. El valor predeterminado es +. Si especifica más de un parámetro, el conjunto de resultados se ordena por el primer parámetro de la lista, seguido del segundo.</p> <p>Por ejemplo, la consulta <code>Person/1242/Names?order=-name</code> muestra el resultado con los nombres en orden descendente.</p> <p>La consulta <code>Person/1242/BillAddresses?order=rowidObject,-effStartDate</code> mostrará las direcciones de facturación con los valores de RowId en orden ascendente y la fecha de inicio efectiva en orden descendente.</p>
resolveLookup	<p>Recupera todo el campo de búsqueda de una entidad de negocio específica. Establezca el parámetro en true para cargar el campo de búsqueda e incluirlo en la respuesta. El valor predeterminado es false.</p> <p>Por ejemplo, el campo <code>addressType</code> es un campo de búsqueda en el nivel secundario de la entidad de negocio de persona.</p> <p>Cuando el parámetro <code>resolveLookup</code> se establece en false, es posible que reciba la siguiente respuesta de la API REST:</p> <pre> { "label": "LU Address Type", "addressType": "BILL" } </pre> <p>Cuando el parámetro <code>resolveLookups</code> se establece en true, la respuesta de la API REST incluye detalles adicionales y es posible que reciba la siguiente respuesta de la API REST:</p> <pre> { "label": "LU Address Type", "addressType": "BILL", "addressTypeDisp": "BILLING" } </pre>

En el siguiente ejemplo se muestra cómo se deben filtrar los detalles de un registro:

```
GET http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/Person/123/Phone/SFA:456/PhoneUse?recordsToReturn=100&recordStates=ACTIVE,PENDING&contentMetadata=XREF
```

TEMAS RELACIONADOS

- [“Formatos de fecha y hora UTC” en la página 36](#)

Solicitud POST para especificar el orden de clasificación de los elementos secundarios

Utilice una solicitud POST para ordenar el conjunto de resultados en varios campos. Incluya los parámetros o los campos en el cuerpo de la solicitud POST.

En la siguiente solicitud de ejemplo, se muestra cómo utilizar la solicitud POST para una operación de lectura y ordenar los datos en varios campos:

```
http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/ReadPerson.json
{
  parameters:
  {
    coFilter: {
      object: {
        name: "Person",
        key: {
          rowid: 1242
        },
        order: "lastName",
        object: [
          {name: "Names", order: "-name"},
          {name: "Phone", order: "phoneNum, -phoneCountryCd",
            object: [{name: "PhonePermissions", order: "-column1"}]}
        ]
      }
    }
  }
}
```

Nota: En cada nivel de la entidad de negocio, solo se permite un tipo de orden de clasificación por cada tipo de elemento secundario.

Consideraciones del orden de clasificación

La API de lectura de registros es compatible con la ordenación de clasificación en uno o más campos para cada nodo secundario de entidad de negocio. La siguiente sección describe ciertas consideraciones que debe tomar en cuenta al especificar el orden de clasificación.

- Si especifica el orden de clasificación del elemento secundario de segundo nivel, pero no para el elemento secundario, el elemento secundario de segundo nivel se organiza en el orden especificado. El elemento secundario no se organiza en el orden de clasificación que ha especificado para el elemento secundario de segundo nivel. A continuación se muestra una solicitud de ejemplo:

```
http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/Person/1242/Phone/861/
PhonePermissions?order=-column1
```

En la solicitud de ejemplo, a pesar de que se ha especificado un orden de clasificación descendente para el elemento secundario de segundo nivel PhonePermissions, no se especifica ninguno para el elemento secundario Phone. Phone no se ha organizado en el orden de clasificación de PhonePermissions.

- Si especifica el orden de clasificación del elemento secundario y no el del elemento secundario de segundo nivel, el elemento secundario se organiza en el orden de clasificación que se ha especificado. El elemento secundario de segundo nivel no se organiza en el orden de clasificación que ha especificado para el elemento secundario. A continuación se muestra una solicitud de ejemplo:

```
{parameters:
  {coFilter: {
    object: {
      name: "Person", key: { rowid: 1242 }, order: "lastName",
```

```

    object:[
      {name:"Names", order:"-name"},
      {name:"Phone", order:"-phoneCountryCd, -phoneNum", object:
    [{name:"PhonePermissions"}]},
    ]}
  ]}
}}

```

En la solicitud de ejemplo, se especifica el orden de clasificación para el elemento secundario Phone, pero no para el elemento secundario de segundo nivel PhonePermissions. El elemento secundario Phone se organiza en el orden que se ha especificado.

- Si especifica el orden de clasificación del elemento secundario y del elemento secundario de segundo nivel, ambos se organizan por el orden de clasificación. La siguiente solicitud de ejemplo especifica el orden de clasificación de Phone (elemento secundario) y de PhonePermissions (elemento secundario de segundo nivel):

```

{parameters:
  {coFilter: {
    object: {
      name:"Person", key: { rowid: 1242 }, order: "lastName",
      object:[
        {name:"Names", order:"-name"},
        {name:"Phone", order:"-phoneCountryCd, -phoneNum", object:
    [{name:"PhonePermissions", order:"-column1"}]},
    ]}
  }}}

```

- Un elemento secundario solo se puede ordenar por las columnas del propio elemento secundario, mientras que el elemento secundario de segundo nivel se puede ordenar por las columnas del elemento secundario de segundo nivel. En los siguientes ejemplos de solicitud, Phone se ordena por PhoneType y PhonePermissions se ordena por la columna 1. PhoneType es una columna de Phone (elemento secundario) y la columna 1 es una columna de PhonePermissions (elemento secundario de segundo nivel).

```

http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/Person/1242/Phone?order=-PhoneType
http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/Person/1242/Phone/861/
phonePermissions?order=column1

```

- En cada nivel de la entidad de negocio, solo se permite un tipo de orden de clasificación por cada tipo de elemento secundario. En la siguiente solicitud, se han especificado diferentes tipos de órdenes de clasificación para los elementos secundarios de PhonePermissions de diferentes elementos primarios. Sin embargo, puesto que el primer orden de clasificación se ha establecido como descendiente, los elementos secundarios de PhonePermissions de ambos elementos primarios (RowId 861 y RowId 862) se ordenan de esta manera.

```

{parameters:
  {coFilter: {
    object: {
      name:"Person", key: { rowid: 1242 }, order: "lastName",
      object:[
        {name:"Names", order:"-name"},
        {name:"Phone", key: { rowid:861 }, order:"+phoneCountryCd, -phoneNum", object:
    [{name:"PhonePermissions", order:"-column1"}]},
        {name:"Phone", key: {rowid:862}, order:"phoneNum, -phoneCountryCd", object:
    [{name:"PhonePermissions", order:"column1"}]}
      ]}
  }}}

```

Ejemplo de solicitudes de API

La siguiente solicitud de ejemplo devuelve los detalles del registro raíz en la entidad de negocio de persona.

```
GET http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/Person/102
```


La siguiente solicitud de ejemplo devuelve los detalles de un registro secundario con rowId 2. El objeto base secundario es genderCd y el registro secundario se encuentra en un nivel 2.

```
GET http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/Person/102/genderCd/2?depth=2
```

La siguiente solicitud de ejemplo utiliza el nombre del sistema y la clave de fuente para devolver los detalles del registro raíz:

```
GET http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/Person/SFA:9000000000
```

La siguiente solicitud de ejemplo devuelve los detalles del registro raíz y sus registros XREF:

```
GET http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/Person/102?contentMetadata=XREF
```

La siguiente solicitud de ejemplo devuelve los detalles del registro raíz con los nombres ordenados de manera descendente:

```
GET http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/Person/1242/Names?order=-name
```

La siguiente solicitud de ejemplo devuelve las direcciones de facturación con los valores de RowId ordenados de manera ascendente y las fechas de inicio efectivas ordenadas de manera descendente:

```
GET http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/Person/1242/BillAddresses?order=rowidObject,-effStartDate
```

Ejemplo de respuesta de API

El siguiente ejemplo muestra los detalles del registro raíz en la entidad de negocio de persona.

```
{
  "link": [
    {
      "href": "http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/Person/102",
      "rel": "self"
    },
    {
      "href": "http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/Person/102?depth=2",
      "rel": "children"
    },
    {
      "href": "http://localhost:8080/cmx/request/hm_icons/person_small.jpeg?ors=localhost-orcl-DS_UI1",
      "rel": "icon"
    }
  ],
  "rowidObject": "102",
  "label": "DARWENT, JIMMY",
  "partyType": "Person",
  "statusCd": "A",
  "lastName": "DARWENT",
  "middleName": "N",
  "firstName": "JIMMY",
  "displayName": "JIMMY N DARWENT",
  "genderCd": {
    "link": [
      {
        "href": "http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/Person/102",
        "rel": "parent"
      },
      {
        "href": "http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/Person/102/genderCd",
        "rel": "self"
      },
      {
        "href": "http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/Person/102/genderCd?depth=2",
        "rel": "children"
      }
    ]
  }
}
```

```

    ],
    "genderCode": "M"
  },
  "generationSuffixCd": {
    "link": [
      {
        "href": "http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/Person/102",
        "rel": "parent"
      },
      {
        "href": "http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/Person/102/
generationSuffixCd?depth=2",
        "rel": "children"
      },
      {
        "href": "http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/Person/102/
generationSuffixCd",
        "rel": "self"
      }
    ],
    "generationSuffixCode": "I"
  }
}

```

El siguiente ejemplo muestra los detalles de un registro secundario del objeto base genderCd.

```

{
  "link": [
    {
      "href": "http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/Person/102",
      "rel": "parent"
    },
    {
      "href": "http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/Person/102/
genderCd/2?depth=2",
      "rel": "children"
    },
    {
      "href": "http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/Person/102/
genderCd/2",
      "rel": "self"
    }
  ],
  "rowidObject": "2",
  "label": "LU Gender",
  "genderDisp": "MALE",
  "genderCode": "M"
}

```

Crear registros

La API de REST de creación de registros crea un registro en la entidad de negocio especificada. Envíe los datos del registro en el cuerpo de la solicitud. Utilice la API para la promoción para promover y añadir el registro en la entidad de negocio.

La API utiliza el método POST para crear un registro.

URL de solicitud

La dirección URL de creación de registros tiene el siguiente formato:

```

http://<host>:<port>/<context>/<database ID>/<business entity>?systemName=<name of the
source system>

```

Nota: El nombre del sistema de origen es un parámetro obligatorio en la URL.

Haga la siguiente solicitud HTTP POST a la dirección URL de creación de registros:

```
POST http://<host>:<port>/<context>/<database ID>/<business entity>?systemName=<name of the source system>
```

Agregue el encabezado Content-Type para especificar el tipo de medio de los datos que desea enviar con la solicitud:

```
POST http://<host>:<port>/<context>/<database ID>/<business entity>?systemName=<name of the source system>
Content-Type: application/<json/xml>
```

Parámetros de URL

El nombre del sistema de origen es un parámetro obligatorio en la URL de solicitud.

La siguiente tabla muestra los parámetros que puede usar en la URL:

Parámetro	Descripción
systemName	Nombre del sistema de origen.
interactionId	ID de la interacción. Puede agrupar varias solicitudes en una única interacción. Todos los cambios se realizan con el ID de interacción.
ignoreWarnings	Indique si se omiten los mensajes de validación a nivel de advertencia. El valor predeterminado es false. Establezca en true para omitir los mensajes de advertencia.
startDate y endDate	Especifica el periodo de tiempo en el que el registro es efectivo. Proporcione estos parámetros para un objeto base que tenga habilitada la línea temporal.
validateOnly	Indica si el servicio de entidad de negocio de escritura valida los datos entrantes. El valor predeterminado es false.
recordState	El estado del registro. Utilice el parámetro para especificar el estado inicial del registro. Utilice ACTIVE o PENDING. El valor predeterminado es ACTIVE.
taskComment	Añada un comentario a la tarea de flujo de trabajo que ha activado la API.
taskAttachments	Si los adjuntos de tarea están habilitados, adjunte un archivo a la tarea de flujo de trabajo que ha activado la API.

TEMAS RELACIONADOS

- [“Formatos de fecha y hora UTC” en la página 36](#)

Cuerpo de la solicitud

Envíe los datos para el registro en el cuerpo de la solicitud de REST. Utilice el formato JSON o el formato XML para enviar los datos. Utilice la API de obtención de metadatos para obtener la estructura de la entidad de negocio y proporcionar los valores de parámetro que se necesitan en el cuerpo de la solicitud.

Después de enviar una solicitud de API REST para crear un registro de entidad de negocio, Informatica MDM Hub calcula la puntuación de confianza del campo. Puede enviar una solicitud de API REST para campos vacíos con puntuaciones de confianza calculadas. Debe especificar comillas dobles en el cuerpo de la solicitud.

El siguiente ejemplo de código JSON solicita el campo Código de género vacío con una puntuación de confianza calculada:

```
{
  "firstName": "Sasha",
  "genderCd": ""
}
```

Si utiliza un valor nulo en lugar de comillas dobles, Informatica MDM Hub no calcula la puntuación de confianza del campo.

Cuerpo y encabezado de respuesta

Cuando la respuesta sea satisfactoria, la API devolverá el ID de interacción (interactionId) y el ID de proceso (processId) en el encabezado de la respuesta y los detalles del registro en el cuerpo de la respuesta.

Si el proceso genera un ID de interacción y lo utiliza para crear el registro, la API devolverá el ID de interacción. Si el proceso inicia un flujo de trabajo en lugar de guardar directamente el registro en la base de datos, la API devolverá el ID de proceso, que es el ID del proceso de flujo de trabajo.

En el ejemplo siguiente se muestra un encabezado de respuesta con un ID de interacción y un ID de proceso:

```
BES-interactionId: 72200000242000
BES-processId: 15948
```

El cuerpo de la respuesta contiene el registro con los ID de fila generados.

Ejemplo de solicitudes de API

La siguiente solicitud de ejemplo crea un registro en la entidad de negocio de persona: La solicitud añade un comentario y un archivo adjunto a la tarea del flujo de trabajo que activa la API.

```
POST http://localhost:8080/cmX/cs/localhost-orcl-DS_UI1/Person?
systemName=Admin&taskComment=Read my
comment&taskAttachments=TEMP_SVR1.29OT8,TEMP_SVR1.29OT9
Content-Type: application/json

{
  "firstName": "John",
  "lastName": "Smith",
  "Phone": {
    "item": [
      {
        "phoneNumber": "111-11-11"
      }
    ]
  }
}
```

Cuando utiliza una solicitud POST de API REST para crear un nuevo registro, puede establecer el atributo `sourceKey` en el cuerpo de la solicitud POST. El atributo `sourceKey` se utiliza como el valor `pkey_src_object` en la tabla XREF. Por ejemplo, puede enviar la siguiente solicitud POST para crear un nuevo registro Party y el registro Party XREF con el sistema de origen SFA y `pkey_src_object` 123:

```
POST http://localhost:8080/cmX/cs/localhost-orcl-DS_UI1/Person?systemName=SFA

{
  "firstName": "John",
  "middleName": "jr",
  "lastName": "Smith",
  "genderCd": {
    "genderCode": "M"
  },
  "key": {
    "sourceKey": "123"
  }
}
```

```
}  
}
```

Ejemplo de respuesta de API

La siguiente respuesta de ejemplo muestra el encabezado y el cuerpo de la respuesta después de crear correctamente un registro:

```
BES-interactionId: 72200000242000  
BES-processId: 15948  
Content-Type: application/json  
{  
  "Person": {  
    "key": {  
      "rowid": "2198246",  
      "sourceKey": "72200000241000"  
    },  
    "rowidObject": "2198246",  
    "Phone": {  
      "item": [  
        {  
          "key": {  
            "rowid": "260961",  
            "sourceKey": "72200000243000"  
          },  
          "rowidObject": "260961"  
        }  
      ]  
    }  
  }  
}
```

Actualizar registros

La API de REST de actualización de registros actualiza el registro raíz especificado y sus registros secundarios. Envíe el ID del registro en la URL de solicitud. Envíe el resumen de los cambios en el cuerpo de la solicitud.

Después del cambio, si el registro está en un estado pendiente, use la API de promoción para promover los cambios. Por ejemplo, si la actualización desencadena un flujo de trabajo de revisión, el registro está en el estado pendiente hasta que se completa la revisión.

La API usa el método POST.

Para los registros raíz, puede decidir usar la versión de PUT simplificada, donde el cuerpo de la solicitud contiene el campo cambiado, pero no un resumen del cambio. Para actualizar los registros raíz y los registros secundarios o secundarios de segundo nivel, utilice la versión de POST. También tiene la opción de utilizar la versión de PUT con PKEY o el rowidObject especificado del registro secundario.

URL de solicitud

La dirección URL de actualización de registros tiene el siguiente formato:

```
http://<host>:<port>/<context>/<database ID>/<business entity>/<rowId>?systemName=<name  
of the source system>
```

Nota: El nombre del sistema de origen es un parámetro obligatorio en la URL.

Haga la siguiente solicitud HTTP POST a la dirección URL de actualización de registro:

```
POST http://<host>:<port>/<context>/<database ID>/<business entity>/<rowId>?  
systemName=<name of the source system>
```

Agregue el encabezado Content-Type para especificar el tipo de medio de los datos que desea enviar con la solicitud:

```
PUT http://<host>:<port>/<context>/<database ID>/<business entity>/<rowId>?
systemName=<name of the source system>
Content-Type: application/<json/xml>
```

Parámetros de consulta

El nombre del sistema de origen es un parámetro de consulta obligatorio.

Puede usar los siguientes parámetros de consulta en la solicitud:

Parámetro	Descripción
systemName	Nombre del sistema de origen.
interactionId	ID de la interacción. Puede agrupar varias solicitudes en una única interacción. Todos los cambios se realizan con el ID de interacción.
ignoreWarnings	Indique si se omiten los mensajes de validación a nivel de advertencia. El valor predeterminado es false. Establezca en true para omitir los mensajes de advertencia.
startdate y enddate	Especifica el periodo de tiempo en el que el registro es efectivo. Proporcione estos parámetros para un objeto base que tenga habilitada la línea temporal.
validateOnly	Indica si los datos entrantes se validaron. El valor predeterminado es false.
recordState	Establece el estado del registro. Utilice ACTIVE, PENDING o DELETED. Por ejemplo, si establece recordState=ACTIVE y la solicitud se ejecuta en un registro eliminado temporalmente, la solicitud devuelve el registro al estado activo.
taskComment	Añada un comentario a la tarea de flujo de trabajo que ha activado la API.
taskAttachments	Si los adjuntos de tarea están habilitados, adjunte un archivo a la tarea de flujo de trabajo que ha activado la API.

TEMAS RELACIONADOS

- [“Formatos de fecha y hora UTC” en la página 36](#)

Cuerpo de la solicitud

Envíe los datos que se van a actualizar en el cuerpo de la solicitud REST. Utilice el formato JSON o el formato XML para enviar los datos.

Proporcione los nuevos valores de parámetro. Utilice el parámetro \$original para indicar el valor anterior de un parámetro que desee actualizar.

Además, puede usar las siguientes propiedades con los registros secundarios:

Propiedades/elementos	Tipo	Descripción
MATCH	object	Si desea añadir o quitar candidatos de coincidencia de la tabla de coincidencia para el registro secundario, añada el objeto MATCH al registro secundario.
MERGE	object	Si desea fusionar registros secundarios o quitar candidatos de la fusión, añada el objeto MERGE al registro secundario.

En el siguiente ejemplo de código JSON se cambia el nombre en el registro raíz a Bob:

```
{
  rowidObject: "123",
  firstName: "Bob",
  lastName: "Smith",
  $original: {
    firstName: "John"
  }
}
```

En el siguiente ejemplo de código JSON se quita un candidato de coincidencia de un registro secundario de dirección y se define la fusión para dos registros secundarios de número de teléfono:

```
{
  rowidObject: "123",
  firstName: "Bob",
  lastName: "Smith",
  $original: {
    firstName: "John"
  }
  Address: { // remove A3 from the matches for A2 in the Address_MTCH table
    item: [
      {
        rowidObject: "A2",
        MATCH: {
          item: [ // to remove matched child records for A2, specify null
            null
          ],
          $original: {
            item: [{key: {rowid: 'A3'}}]
          }
        }
      }
    ]
  }
  Telephone: { // override the matches for the telephone child records
    item:[
      {
        rowid: "T1",
        MERGE: {
          item: [ // to remove merge candidates for T1, specify null
            null,
            null
          ],
          $original: {
            item: [
              {rowid: "T2"},
              {rowid: "T3"}
            ]
          }
        }
      }
    ],
    {
      rowid: "T4",
      MERGE: {

```

```

        item: [ // to add or override matches, specify matched records
          {rowid: "T2"}
        ],
        $original: {
          item: [
            null
          ]
        }
      ]
    }
  ]
}

```

Después de enviar una solicitud de API REST para actualizar un registro de entidad de negocio, Informatica MDM Hub calcula la puntuación de confianza del campo. Puede enviar una solicitud de API REST para campos vacíos con puntuaciones de confianza calculadas. Debe especificar comillas dobles en el cuerpo de la solicitud.

El siguiente ejemplo de código JSON solicita el campo Código de género vacío con una puntuación de confianza calculada:

```

{
  "firstName": "Sasha",
  "genderCd": ""
}

```

Si utiliza un valor nulo en lugar de comillas dobles, Informatica MDM Hub no calcula la puntuación de confianza del campo.

Encabezado de respuesta

Cuando la respuesta sea satisfactoria, la API devolverá el ID de interacción (interactionId) y el ID de proceso (processId) en el encabezado de la respuesta y los detalles del registro en el cuerpo de la respuesta.

Si el proceso genera un ID de interacción y lo utiliza para crear el registro, la API devolverá el ID de interacción. Si el proceso inicia un flujo de trabajo en lugar de guardar directamente el registro en la base de datos, la API devolverá el ID de proceso, que es el ID del proceso de flujo de trabajo.

En el ejemplo siguiente se muestra un encabezado de respuesta con un ID de interacción y un ID de proceso:

```

BES-interactionId: 72200000242000
BES-processId: 15948

```

El cuerpo de la respuesta contiene el registro con los ID de fila generados.

Ejemplo de solicitud de API

La siguiente solicitud de ejemplo actualiza un registro raíz y su registro secundario en una entidad de negocio. La persona es la entidad de negocio y el teléfono es un objeto base secundario. La solicitud añade un comentario y un adjunto a la tarea de flujo de trabajo que ha activado la API.

```

POST http://localhost:8080/cmxcsllocalhost-orcl-DS_UI1/Person/233?systemName=Admin
{
  rowidObject: "233",
  firstName: "BOB",
  lastName: "LLOYD",
  Phone: {
    item: [
      {
        rowidObject: "164",
        phoneNumber: "777-77-77",
        $original: {
          phoneNumber: "(336)366-4936"
        }
      }
    ]
  }
}

```



```

    }
  ]
},
$original: {
  firstName: "DUNN"
}
}

```

Ejemplo de respuesta de API

La siguiente respuesta de ejemplo muestra el encabezado y el cuerpo de la respuesta después de actualizar correctamente un registro:

```

BES-interactionId: 72300000001000
BES-processId: 16302
{
  Person: {
    key: {
      rowid: "233",
      sourceKey: "SYS:233"
    },
    rowidObject: "233",
    preferredPhone: {
      key: {}
    }
  }
}

```

Eliminar registros

La API de REST de eliminación de registros elimina un registro raíz en una entidad de negocio. Utilice la API para eliminar los registros secundarios de un registro raíz.

La API utiliza el método DELETE para eliminar un registro.

URL de solicitud

La dirección URL de eliminación de registros tiene el siguiente formato:

```

http://<host>:<port>/<context>/<database ID>/<business entity>/<rowID of the root
record>?systemName=Admin

```

Nota: El nombre del sistema de origen es un parámetro obligatorio en la URL.

Haga la siguiente solicitud HTTP DELETE a la dirección URL de eliminación de registros:

```

DELETE http://<host>:<port>/<context>/<database ID>/<business entity>/<rowID of the
record>?systemName=Admin

```

Utilice el siguiente formato de URL para eliminar un registro secundario de un registro raíz:

```

DELETE http://<host>:<port>/<context>/<database ID>/<business entity>/<rowID of the
record>/<child base object>/<rowID of the child record>?systemName=Admin

```

Parámetro de consulta

El nombre del sistema de origen es un parámetro de URL obligatorio. Utilice el parámetro `systemName` para especificar el sistema de origen.

Ejemplo de solicitud de API

La siguiente solicitud de ejemplo elimina un registro raíz de la entidad de negocio de persona:

```
DELETE http://localhost:8080/cmxcsllocalhost-orcl-DS_UI1/Person/292258?systemName=Admin
```

Ejemplo de respuesta de API

La siguiente respuesta de ejemplo muestra la respuesta después de eliminar correctamente un registro raíz de la entidad de negocio de persona:

```
{
  "Person": {
    "key": {
      "rowid": "292258",
      "sourceKey": "WRK50000_7016"
    },
    "rowidObject": "292258"
  }
}
```

Lista Registro

La API de REST de enumeración de registros devuelve la lista de valores de búsqueda o valores de clave externa. Las búsquedas proporcionan datos de referencia y ofrecen una lista de posibles valores para una determinada columna.

La API utiliza el método GET.

Utilice la API para obtener los valores y las descripciones de los códigos de búsqueda. Puede especificar el orden de clasificación de las búsquedas. Utilice el método POST cuando necesite parámetros complejos y en mayor cantidad.

URL de solicitud

La dirección URL de REST de enumeración de registros tiene el siguiente formato:

```
http://<host>:<port>/<context>/<database ID>/<lookup table>?action=list
```

Haga la siguiente solicitud HTTP GET a la URL:

```
GET http://<host>:<port>/<context>/<database ID>/<lookup table>?action=list
```

Utilice el siguiente formato de URL para enumerar los códigos de los valores de búsqueda:

```
http://<host>:<port>/<context>/<database ID>/<lookup table>?action=list&idlabel=<lookup code>%3A<lookup display name>
```

Nota: Utilice la API de obtención de metadatos para obtener la URL exacta y enumerar los valores de búsqueda.

Parámetros de consulta

Para filtrar los resultados, puede anexar los parámetros de consulta a la URL de solicitud.

En la tabla siguiente se muestran los parámetros de la consulta:

Parámetro	Descripción
suppressLinks	<p>Indica si los vínculos entre elementos primarios y secundarios se pueden ver en la respuesta de la API. Establezca el parámetro en true para suprimir todos los vínculos entre elementos primarios y secundarios en la respuesta. El valor predeterminado es false.</p> <p>Por ejemplo, la consulta <code>Person/1242?depth=10&suppressLinks=true</code> mostrará los detalles del registro de hasta 10 niveles secundarios, sin incluir vínculos entre elementos primarios y secundarios en la respuesta.</p>
order	<p>Se utiliza para enumerar los valores de búsqueda en orden ascendente o descendente. Utilice el carácter + como prefijo para especificar el orden ascendente y el carácter - como prefijo para especificar el orden descendente. Si no especifica el prefijo, el conjunto de resultados se ordena en orden ascendente de manera predeterminada.</p> <p>Por ejemplo, la consulta <code>LUNamePrefix?action=list&order=-namePrefixDisp</code> enumera los prefijos por nombre, los cuales se ordenan por los nombres para mostrar de los prefijos en orden descendente.</p>
resolveLookup	<p>Recupera todo el campo de búsqueda de una entidad de negocio específica. Establezca el parámetro en true para cargar el campo de búsqueda e incluirlo en la respuesta. El valor predeterminado es false.</p> <p>Por ejemplo, el campo <code>addressType</code> es un campo de búsqueda en el nivel secundario de la entidad de negocio de persona.</p> <p>Cuando el parámetro <code>resolveLookup</code> se establece en false, es posible que reciba la siguiente respuesta de la API REST:</p> <pre>{ "label": "LU Address Type", "addressType": "BILL" }</pre> <p>Cuando el parámetro <code>resolveLookups</code> se establece en true, la respuesta de la API REST incluye detalles adicionales y es posible que reciba la siguiente respuesta de la API REST:</p> <pre>{ "label": "LU Address Type", "addressType": "BILL", "addressTypeDisp": "BILLING" }</pre>

Solicitud POST para especificar el orden de clasificación

Utilice la solicitud POST para especificar el orden de clasificación de los valores de búsqueda. Incluya los parámetros o los campos en el cuerpo de la solicitud POST.

El siguiente ejemplo describe la manera en la que se debe utilizar la solicitud POST para una operación de lista:

```
http://localhost:8080/cmxc/cs/localhost-orcl-DS_UI1/ListCO.json
{
```

```

parameters:
{
  coFilter: {
    object: {
      name: "LUCountry",
      order: "-countryNameDisp"
    }
  }
}
}

```

Ejemplo de solicitud de API

La siguiente solicitud de ejemplo enumera los valores de búsqueda:

```
GET http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/LUGender?action=list
```

La siguiente solicitud de ejemplo enumera los códigos correspondientes al género de los valores de búsqueda:

```
GET http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/LUGender?
action=list&idlabel=genderCode%3AgenderDisp
```

La siguiente solicitud de ejemplo enumera los prefijos para nombres con sus nombres para mostrar en orden descendente:

```
GET http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/LUNamePrefix?action=list&order=-
namePrefixDisp
```

Ejemplo de respuesta de API

La siguiente respuesta de ejemplo muestra la lista de valores de búsqueda:

```

{
  "firstRecord": 1,
  "pageSize": 2147483647,
  "searchToken": "SVR1.AU5LC",
  "item": [
    {
      "rowidObject": "1",
      "genderDisp": "UNKNOWN",
      "genderCode": "N"
    },
    {
      "rowidObject": "2",
      "genderDisp": "MALE",
      "genderCode": "M"
    },
    {
      "rowidObject": "3",
      "genderDisp": "FEMALE",
      "genderCode": "F"
    }
  ]
}

```

La siguiente respuesta de ejemplo muestra el código para los valores de búsqueda:

```

{
  "item": [
    {
      "id": "F",
      "label": "FEMALE"
    },
    {
      "id": "M",
      "label": "MALE"
    }
  ]
}

```

```

        "id": "N",
        "label": "UNKNOWN"
    }
],
"firstRecord": 1,
"recordCount": 0,
"pageSize": 2147483647,
"searchToken": "SVR1.AU5LD"
}

```

Búsqueda Registro

La API de REST de búsqueda de registros busca los valores indexados en un registro raíz que admite búsquedas y en todos los registros secundarios. Puede utilizar filtros y facetas para ver un subconjunto de los resultados de la búsqueda. Las facetas agrupan los resultados de la búsqueda en categorías, mientras que los filtros los acotan. La API busca en todos los campos que se hayan configurado para realizar búsquedas y devuelve los registros que coincidan con los criterios de búsqueda.

La API utiliza el método GET para buscar los índices de los campos en los que se pueden realizar búsquedas.

URL de solicitud

La dirección URL de búsqueda de registro tiene los siguientes formatos:

URL para una búsqueda simple

Utilice la URL siguiente para una búsqueda simple:

```
http://<host>:<port>/<context>/<database ID>/<business entity>?q=<string value>
```

Haga la siguiente solicitud HTTP GET a la dirección URL de eliminación de registro:

```
GET http://<host>:<port>/<context>/<database ID>/<business entity>?q=<string value>
```

URL para una búsqueda por campos

Utilice la URL siguiente para una búsqueda por campos:

```
http://<host>:<port>/<context>/<database ID>/<business entity>?fq=<business entity field name>='<business entity field value>'
```

Si especifica un valor numérico negativo para un campo de entidad de negocio, como -120, se podría ver afectada la clasificación de los registros que se devuelve.

URL para una búsqueda de facetas

Utilice la URL siguiente para una búsqueda simple con facetas:

```
http://<host>:<port>/<context>/<database ID>/<business entity>?q=<string value>&facets=<field name>
```

Utilice la URL siguiente para una búsqueda por campos con facetas:

```
http://<host>:<port>/<context>/<database ID>/<business entity>?fq=<business entity field name>='<business entity field value>*&facets=<field name>
```

URL para una búsqueda por filtros

Utilice la URL siguiente para una búsqueda simple con filtros:

```
http://<host>:<port>/<context>/<database ID>/<business entity>?q=<string value>&filters=<field name1>=<field value1> AND <field name2>=<field value2> ...
```

Utilice el parámetro q o fq en la búsqueda.

Codificación de URL

Debe utilizar la codificación URL porque la dirección URL incluye caracteres, como espacios y comillas simples.

El ejemplo siguiente muestra la representación codificada de URL para la dirección URL de búsqueda de registro:

```
http://<host>:<port>/<context>/<database ID>/<business entity>?q=<field name>%3D%27<value of the field>
```

Parámetros de consulta

Utilice los parámetros de consulta `q` o `fq` para proporcionar el valor de cadena para la búsqueda. Los parámetros de consulta `q` y `fq` se excluyen mutuamente. Utilice el parámetro `fq` para una búsqueda por campos. Utilice el operador lógico AND para varias condiciones.

La siguiente tabla muestra los parámetros que puede usar en la URL:

Parámetro	Descripción
<code>q</code>	<p>Especifica el valor de cadena o el término de la búsqueda. La consulta busca ocurrencias del término de la búsqueda en cualquier parte de un registro. Se utiliza en una búsqueda simple.</p> <p>Por ejemplo, la consulta <code>Person?q=STEVE</code> busca registros que incluyan el término STEVE.</p> <p>Para buscar dos o más términos a la vez, incluya los términos entre comillas dobles. Anteponga el carácter <code>+</code> a cada término si desea que los resultados de búsqueda contengan este término. Si el valor de campo contiene un espacio, incluya el valor de campo entre comillas simples.</p> <p>Utilice la siguiente consulta para encontrar una coincidencia exacta con WILLIAM JOHN LAWSON:</p> <pre>Person?q="WILLIAM JOHN LAWSON"</pre> <p>Utilice la siguiente consulta para encontrar WILLIAM, JOHN o LAWSON:</p> <pre>Person?q=WILLIAM JOHN LAWSON</pre> <p>Utilice la siguiente consulta para encontrar WILLIAM, JOHN y LAWSON:</p> <pre>Person?q=WILLIAM JOHN LAWSON&queryOperator=AND</pre> <p>Si el término de búsqueda contiene el carácter especial y comercial (<code>&</code>), codifique el carácter como <code>%26</code>. De lo contrario, la solicitud de búsqueda no devuelve los resultados esperados.</p> <p>Utilice la siguiente consulta para buscar Johnson&Johnson:</p> <pre>Organization?q=Johnson%26Johnson&queryOperator=AND</pre>
<code>fq</code>	<p>Especifica el valor de cadena o el término de búsqueda en un campo determinado. La consulta busca el término solo en esa parte de un registro. Se utiliza en una búsqueda dirigida en campos indexados.</p> <p>Por ejemplo, la consulta <code>Person?fq=displayname=STEVE</code> busca registros que incluyan el nombre para mostrar STEVE.</p> <p>Si el término de búsqueda contiene el carácter especial y comercial (<code>&</code>), codifique el carácter como <code>%26</code>. De lo contrario, la solicitud de búsqueda no devuelve los resultados esperados.</p> <p>Por ejemplo, use la siguiente consulta para buscar Mac&Cheese:</p> <pre>Product?fq=fname=Mac%26Cheese&queryOperator=AND</pre>
<code>facetas</code>	<p>Especifica los campos que se deberían tratar como facetas o categorías a partir de los cuales se agrupan los resultados de la búsqueda. Especifique solo campos que admitan búsquedas. Se utiliza con los parámetros <code>q</code> y <code>fq</code>. La sintaxis es <code>&facets=FieldName1,FieldName2,FieldNameN</code></p> <p>Por ejemplo, la consulta <code>Person?q=STEVE&facets=department</code> busca personas que incluyan el nombre para mostrar STEVE y agrupa los resultados de la búsqueda por departamentos. La búsqueda muestra los registros de las personas con el nombre para mostrar STEVE y estos registros se agrupan por departamentos.</p>

Parámetro	Descripción
filtros	<p>Especifica los campos mediante los que puede acotar los resultados de la búsqueda. Especifique solo campos que admitan filtros. Se utiliza con los parámetros q y fq.</p> <p>Por ejemplo, la consulta <code>Person?fq=displayname=STEVE&filters=birthdate='1980-11-27T08:00:00Z'</code> busca personas que incluyan el nombre para mostrar STEVE y filtra los resultados de la búsqueda por fecha de nacimiento. La búsqueda muestra los registros de las personas con el nombre para mostrar STEVE y cuya fecha de nacimiento sea el 27 de noviembre de 1980.</p> <p>Nota: Especifique una fecha entre comillas simples.</p>
depth	<p>Especifica el número de niveles secundarios que se van a devolver. Especifique 2 para devolver el nodo raíz y sus elementos secundarios directos, y 3 para devolver el nodo raíz, el elemento secundario directo y los elementos secundarios de segundo nivel. Especifique 1 para devolver el nodo raíz únicamente. De manera predeterminada, no se especifica ninguna profundidad.</p> <p>Si no se especifica ninguna profundidad, los resultados de búsqueda devuelven el nodo raíz y los elementos secundarios en los que se ha encontrado una coincidencia para el término de búsqueda.</p> <p>Por ejemplo, la consulta <code>Person?q=STEVE&depth=2</code> busca los resultados que incluyan el término STEVE y devuelve la información sobre el registro raíz y sus elementos secundarios directos.</p>
queryOperator	<p>Especifica si la búsqueda encuentra algunas de las cadenas del término de búsqueda o todas las cadenas del término de búsqueda.</p> <p>El parámetro toma uno de los siguientes valores:</p> <ul style="list-style-type: none"> - OR. Busca alguna de las cadenas que aparecen en el parámetro f o fq. - AND. Busca todas las cadenas que aparecen en el parámetro f o fq. <p>Si no especifica este parámetro, el valor predeterminado es OR.</p> <p>Por ejemplo, la consulta <code>Person?q=WILLIAM JOHN LAWSON&queryOperator=AND</code> busca los registros que contengan WILLIAM, JOHN y LAWSON.</p>
suppressLinks	<p>Indica si los vínculos entre elementos primarios y secundarios se pueden ver en la respuesta de la API. Establezca el parámetro en true para suprimir todos los vínculos entre elementos primarios y secundarios en la respuesta. El valor predeterminado es false.</p> <p>Por ejemplo, la consulta <code>Person?q=STEVE&suppressLinks=true</code> busca registros con el término STEVE y devuelve una respuesta en la que no se incluyen vínculos entre elementos primarios y secundarios.</p>
readSystemFields	<p>Indica si desea devolver los campos del sistema en el resultado. El valor predeterminado es false.</p>
order	<p>Lista de nombres de campo separados por comas con el prefijo opcional + o -. El prefijo + indica la ordenación ascendente de los resultados y el prefijo - indica la ordenación descendente de los resultados. El valor predeterminado es +.</p> <p>Si desea usar un campo secundario para ordenar los resultados, use el nombre completo del campo. Por ejemplo, <code>BillAddresses.Address.cityName</code>.</p> <p>Si especifica más de un parámetro, el conjunto de resultados se ordena por el primer parámetro de la lista, seguido del segundo. Por ejemplo, la consulta <code>Person?order=displayName,-BillAddresses.Address.cityName</code> ordena los resultados por el nombre para mostrar en orden ascendente y, a continuación, por el nombre de ciudad en orden descendente.</p>

Parámetro	Descripción
maxRecordsToSort	Número máximo de resultados de búsqueda que desea ordenar. El valor predeterminado es 1000.
resolveLookup	Recupera todo el campo de búsqueda de una entidad de negocio específica. Establezca el parámetro en true para cargar el campo de búsqueda e incluirlo en la respuesta. El valor predeterminado es false.

Especifique un rango con el parámetro `filters`:

Puede usar el parámetro `filters` para acotar los resultados de la búsqueda dentro de un rango determinado. Puede especificar el rango para los campos que admiten filtros de tipos de datos numéricos y de fecha.

Utilice el siguiente formato para el tipo de datos de enteros:

```
fieldName=[fromValue,toValue]
```

El rango oscila de `fromValue` a `toValue`. Asegúrese de que `fromValue` sea inferior a `toValue`. Por ejemplo, la consulta `filters=age=[35,45]` acota los resultados de la búsqueda y busca registros en el grupo de edades comprendido entre 35 y 45.

Utilice el siguiente formato para el tipo de datos de fecha:

```
fieldName=[fromDate,toDate]
```

El rango oscila de `fromDate` a `toDate`. Por ejemplo, la consulta `filters=birthdate=[2000-06-12T12:30:00Z,2015-06-12T12:30:00Z]` especifica la fecha de nacimiento entre el 12 de junio de 2000 y el 12 de junio de 2015.

Nota: Cuando especifique un filtro de datos de coincidencia exacta, inclúyalo entre comillas simples. Cuando especifique un rango de fechas, no utilice las comillas.

TEMAS RELACIONADOS

- [“Formatos de fecha y hora UTC” en la página 36](#)

Ejemplo de solicitud de API

Solicitud con el parámetro `q`

La siguiente solicitud de ejemplo busca la entidad de negocio de persona para los registros con el nombre STEVE.

```
GET http://localhost:8080/cmxcs/localhost-orcl-DS_UI1/Person?q=STEVE
```

Solicitud con el parámetro `fq`

La siguiente solicitud de ejemplo busca la entidad de negocio de persona para los registros con el nombre para mostrar STEVE. El campo `displayName` es un campo indexado.

```
GET http://localhost:8080/cmxcs/localhost-orcl-DS_UI1/Person?fq=displayName=STEVE
```

Solicitud con la opción de ordenación

En la siguiente solicitud de ejemplo se busca la entidad de negocio de persona para los registros con el nombre para mostrar STEVE y los resultados se ordenan por ciudad en orden ascendente:

```
GET http://localhost:8080/cmxcs/localhost-orcl-DS_UI1/Person?fq=displayName=STEVE&order=BillAddresses.Address.cityName
```

Solicitud con el parámetro fq y el operador lógico AND

La siguiente solicitud de muestra busca la entidad de negocio de persona para los registros con el nombre para mostrar STEVE y el ID fiscal DM106:

```
GET http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/Person?fq=displayName=STEVE AND taxId=DM106
```

Solicitud con una faceta

La siguiente solicitud de muestra busca la entidad de negocio de persona para los registros con el nombre para mostrar STEVE y acota los resultados agrupándolos en departamentos:

```
GET http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/Person?fq=displayName=STEVE&facets=department
```

Solicitud con un filtro (filtro exacto)

La siguiente solicitud de muestra busca la entidad de negocio de persona para los registros con el nombre para mostrar STEVE y filtra por la ciudad y el país especificados:

```
GET http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/Person?fq=displayName=STEVE&filters=cityName=Canberra AND country=Australia
```

Solicitud con un rango de filtros

La siguiente solicitud de muestra busca la entidad de negocio de persona para los registros con el nombre para mostrar STEVE y filtra por el grupo de edades comprendidas entre 35 y 45:

```
GET http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/Person?fq=displayName=STEVE&filters=age=[35,45] AND cityName=Canberra
```

Ejemplo de respuesta de API

La siguiente respuesta de ejemplo muestra el resultado de una búsqueda del nombre STEVE:

```
{
  "firstRecord": 1,
  "recordCount": 2,
  "pageSize": 10,
  "item": [
    {
      "Person": {
        "link": [
          {
            "href": "http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/Person/1443",
            "rel": "self"
          },
          {
            "href": "http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/Person/1443?depth=2",
            "rel": "children"
          }
        ]
      },
      "rowidObject": "1443",
      "label": "CRAIG, STEVE",
      "partyType": "Person",
      "lastName": "CRAIG",
      "firstName": "STEVE",
      "taxID": "stevecraig",
      "displayName": "STEVE CRAIG"
    },
    {
      "Person": {
        "link": [
          {
            "href": "http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/Person/1443",
            "rel": "self"
          }
        ]
      }
    }
  ]
}
```

```

    "Person/285",
    {
      "rel": "self"
    },
    {
      "href": "http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/
    Person/285?depth=2",
      "rel": "children"
    }
  ],
  "rowidObject": "285",
  "label": "PEARSON, STEVE",
  "partyType": "Person",
  "lastName": "PEARSON",
  "firstName": "STEVE",
  "displayName": "STEVE PEARSON"
}
]
}
}

```

Proveedor de sugerencias

La API de REST del Proveedor de sugerencias devuelve una lista de los términos relacionados para una cadena de búsqueda basada en los datos presentes en su base de datos. Utilice la API para aceptar los caracteres que escribe en un campo de interfaz de usuario y devolver sugerencias para autocompletar lo que está escribiendo. Puede buscar y seleccionar la cadena desde la lista de sugerencias. Utilice la API del Proveedor de sugerencias para los campos que admiten búsquedas.

La API utiliza el método GET.

Nota: Para usar la API para que proporcione una lista de sugerencias para autocompletar en un campo que admite búsquedas, establezca la propiedad `suggester` del campo como verdadera y vuelva a indexar los datos.

URL de solicitud

La URL de REST del Proveedor de sugerencias tiene el siguiente formato:

```
http://<host>:<port>/<context>/<database ID>/<business entity>?suggest=<string>
```

Haga la siguiente solicitud HTTP GET a la URL:

```
GET http://<host>:<port>/<context>/<database ID>/<business entity>?suggest=<string>
```

Parámetros de consulta

En la tabla siguiente se muestran los parámetros de la consulta:

Parámetro	Descripción
suggest	Obligatorio. Especifica la cadena para la que desea recibir sugerencias.
recordsToReturn	Especifica el número de filas que se van a devolver.
buildIndex	Opcional. Indica si se va a crear el índice. Cuando reinicie el sistema, ajuste este parámetro como verdadero para crear explícitamente el índice de la colección. Este parámetro puede quedar en desuso en versiones futuras.

Ejemplo de solicitud de API

El siguiente ejemplo devuelve una lista de las sugerencias que puede usar en una interfaz de usuario:

```
GET http://localhost:8080/cmx/cs/localhost-infa-DS_UI1/Person.json?suggest=Abhinav
```

Ejemplo de respuesta de API

La siguiente respuesta de ejemplo muestra la lista de sugerencias:

```
{
  term: [2]
  "abhinav goel"
  "abhinav gupta"
}
```

SearchQuery

La API de REST SearchQuery busca los registros que son una coincidencia exacta de los valores de campo especificados en una consulta. La API SearchQuery se puede utilizar para recuperar todos los registros de una entidad de negocio específica o para recuperar registros en función de los valores de campo especificados. A diferencia de la API de búsqueda de registros que busca los valores especificados en todos los campos donde se pueden realizar búsquedas, la API SearchQuery busca los registros especificados dentro de determinados campos.

Los resultados de la consulta se pueden filtrar para mostrar determinados valores en un registro de entidad de negocio raíz y en los registros secundarios. Se pueden utilizar los operadores AND, IN y RANGE en la consulta.

Para incluir determinados campos en los resultados de la consulta, especifique los campos o una vista de entidad de negocio que incluya los campos. Puede ordenar los resultados de la consulta para ver los registros en orden ascendente o descendente.

La API usa el método GET.

URL de solicitud

El contexto de la URL de SearchQuery es `cmx/cs`. En un entorno de MDM alojado, incluya el nombre de inquilino en el contexto, como `<nombre de inquilino>/cmx/cs`.

La URL de SearchQuery tiene los siguientes formatos:

URL para devolver todos los registros de un tipo de entidad de negocio específico

Utilice la siguiente URL para buscar todos los registros del tipo de entidad de negocio especificado:

```
http://<host>:<port>/<context>/<database ID>/<business entity>?action=query
```

Haga la siguiente solicitud HTTP GET a la URL de SearchQuery:

```
GET http://<host>:<port>/<context>/<database ID>/<business entity>?action=query
```

URL para devolver todos los detalles de los registros que coinciden con los valores de campo especificados

Utilice la siguiente URL para buscar los registros que coinciden con los valores de campo que especifique:

```
http://<host>:<port>/<context>/<database ID>/<business entity>?
action=query&filter=<business entity field name 1>='<business entity field value 1>'
AND <business entity field name 2>='<business entity field value 2>'...AND <business
entity field name n>='<business entity field value n>'
```

URL para devolver detalles concretos de los registros que coinciden con los valores de campo especificados

Utilice la siguiente URL para buscar los registros y mostrar los campos de registro específicos en los resultados de búsqueda:

```
http://<host>:<port>/<context>/<database ID>/<business entity>?
action=query&filter=<business entity field name 1>=<business entity field value 1>'
AND <business entity field name 2>=<business entity field value 2>'...AND <business
entity field name n>=<business entity field value n>'&outputView=<business entity
view>
```

Parámetros de consulta

Defina la consulta como una lista de pares de campo y valor.

En la siguiente tabla se describen los parámetros de consulta que puede usar en la URL:

Parámetro	Descripción
action	<p>Obligatorio. Devuelve todos los registros del tipo de entidad de negocio especificado en los resultados de la consulta. Establézcalo en <code>query</code> y use el parámetro con el parámetro <code>filter</code>. Cuando se utiliza sin el parámetro <code>filter</code>, la consulta busca todos los registros del tipo de entidad de negocio especificado.</p> <p>Por ejemplo, utilice la siguiente consulta para buscar todos los registros de entidad de negocio de persona:</p> <pre>Person?action=query</pre>
filter	<p>Obligatorio. Especifica una lista de pares de campo y valor separados por operadores. Los operadores válidos son AND, IN y Range.</p> <p>Por ejemplo, utilice la siguiente consulta para buscar los registros de persona que tengan el nombre STEVE y el apellido SMITH:</p> <pre>Person?action=query&filter=firstName='STEVE' AND lastName='SMITH'</pre>
depth	<p>Especifica el número de niveles de registros secundarios que se van a devolver. Por ejemplo, puede especificar los siguientes niveles:</p> <ul style="list-style-type: none">- 1. Devuelve el registro raíz.- 2. Devuelve el registro raíz y sus registros secundarios directos.- 3. Devuelve el registro raíz, los registros secundarios directos y los registros de elementos secundarios de segundo nivel. <p>Por ejemplo, utilice la siguiente consulta para buscar los registros que tengan el nombre STEVE y devolver información sobre el registro raíz y sus registros secundarios directos:</p> <pre>Person?action=query&filter=firstName='STEVE' AND lastName='SMITH'&depth=2</pre>
suppressLinks	<p>Indica si los vínculos entre elementos primarios y secundarios se pueden ver en la respuesta de la API. Establezca el parámetro en <code>true</code> para suprimir todos los vínculos entre elementos primarios y secundarios en la respuesta. El valor predeterminado es <code>false</code>.</p> <p>Por ejemplo, utilice la siguiente consulta para buscar los registros que tengan el nombre STEVE y devolver una respuesta en la que no se vean los vínculos entre elementos principales y secundarios:</p> <pre>Person?action=query&filter=firstName='STEVE'&suppressLinks=true</pre>
readSystemFields	<p>Indica si desea devolver los campos del sistema en el resultado. El valor predeterminado es <code>false</code>.</p>
fields	<p>Especifica los campos que se van a mostrar en los resultados de la consulta.</p>

Parámetro	Descripción
outputView	Especifica la vista de entidad de negocio que quiere utilizar para mostrar los resultados de la consulta. Cuando configure la vista de entidad de negocio para los resultados de la consulta, incluya los campos que quiere que aparezcan en los resultados de la consulta.
Order	Especifica el orden de los resultados de la consulta. Utilice el carácter más (+) como prefijo para especificar el orden ascendente y el carácter menos (-) como prefijo para especificar el orden descendente. De forma predeterminada, el resultado de la consulta se mostrará en orden ascendente. Si especifica más de un parámetro, el conjunto de resultados se ordena por el primer parámetro de la lista, seguido del segundo.
resolveLookup	Recupera todo el campo de búsqueda de una entidad de negocio específica. Establezca el parámetro en true para cargar el campo de búsqueda e incluirlo en la respuesta. El valor predeterminado es false. Por ejemplo, el campo addressType es un campo de búsqueda en el nivel secundario de la entidad de negocio de persona. Cuando el parámetro resolveLookup se establece en false, es posible que reciba la siguiente respuesta de la API REST: <pre> { "label": "LU Address Type", "addressType": "BILL" } </pre> Cuando el parámetro resolveLookups se establece en true, la respuesta de la API REST incluye detalles adicionales y es posible que reciba la siguiente respuesta de la API REST: <pre> { "label": "LU Address Type", "addressType": "BILL", "addressTypeDisp": "BILLING" } </pre>

Operadores

Se pueden utilizar los siguientes parámetros dentro del parámetro de filtro:

AND

Busca los registros que tienen todos los valores de campo enumerados en el parámetro de filtro.

Por ejemplo, utilice la siguiente consulta para buscar los registros que tengan el nombre STEVE y el apellido SMITH:

```
Person?action=query&filter=firstName='STEVE' AND lastName='SMITH'
```

IN

Busca los registros que tienen alguno de los valores enumerados en el parámetro de filtro.

Por ejemplo, utilice la siguiente consulta para buscar los registros que tengan el nombre STEVE o JOHN:

```
Person?action=query&filter=firstName IN [STEVE,JOHN]
```

Range

Busca los registros dentro de un intervalo especificado. Puede especificar un intervalo para los campos de tipo de datos de fecha y numéricos.

Utilice el siguiente formato para el tipo de datos de enteros:

```
<business entity field name>=[fromValue,toValue]
```

El rango oscila de fromValue a toValue. Asegúrese de que fromValue sea inferior a toValue.

Por ejemplo, utilice la siguiente consulta para buscar registros que estén en el grupo de edad de 35 a 45:

```
Person?action=query&filter=firstName IN [STEVE,JOHN] AND age=[35,45]
```

Utilice el siguiente formato para el tipo de datos de fecha:

```
<business entity field name>=[fromDate,toDate]
```

El intervalo oscila de fromDate a toDate.

Por ejemplo, utilice la siguiente consulta para buscar registros cuya fecha de nacimiento esté entre el 12 de junio de 2000 y el 12 de junio de 2015:

```
Person?action=query&filter=birthdate=[2000-06-12T12:30:00Z,2015-06-12T12:30:00Z]
```

!=

Busca registros que no coincidan con un rango o valor de campo especificado.

Por ejemplo, utilice la siguiente consulta para buscar los registros que tengan un nombre que no sea ADAM:

```
Person?action=query&filter=firstName!='ADAM'
```

Por ejemplo, utilice la siguiente consulta para buscar los registros que tengan un nombre que no sea ADAM y la fecha de nacimiento sea antes del 16 de noviembre de 2017 y después del 16 de noviembre de 2020.

```
Person?action=query&filter=firstName!='ADAM' AND birthdate!  
=[2017-11-16T00:00:00,2020-11-16T00:00:00]
```

Comodín

Puede utilizar un operador comodín de asterisco (*) y texto para especificar un patrón de texto en lugar de una cadena de búsqueda completa. El operador comodín de asterisco se usa para aumentar las opciones de encontrar los registros que desea. El operador comodín de asterisco es útil cuando no sabe cuál es el texto exacto o quiere buscar un texto similar.

En la siguiente tabla se muestran cadenas de búsqueda de ejemplo y se explica cómo funcionan:

Cadena de consulta de ejemplo	Comportamiento de la consulta
John*	Consultas para registros que contienen un valor que empieza por John. Por ejemplo, Johnson o Johnny.
Jo*n	Consultas para registros que empiezan por Jo y terminan en n. Por ejemplo, Johansson o Jordan.
*	Devuelve todos los registros.

Ejemplo de solicitudes de API

En este tema se brindan consultas de solicitud de ejemplo.

Con la siguiente solicitud de ejemplo se realizan consultas en la entidad de negocio de persona para registros que tienen el nombre STEVE y el apellido SMITH:

```
GET http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/Person?  
action=query&filter=firstName='STEVE' AND lastName='SMITH'&outputView=PersonAddressView
```

En las siguientes consultas de solicitud de ejemplo, la entidad de negocio de persona para registros con los ROWID originales que corresponden. En este ejemplo, solo se devuelve un registro, según el valor '1' usado como un ROWID_OBJECT de identificador único para esta persona.

```
GET http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/Person?
action=query&filter=origRowidObject='1'&outputView=PersonAddressView
```

La siguiente solicitud de ejemplo consulta a la entidad de negocio de persona respecto de los registros conservados con los ROWID correspondientes. En este ejemplo, solo se devuelve un registro, según el valor '1' usado como un ROWID_OBJECT de identificador único para esta persona.

```
GET http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/Person?
action=query&filter=rowidObject='1'&outputView=PersonAddressView
```

Ejemplo de respuesta de API

En la siguiente respuesta de ejemplo se muestran los resultados de una consulta para los registros de persona que tienen el nombre STEVE y el apellido SMITH:

```
{
  "link": [],
  "firstRecord": 1,
  "pageSize": 10,
  "searchToken": "SVR1.1T8UU",
  "facet": [],
  "item": [
    {
      "Person": {
        "rowidObject": "268",
        "label": "Person: SMITH, STEVE,268",
        "partyType": "Person",
        "lastName": "SMITH",
        "displayName": "STEVE SMITH",
        "firstName": "STEVE"
      }
    },
    {
      "Person": {
        "rowidObject": "448",
        "label": "Person: SMITH, STEVE,448",
        "partyType": "Person",
        "lastName": "SMITH",
        "displayName": "STEVE SMITH",
        "firstName": "STEVE"
      }
    }
  ]
}
```

Exportar los resultados de SearchQuery a un archivo CSV

Para exportar los resultados de una solicitud SearchQuery como un archivo CSV, en la ruta de la URL de solicitud, especifique el nombre de la entidad de negocio como un archivo .CSV. Puede usar todos los parámetros de consulta en la URL de solicitud.

Por ejemplo, con la siguiente URL de solicitud, exporte los resultados de una búsqueda de registros que coinciden con los valores de campo que se especifican:

```
http://<host>:<port>/<context>/<database ID>/<business entity>.CSV?
action=query&filter=<business entity field name 1>='<business entity field value 1>' AND
<business entity field name 2>='<business entity field value 2>'...AND <business entity
field name n>='<business entity field value n>'
```


Ejemplo de solicitud de API

En la siguiente solicitud de ejemplo se buscan los registros que tienen el nombre STEVE y el apellido SMITH, y se devuelven los resultados de la consulta en formato CSV:

```
GET http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/Person.CSV?
action=query&filter=firstName='STEVE' AND
lastName='SMITH'&fields=recordID,firstName,lastName
```

Ejemplo de respuesta de API

En el siguiente ejemplo se muestran los resultados de la consulta en formato CSV para una consulta que tiene el nombre STEVE y el apellido SMITH:

```
recordID,firstName,lastName
00023,Steve,Smith
00048,Steve,Smith
```

SearchMatch

La API de REST SearchMatch busca los registros que son coincidencias parciales. La API identifica los registros coincidentes según los campos de entidad de negocio específicos que se configuran como columnas de coincidencia en MDM Hub. Antes de utilizar la API SearchMatch, utilice la API para enumerar columnas de coincidencia con la intención de identificar las columnas de coincidencia de una entidad de negocio.

Opcionalmente, puede especificar un conjunto de reglas de coincidencia en lugar de columnas de coincidencia. Para utilizar un conjunto de reglas de coincidencia para la consulta, compruebe que la opción Habilitar la búsqueda por reglas está habilitada para el conjunto de reglas de coincidencia. Si desea información sobre cómo configurar columnas de coincidencia y conjuntos de reglas de coincidencia, consulte la *Guía de configuración de Multidomain MDM*.

Se pueden utilizar los operadores AND, IN y RANGE en la consulta.

Para incluir determinados campos en los resultados de la consulta, especifique los campos o una vista de entidad de negocio que incluya los campos. El orden de clasificación se puede especificar para ver los registros en los resultados de la consulta en orden ascendente o descendente.

La API utiliza el método GET para realizar consultas sobre los campos de entidad de negocio y devuelve los registros que son coincidencias parciales junto con sus puntuaciones de coincidencia y las reglas de coincidencia asociadas.

URL de solicitud

El contexto de la URL de SearchMatch es `cmx/cs`. En un entorno de MDM alojado, incluya el nombre de inquilino en el contexto, como `<nombre de inquilino>/cmx/cs`.

La URL de SearchMatch tiene los siguientes formatos:

URL para devolver los registros coincidentes en función de los valores de campos específicos que se configuran como columnas de coincidencia

Utilice la siguiente URL para buscar los registros que coinciden con los valores de campo que especifique:

```
http://<host>:<port>/<context>/<database ID>/<business entity>?
action=match&fuzzyFilter=<business entity field name 1>='<business entity field
value 1>',<business entity field name 2>='<business entity field value
2>',...<business entity field name n>='<business entity field value n>'
```

Haga la siguiente solicitud HTTP GET a la URL de SearchMatch:

```
GET http://<host>:<port>/<context>/<database ID>/<business entity>?
action=match&fuzzyFilter=<business entity field name 1>=<business entity field
value 1>,<business entity field name 2>=<business entity field value
2>,...<business entity field name n>=<business entity field value n>
```

URL para devolver los registros coincidentes en función de un conjunto de reglas de coincidencia

Utilice la siguiente URL para buscar los registros coincidentes en función de un conjunto de reglas de coincidencia que especifique:

```
http://<host>:<port>/<context>/<database ID>/<business entity>?
action=match&fuzzyFilter=<business entity field name 1>=<business entity field
value 1>,<business entity field name 2>=<business entity field value
2>,...<business entity field name n>=<business entity field value
n>&matchRuleSet=<match rule set name>
```

Parámetros de consulta

Utilice el parámetro `fuzzyFilter` para especificar los valores de campo que desea consultar. Utilice el parámetro `fuzzyFilter` con el parámetro `action`.

En la siguiente tabla se describen los parámetros de consulta que puede usar en la URL:

Parámetro	Descripción
action	Obligatorio. Devuelve los registros coincidentes para la entidad de negocio especificada. Establézcalo en <code>match</code> y use el parámetro con el parámetro <code>fuzzyFilter</code> . Por ejemplo, utilice la siguiente consulta para buscar una persona con el nombre STEVE: <code>Person?action=match&fuzzyFilter=STEVE</code>
fuzzyFilter	Obligatorio. Especifica una lista de pares de nombre de campo y valor de campo separados por comas que quiere usar para consultar los registros de un tipo de entidad de negocio específico. Por ejemplo, utilice la siguiente consulta para buscar registros que tengan el nombre STEVE y una dirección de Toronto: <code>Person?action=match&fuzzyFilter=firstName=STEVE,Address.Address.City=TORONTO</code>
matchRuleSet	Especifica un conjunto de reglas de coincidencia en función del cual desea identificar los registros coincidentes. Si no tiene un conjunto de reglas de coincidencia específico, especifique NONE. Se utilizan las reglas de coincidencia de fusión automática y manual.
filter	Especifica los valores de campo que se van a usar para filtrar los resultados de una búsqueda parcial. Por ejemplo, utilice la siguiente consulta para buscar los registros que tengan el nombre STEVE y que vivan en Toronto: <code>Person? action=match&fuzzyFilter=firstName='STEVE',lastName='SMITH'&filter=city=Toronto</code>

Parámetro	Descripción
depth	<p>Especifica el número de niveles de registros secundarios que se van a devolver. Por ejemplo, puede especificar los siguientes niveles:</p> <ul style="list-style-type: none"> - 1. Devuelve el registro raíz. - 2. Devuelve el registro raíz y sus registros secundarios directos. - 3. Devuelve el registro raíz, los registros secundarios directos y los registros de elementos secundarios de segundo nivel. <p>Por ejemplo, utilice la siguiente consulta para buscar los registros que tengan el nombre STEVE y devolver información sobre el registro raíz y sus registros secundarios directos:</p> <pre>Person?action=match&fuzzyFilter=firstName='STEVE'&filter=city=Toronto</pre>
suppressLinks	<p>Indica si los vínculos entre elementos primarios y secundarios se pueden ver en la respuesta de la API. Establezca el parámetro en true para suprimir todos los vínculos entre elementos primarios y secundarios en la respuesta. El valor predeterminado es false.</p> <p>Por ejemplo, utilice la siguiente consulta para buscar los registros que tengan el nombre STEVE y devolver una respuesta en la que no se vean los vínculos entre elementos principales y secundarios:</p> <pre>Person?action=match&fuzzyFilter=firstName='STEVE'&suppressLinks=true</pre>
readSystemFields	Indica si desea devolver los campos del sistema en el resultado. El valor predeterminado es false.
fields	Especifica los campos que se van a mostrar en los resultados de la consulta.
outputView	Especifica la vista de entidad de negocio que quiere utilizar para mostrar los resultados de la consulta. Cuando configure la vista de entidad de negocio para los resultados de la consulta, incluya los campos que quiere que aparezcan en los resultados de la consulta.
resolveLookup	<p>Recupera todo el campo de búsqueda de una entidad de negocio específica. Establezca el parámetro en true para cargar el campo de búsqueda e incluirlo en la respuesta. El valor predeterminado es false. Por ejemplo, el campo addressType es un campo de búsqueda en el nivel secundario de la entidad de negocio de persona.</p> <p>Cuando el parámetro resolveLookup se establece en false, es posible que reciba la siguiente respuesta de la API REST:</p> <pre>{ "label": "LU Address Type", "addressType": "BILL" }</pre> <p>Cuando el parámetro resolveLookups se establece en true, la respuesta de la API REST incluye detalles adicionales y es posible que reciba la siguiente respuesta de la API REST:</p> <pre>{ "label": "LU Address Type", "addressType": "BILL", "addressTypeDisp": "BILLING" }</pre>

Se pueden utilizar los siguientes parámetros dentro del parámetro de filtro:

AND

Busca los registros que tienen todos los valores de campo enumerados en el parámetro de filtro.


```

        "label": "SAMUEL, STEVE",
        "partyType": "Person",
        "lastName": "SAMUEL",
        "displayName": "MR STEVE SAMUEL",
        "statusCd": "A",
        "firstName": "STEVE",
        "genderCd": {
            "genderCode": "M"
        },
        "namePrefixCd": {
            "namePrefixCode": "MR"
        }
    },
    "matchRule": "IDL2|1",
    "matchScore": "93",
    "link": [
        {
            "href": "http://localhost:8080/cmx/cs/localhost-orcl-DS_UI2/meta/matchRule/Person/IDL2|1.json",
            "rel": "matchRule"
        }
    ]
},
{
    "businessEntity": {
        "Person": {
            "rowidObject": "268",
            "label": "SMITH, STEVE",
            "partyType": "Person",
            "lastName": "SMITH",
            "displayName": "STEVE SMITH",
            "firstName": "SAM"
        }
    },
    "matchRule": "IDL2|1",
    "matchScore": "98",
    "link": [
        {
            "href": "http://localhost:8080/cmx/cs/localhost-orcl-DS_UI2/meta/matchRule/Person/IDL2|1.json",
            "rel": "matchRule"
        }
    ]
},
{
    "businessEntity": {
        "Person": {
            "rowidObject": "448",
            "label": "SMITH, STEVEN",
            "partyType": "Person",
            "lastName": "SMITH",
            "displayName": "SAM STEVEN",
            "firstName": "STEVEN"
        }
    },
    "matchRule": "IDL2|1",
    "matchScore": "98",
    "link": [
        {
            "href": "http://localhost:8080/cmx/cs/localhost-orcl-DS_UI2/meta/matchRule/Person/IDL2|1.json",
            "rel": "matchRule"
        }
    ]
}
],
"facet": []
}

```

Exportar los resultados de SearchMatch a un archivo CSV

Para exportar los resultados de una solicitud SearchMatch como un archivo CSV, en la ruta de la URL de solicitud, especifique el nombre de la entidad de negocio como un archivo .CSV. Puede usar todos los parámetros de consulta en la URL de solicitud.

Por ejemplo, con la siguiente URL de solicitud, exporte los resultados de una búsqueda de registros coincidentes en función de un conjunto de reglas de coincidencia:

```
http://<host>:<port>/<context>/<database ID>/<business entity>.CSV?
action=match&fuzzyFilter=<business entity field name 1>=<business entity field value
1>,<business entity field name 2>=<business entity field value 2>',...<business entity
field name n>=<business entity field value n>&matchRuleSet=<match rule set name>
```

Ejemplo de solicitud de API

En la siguiente solicitud de ejemplo se buscan los registros que coinciden con el nombre STEVE y el apellido SMITH, y se devuelven los resultados de la consulta en formato CSV:

```
GET http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/Person.CSV?
action=match&fuzzyFilter=firstName=STEVE,lastName=SMITH&fields=recordID,firstName,lastNam
e
```

Ejemplo de respuesta de API

En el siguiente ejemplo se muestran los resultados de la consulta de los registros que coinciden con el nombre STEVE y el apellido SMITH en formato CSV:

```
recordID,firstName,lastName
00023,Steve,Smith
00035,Steven,Smith
00048,Steve,Smith
00079,Steve,Smithson
```

Obtener metadatos de BPM

La API de REST para la obtención de metadatos de BPM devuelve los tipos de tarea y dos indicadores que especifican si la herramienta del flujo de trabajo de BPM puede realizar el servicio para obtener el linaje de la tarea y los servicios de administración.

La API utiliza el método GET.

URL de solicitud

La URL para la obtención de metadatos de BPM tiene el siguiente formato:

```
http://<host>:<port>/<context>/<database ID>/BPMMetadata
```

Haga la siguiente solicitud HTTP GET a la URL para la obtención de metadatos de BPM:

```
GET http://<host>:<port>/<context>/<database ID>/BPMMetadata
```

Ejemplo de solicitud de API

La siguiente solicitud de ejemplo devuelve información sobre los tipos de tarea y la herramienta de flujo de trabajo de BPM:

```
GET http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/BPMMetadata
```

Ejemplo de respuesta de API

La siguiente respuesta de ejemplo muestra los tipos de tarea y el valor de los dos indicadores para la herramienta del flujo de trabajo de BPM:

```
{
  "parameters": {
    "doesSupportAdministration": true,
    "doesSupportLineage": true,
    "doesSupportAttachments": true,
    "maximumAttachmentFileSizeInMb": 20,
    "taskTypes": {
      "taskTypes": [
        {
          "name": "Merge",
          "label": "Merge"
        },
        {
          "name": "FinalReview",
          "label": "FinalReview"
        },
        {
          "name": "Update",
          "label": "Update"
        },
        {
          "name": "Notification",
          "label": "Notification"
        },
        {
          "name": "ReviewNoApprove",
          "label": "ReviewNoApprove"
        },
        {
          "name": "Unmerge",
          "label": "Unmerge"
        }
      ]
    }
  }
}
```

Lista de tareas

La API para la obtención de una lista de tareas devuelve una lista de tareas del flujo de trabajo. Un flujo de trabajo define las actividades en un proceso de negocio y las rutas de ejecución a través de las actividades. Cada actividad se denomina una tarea.

La API utiliza el método GET para devolver una lista de tareas clasificada y paginada.

URL de solicitud

La URL para la obtención de una lista de tareas tiene el siguiente formato.

```
http://<host>:<port>/<context>/<database ID>/task
```

Haga la siguiente solicitud HTTP GET a la URL para la obtención de una lista de tareas:

```
GET http://<host>:<port>/<context>/<database ID>/task
```

Puede agregar encabezados HTTP a la solicitud.

Parámetros de consulta

Utilice los campos de datos de tarea como parámetros de consulta para filtrar la lista de tareas.

Puede utilizar los siguientes parámetros de consulta:

Parámetro	Descripción
taskType	Conjunto de acciones que puede realizar en un registro. Utilice el atributo name para especificar el tipo de tarea. Para obtener más información sobre tipos de tarea, consulte la <i>Guía de implementación de Multidomain MDM Data Director</i> .
taskId	ID de la tarea.
processId	ID del proceso de flujo de trabajo que contiene la tarea.
owner	Usuario que realiza la tarea.
title	Breve descripción de la tarea.
status	Estado de la tarea en el flujo de trabajo. Utilice uno de los siguientes dos valores: - Abierto: la tarea no se ha iniciado o está en curso. - Cerrado: la tarea ha finalizado o se ha cancelado.
priority	Nivel de importancia de la tarea. Utilice uno de los siguientes valores: alto, normal y bajo.
creator	Usuario que crea la tarea.
createDateBefore y createDateAfter	Intervalo de fechas. Las tareas se pueden filtrar por el campo createDate.
dueDateBefore y dueDateAfter	Intervalo de fechas. Las tareas se pueden filtrar por el campo dueDate.
ámbito	Muestra tareas disponibles según la función del usuario. Use el valor <code>admin</code> para filtrar la lista de tareas según su función. Nota: Los usuarios asignados a la función de administrador de tareas pueden administrar todas las tareas, mientras los usuarios a los que se les asignaron otras funciones de usuario solo pueden administrar tareas disponibles para su función.
q	Especifica el término de búsqueda. La consulta busca ocurrencias del término de búsqueda en tareas relacionadas con los registros de la entidad de negocio. Por ejemplo, la consulta <code>q=KAREN</code> busca tareas con el término de búsqueda KAREN.

Utilice los parámetros de consulta como los pares name-value en la URL de la solicitud.

En el siguiente ejemplo se muestra cómo se pueden utilizar los parámetros de consulta para filtrar tareas:

```
GET http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/task?
scope=admin&recordsToReturn=100&owner=sergey&status=OPEN&q=KAREN
```


TEMAS RELACIONADOS

- [“Formatos de fecha y hora UTC” en la página 36](#)

Parámetros de clasificación

Para determinar el orden de clasificación en la respuesta de la API de REST, utilice el parámetro de clasificación genérico y proporcione una lista de campos de tarea separados por comas. Puede especificar el orden de clasificación para cada campo. Utilice el signo de guión (-) para especificar el orden descendiente. El orden de clasificación predeterminado es el ascendente.

El siguiente ejemplo muestra cómo se clasifican los resultados:

```
GET http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/task?
recordsToReturn=100&owner=sergey&status=OPEN&sort=-priority
```

Ejemplo de solicitud de API

La siguiente solicitud de ejemplo recupera la lista de tareas:

```
GET http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/task
```

La solicitud no contiene un cuerpo.

Ejemplo de respuesta de API

La siguiente respuesta de ejemplo muestra la lista de tareas en formato JSON:

```
{
  "firstRecord": 1,
  "recordCount": 10,
  "pageSize": 10,
  "task": [
    {
      "link": [
        {
          "href": "http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/task/
urn:b4p2:15443",
          "rel": "self"
        }
      ],
      "taskType": {
        "name": "ReviewNoApprove",
        "label": "Review No approve",
        "taskKind": "REVIEW",
        "taskAction": [
          {
            "name": "Escalate",
            "label": "Escalate",
            "nextTaskType": "AVOSBeFinalReview",
            "comment": "AS_REQUIRED",
            "attachment": "NEVER",
            "manualReassign": false,
            "closeTaskView": true,
            "cancelTask": false
          },
          {
            "name": "Reject",
            "label": "Reject",
            "nextTaskType": "AVOSBeUpdate",
            "comment": "MANDATORY",
            "attachment": "MANDATORY",
            "manualReassign": false,
            "closeTaskView": true,
            "cancelTask": false
          }
        ]
      }
    }
  ]
}
```

```

        "name": "Disclaim",
        "label": "Disclaim",
        "nextTaskType": "AVOSBeReviewNoApprove",
        "comment": "AS_REQUIRED",
        "attachment": "NEVER",
        "manualReassign": false,
        "closeTaskView": true,
        "cancelTask": false
    }
},
    "pendingBVT": true,
    "updateType": "PENDING"
},
"taskId": "urn:b4p2:15443",
"title": "Review changes in SMITH,SMITH",
"dueDate": "2015-07-15T21:45:59-07:00",
"status": "OPEN",
"priority": "NORMAL",
"businessEntity": "Person"
},
{
    "link": [
        {
            "href": "http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/task/urn:b4p2:15440",
            "rel": "self"
        }
    ],
    "taskType": {
        "name": "ReviewNoApprove",
        "label": "Review No approve",
        "taskKind": "REVIEW",
        "pendingBVT": true,
        "updateType": "PENDING"
    },
    "taskId": "urn:b4p2:15440",
    "title": "Review changes in SMITH,JOHN",
    "dueDate": "2015-07-15T21:37:50-07:00",
    "status": "OPEN",
    "priority": "NORMAL",
    "businessEntity": "Person"
},
{
    "link": [
        {
            "href": "http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/task/urn:b4p2:15437",
            "rel": "self"
        }
    ],
    "taskType": {
        "name": "ReviewNoApprove",
        "label": "Review No approve",
        "taskKind": "REVIEW",
        "taskAction": [
            {
                "name": "Reject",
                "label": "Reject",
                "nextTaskType": "AVOSBeUpdate",
                "comment": "AS_REQUIRED",
                "attachment": "MANDATORY",
                "manualReassign": false,
                "closeTaskView": true,
                "cancelTask": false
            }
        ]
    },
    "pendingBVT": true,
    "updateType": "PENDING"
},
"taskId": "urn:b4p2:15437",
"title": "Review changes in SMITH,JOHN",

```

```

        "dueDate": "2015-07-15T21:34:32-07:00",
        "status": "OPEN",
        "priority": "NORMAL",
        "businessEntity": "Person"
    },
    {
        "link": [
            {
                "href": "http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/task/
urn:b4p2:14820",
                "rel": "self"
            }
        ],
        "taskType": {
            "name": "ReviewNoApprove",
            "label": "Review No approve",
            "taskKind": "REVIEW",
            "pendingBVT": true,
            "updateType": "PENDING"
        },
        "taskId": "urn:b4p2:14820",
        "title": "Review changes in STAS,STAS",
        "dueDate": "2015-07-14T10:40:51-07:00",
        "status": "OPEN",
        "priority": "NORMAL",
        "businessEntity": "Person"
    },
    {
        "link": [
            {
                "href": "http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/task/
urn:b4p2:14809",
                "rel": "self"
            }
        ],
        "taskType": {
            "name": "ReviewNoApprove",
            "label": "Review No approve",
            "taskKind": "REVIEW",
            "pendingBVT": true,
            "updateType": "PENDING"
        },
        "taskId": "urn:b4p2:14809",
        "title": "Review changes in ,93C8ORSCOFSA687",
        "dueDate": "2015-07-14T08:28:15-07:00",
        "status": "OPEN",
        "priority": "NORMAL",
        "businessEntity": "Person"
    },
    {
        "link": [
            {
                "href": "http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/task/
urn:b4p2:14609",
                "rel": "self"
            }
        ],
        "taskType": {
            "name": "ReviewNoApprove",
            "label": "Review No approve",
            "taskKind": "REVIEW",
            "pendingBVT": true,
            "updateType": "PENDING"
        },
        "taskId": "urn:b4p2:14609",
        "title": "Review changes in A8,A8",
        "dueDate": "2015-07-13T08:40:11-07:00",
        "status": "OPEN",
        "priority": "NORMAL",
        "businessEntity": "Person"
    },
    },

```

```

    {
      "link": [
        {
          "href": "http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/task/
urn:b4p2:14425",
          "rel": "self"
        }
      ],
      "taskType": {
        "name": "ReviewNoApprove",
        "label": "Review No approve",
        "taskKind": "REVIEW",
        "pendingBVT": true,
        "updateType": "PENDING"
      },
      "taskId": "urn:b4p2:14425",
      "title": "Review changes in A7,A7",
      "dueDate": "2015-07-10T14:11:02-07:00",
      "status": "OPEN",
      "priority": "NORMAL",
      "businessEntity": "Person"
    },
    {
      "link": [
        {
          "href": "http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/task/
urn:b4p2:14422",
          "rel": "self"
        }
      ],
      "taskType": {
        "name": "ReviewNoApprove",
        "label": "Review No approve",
        "taskKind": "REVIEW",
        "pendingBVT": true,
        "updateType": "PENDING"
      },
      "taskId": "urn:b4p2:14422",
      "title": "Review changes in A6,A6",
      "dueDate": "2015-07-10T13:54:09-07:00",
      "status": "OPEN",
      "priority": "NORMAL",
      "businessEntity": "Person"
    },
    {
      "link": [
        {
          "href": "http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/task/
urn:b4p2:14415",
          "rel": "self"
        }
      ],
      "taskType": {
        "name": "ReviewNoApprove",
        "label": "Review No approve",
        "taskKind": "REVIEW",
        "pendingBVT": true,
        "updateType": "PENDING"
      },
      "taskId": "urn:b4p2:14415",
      "title": "Review changes in A5,A5",
      "dueDate": "2015-07-10T13:51:12-07:00",
      "status": "OPEN",
      "priority": "NORMAL",
      "businessEntity": "Person"
    },
    {
      "link": [
        {
          "href": "http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/task/
urn:b4p2:14355",

```

```

        "rel": "self"
      }
    ],
    "taskType": {
      "name": "Notification",
      "label": "Notification",
      "taskKind": "REVIEW",
      "pendingBVT": false,
      "updateType": "ACTIVE"
    },
    "taskId": "urn:b4p2:14355",
    "title": "Review changes in A4,A4",
    "dueDate": "2015-07-10T10:31:57-07:00",
    "status": "OPEN",
    "priority": "NORMAL",
    "businessEntity": "Person"
  }
]
}

```

Leer tarea

La API de REST para la lectura de tareas devuelve los detalles de una tarea, como son el tipo de tarea, su prioridad y estado.

La API utiliza el método GET.

URL de solicitud

La dirección URL para la lectura de tareas tiene el siguiente formato:

```
http://<host>:<port>/<context>/<database ID>/task/<taskId>
```

Nota: Utilice la API para la obtención de una lista de tareas para obtener el ID de la tarea.

Haga la siguiente solicitud HTTP GET a la URL para la lectura de tareas:

```
GET http://<host>:<port>/<context>/<database ID>/task/<taskId>
```

Ejemplo de solicitud de API

La siguiente solicitud de ejemplo devuelve los detalles de una tarea:

```
GET http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/task/urn:b4p2:16605
```

Ejemplo de respuesta de API

La siguiente respuesta de ejemplo muestra los detalles de la tarea:

```

{
  "taskType": {
    "name": "ReviewNoApprove",
    "label": "Review No Approve",
    "taskKind": "REVIEW",
    "taskAction": [
      {
        "name": "Escalate",
        "label": "Escalate",
        "nextTaskType": "AVOSBeFinalReview",
        "comment": "AS REQUIRED",
        "attachment": "NEVER",
        "manualReassign": false,
        "closeTaskView": true,
        "cancelTask": false
      }
    ],
  },
  {

```

```

        "name": "Reject",
        "label": "Reject",
        "nextTaskType": "AVOSBeUpdate",
        "comment": "MANDATORY",
        "attachment": "MANDATORY",
        "manualReassign": false,
        "closeTaskView": true,
        "cancelTask": false
    },
    {
        "name": "Disclaim",
        "label": "Disclaim",
        "nextTaskType": "AVOSBeReviewNoApprove",
        "comment": "AS_REQUIRED",
        "attachment": "NEVER",
        "manualReassign": false,
        "closeTaskView": true,
        "cancelTask": false
    }
],
    "pendingBVT": true,
    "updateType": "PENDING"
},
"taskId": "urn:b4p2:16605",
"processId": "16603",
"title": "Review changes in HERNANDEZ,ALEJANDRO",
"dueDate": "2015-07-23T01:18:39.125-07:00",
"status": "OPEN",
"priority": "NORMAL",
"taskRecord": [
    {
        "businessEntity": {
            "key": {
                "rowid": "114",
                "sourceKey": "SYS:114"
            },
            "name": "Person"
        }
    },
    {
        "businessEntity": {
            "key": {
                "rowid": "114",
                "sourceKey": "SYS:114",
                "rowidXref": "4680363"
            },
            "name": "Person.XREF"
        }
    }
],
"creator": "avos",
"createDate": "2015-07-16T01:18:46.148-07:00",
"attachments": [
    {
        "id": "urn.b4p2:22203::file1.txt",
        "name": "file1.txt",
        "contentType": "text/plain",
        "creator": "admin",
        "createDate": "2018-02-26T14:31:05.590-05:00"
    }
],
"businessEntity": "Person",
"interactionId": "72340000003000",
"orsId": "localhost-orcl-DS_UI1"
}

```

Crear tarea

La API de REST para la creación de tareas crea una tarea e inicia un flujo de trabajo.

La API utiliza el método POST para crear una tarea y devolver el ID del proceso de flujo de trabajo que contiene la tarea.

URL de solicitud

La dirección URL para la creación de tareas tiene el siguiente formato:

```
http://<host>:<port>/<context>/<database ID>/task
```

Haga la siguiente solicitud POST HTTP a la URL para la creación de tareas:

```
POST http://<host>:<port>/<context>/<database ID>/task
```

Agregue el encabezado Content-Type para especificar el tipo de medio de los datos que desea enviar con la solicitud:

```
POST http://<host>:<port>/<context>/<database ID>/task
Content-Type: application/<json/xml>
```

Cuerpo de la solicitud

Especifique los atributos de tarea cuando cree la tarea. Utilice el formato JSON o el formato XML para enviar los datos de la tarea en la solicitud.

En la siguiente tabla se describen los parámetros de tarea del cuerpo de la solicitud:

Parámetro	Descripción
taskType	Conjunto de acciones que puede realizar en un registro. Utilice el atributo name para especificar el tipo de tarea. Para obtener más información sobre tipos de tarea, consulte la <i>Guía de implementación de Multidomain MDM Data Director</i> .
owner	El usuario al que el creador asigna la tarea.
title	Breve descripción de la tarea.
comments	Comentarios de la tarea.
adjuntos	Adjuntos de la tarea.
dueDate	La fecha en la que el propietario debe completar la tarea.
status	Estado de la tarea en el flujo de trabajo. Utilice uno de los siguientes dos valores: <ul style="list-style-type: none">- Abierto: la tarea no se ha iniciado o está en curso.- Cerrado: la tarea ha finalizado o se ha cancelado.
priority	Nivel de importancia de la tarea. Utilice uno de los siguientes valores: alto, normal y bajo. El valor predeterminado es Normal.
creator	Usuario que crea la tarea.
createDate	La fecha en la que se crea la tarea.
orsId	El ID del Almacén de referencias operativas (ORS) como está registrado en la herramienta de bases de datos de la Consola del concentrador.

Parámetro	Descripción
processId	El ID de tipo de tarea de ActiveVOS®. Para obtener más información, véase <i>Guía de implementación de Multidomain MDM Data Director</i> .
taskRecord	El registro raíz de objeto de negocio o el registro de referencia cruzada asociado a la tarea. Utilice el ID de fila o el sistema de origen y la clave de fuente para especificar el registro.
businessEntity	Nombre de la entidad de negocio al que pertenece taskRecord.
interactionId	ID de la interacción. Use el ID de interacción para mantener la relación de contexto de tarea entre una tarea y los registros.
groups	Asigne una tarea a todos los usuarios de los grupos de usuarios especificados. Defina los grupos de usuarios en la consola de MDM Hub. Especifique los grupos como una matriz.

En el siguiente código de ejemplo se utiliza el ID de fila (rowId) para especificar el registro de tarea (taskRecord):

```
taskRecord: [{
  businessEntity:{
    name: "Person",
    key:{
      rowid: "233",
    }
  }
}]
```

El cuerpo de la solicitud presenta el siguiente formato:

```
{
  taskType: {name:"name of the task"},
  owner: "user who performs the task",
  title: "title of the task",
  comments: "description of the task",
  attachments: [
    {
      id: "TEMP_SVR1.1VDVS"
    }
  ],
  dueDate: "date to complete the task",
  status: "status of the task",
  priority: "priority of the task",
  creator: "use who creates the task",
  createDate: "date on which the task is created",
  updatedBy: "user who last updated the task",
  lastUpdateDate: "date on which the task was last updated",
  businessEntity: "name of the business entity",
  interactionID: "ID of an interaction",
  groups: ["group name A", "group name B", ...],
  orsId: "database ID",
  processId: "ActiveVOS task type ID",
  taskRecord: [{
    businessEntity:{
      name: "name of the business entity",
      key:{
        rowid: "rowId of the record", //Use the rowId or the source system and
        source key to identify the record.
      }
    }
  ]
}
```


TEMAS RELACIONADOS

- [“Formatos de fecha y hora UTC” en la página 36](#)

Ejemplo de solicitud de API

La siguiente solicitud de ejemplo crea una tarea para un registro raíz:

```
POST http://localhost:8080/cmxcsllocalhost-orcl-DS_UI1/task
{
  taskType: {name:"UpdateWithApprovalWorkflow"},
  taskId: "",
  owner: "manager",
  title: "Smoke test task",
  comments: "Smoke testing",
  dueDate: "2015-06-15T00:00:00",
  status: "OPEN",
  priority: "NORMAL",
  creator: "admin",
  createDate: "2015-06-15T00:00:00",
  updatedBy: "admin",
  lastUpdateDate: "2015-06-15T00:00:00",
  businessEntity: "Person",
  orsId: "localhost-orcl-DS_UI1",
  processId: "IDDUpdateWithApprovalTask",
  taskRecord: [{
    businessEntity:{
      name: "Person",
      key:{
        rowid: "123"
      }
    }
  ]
}
```

Ejemplo de respuesta de API

El ejemplo siguiente muestra la respuesta al crear correctamente una tarea. La API devuelve el ID del proceso de flujo de trabajo que contiene la tarea.

```
{
  "parameters": {
    "processId": "15827"
  }
}
```

Actualizar tarea

La API de REST para la actualización de tarea actualiza una única tarea.

La API utiliza el método PATCH para actualizar algunos campos de las tareas y el método PUT para actualizar la tarea completa. Proporcione el ID de la tarea como el parámetro URL.

URL de solicitud

La dirección URL para la actualización de tareas tiene el siguiente formato:

```
http://<host>:<port>/<context>/<database ID>/task/<taskId>
```

Nota: Utilice la API para la obtención de una lista de tareas para obtener el ID de la tarea.

Haga la siguiente solicitud HTTP PUT a la URL para la actualización de tareas con el fin de actualizar la tarea completada:

```
PUT http://<host>:<port>/<context>/<database ID>/task/<taskId>
```

Haga la siguiente solicitud HTTP PATCH a la URL para la actualización de tareas con el fin de actualizar algunos campos de la tarea:

```
PATCH http://<host>:<port>/<context>/<database ID>/task/<taskId>
```

Agregue el encabezado Content-Type para especificar el tipo de medio de los datos que desea enviar con la solicitud:

```
PUT http://<host>:<port>/<context>/<database ID>/task/<taskId>
Content-Type: application/<json/xml>
```

Cuerpo de la solicitud

Utilice la API para la lectura de tareas para obtener los detalles de la tarea. Especifique los atributos de tarea cuando actualice la tarea. Utilice el formato JSON o el formato XML para enviar los datos que se van a actualizar en la solicitud.

En la siguiente tabla se describen los parámetros de tarea del cuerpo de la solicitud:

Parámetro	Descripción
taskType	Conjunto de acciones que puede realizar en un registro. Utilice el atributo name para especificar el tipo de tarea. Para obtener más información sobre tipos de tarea, consulte <i>la Guía de implementación de Multidomain MDM Data Director</i> .
taskId	ID de la tarea.
owner	Usuario que realiza la tarea.
title	Breve descripción de la tarea.
comments	Comentarios de la tarea.
adjuntos	Adjuntos de la tarea.
dueDate	La fecha en la que el propietario debe completar la tarea.
status	Estado de la tarea en el flujo de trabajo. Utilice uno de los siguientes dos valores: - Abierto: la tarea no se ha iniciado o está en curso. - Cerrado: la tarea ha finalizado o se ha cancelado.
priority	Nivel de importancia de la tarea. Utilice uno de los siguientes valores: alto, normal y bajo. El valor predeterminado es Normal.
creator	Usuario que crea la tarea.
createDate	La fecha en la que se creó la tarea.
updatedBy	Usuario que actualiza la tarea.
lastUpdateDate	Fecha de la última actualización de la tarea.
orsId	El ID de ORS como está registrado en la herramienta de bases de datos de la Consola del concentrador.
processId	ID del proceso de flujo de trabajo que contiene la tarea.

Parámetro	Descripción
taskRecord	El registro raíz o el registro de referencia cruzada asociado a la tarea. Utilice el ID de fila o el sistema de origen y la clave de fuente para especificar el registro.
businessEntity name	Nombre de la entidad de negocio al que pertenece taskRecord.

En el siguiente código de ejemplo se utiliza el ID de fila (rowId) para especificar el registro de tarea (taskRecord):

```
taskRecord: [{
  businessEntity:{
    name: 'Person',
    key:{
      rowid: '233',
      systemName: '',
      sourceKey: ''
    }
  }
}]
```

En una solicitud de parche (PATCH), el cuerpo de la solicitud contiene los campos de tarea que desea cambiar. Es posible cambiar el título de la tarea, su prioridad, la fecha de vencimiento y el propietario.

En una solicitud de colocación (PUT), el cuerpo de la solicitud debe incluir todos los campos de tarea. Utilice el siguiente cuerpo de solicitud para crear una solicitud PUT:

```
{
  taskType: {name:"name of the task"},
  taskId: "ID of the task",
  owner: "user who performs the task",
  title: "title of the task",
  comments: "description of the task",
  attachments: [
    {
      id: "TEMP_SVR1.1VDVS"
    }
  ],
  dueDate: "date to complete the task",
  status: "status of the task",
  priority: "priority of the task",
  creator: "use who creates the task",
  createDate: "date on which the task is created",
  updatedBy: "user who last updated the task",
  lastUpdateDate: "date on which the task was last updated",
  businessEntity: "name of the business entity",
  orsId: "database ID",
  processId: 'ActiveVOS task type ID',
  taskRecord: [{
    businessEntity:{
      name: 'name of the business entity',
      key:{
        rowid: 'rowId of the record',//Use the rowId or the source system and source
        key to identify the record.
        systemName: '',
        sourceKey: ''
      }
    }
  ]
}
```

TEMAS RELACIONADOS

- [“Formatos de fecha y hora UTC” en la página 36](#)

Ejemplo de solicitud de API

La siguiente solicitud PUT de ejemplo actualiza una tarea completa:

```
PUT http://localhost:8080/cmxcsllocalhost-orcl-DS_UI1/task/urn:b4p2:15934
{
  taskType: {name:"UpdateWithApprovalWorkflow"},
  taskId: "urn:b4p2:15934",
  owner: "John",
  title: "Smoke test task - updated",
  comments: "Smoke testing - updated",
  attachments: [
    {
      id: "TEMP_SVR1.1VDVS"
    }
  ],
  dueDate: "2015-08-15T00:00:00",
  status: "OPEN",
  priority: "HIGH",
  creator: "admin",
  createDate: "2015-06-15T00:00:00",
  updatedBy: "admin",
  lastUpdateDate: "2015-06-15T00:00:00",
  businessEntity: "Person",
  orsId: "localhost-orcl-DS_UI1",
  processId: '3719',
  taskRecord: [{
    businessEntity:{
      name: 'Person',
      key:{
        rowid: '123',
        systemName: '',
        sourceKey: ''
      }
    }
  ]
}
```

La siguiente solicitud PATCH de ejemplo actualiza algunos campos de tarea:

```
PATCH http://localhost:8080/cmxcsllocalhost-orcl-DS_UI1/task/urn:b4p2:15934
{
  processId: "3719",
  priority: "HIGH",
  owner: "John"
}
```

Ejemplo de respuesta de API

La API devuelve un código de respuesta de 200 OK al actualizar correctamente una tarea. El cuerpo de la respuesta está vacío.

Tarea completada

La API de REST para tareas completadas cierra un flujo de trabajo de tareas después de que finalicen todas las tareas del flujo de trabajo. Utilice la API para cerrar un flujo de trabajo después de procesar todos los registros relacionados con la tarea. Por ejemplo, cuando elija fusionar candidatos, podrá crear una tarea que inicie el flujo de trabajo de la fusión. La tarea de fusión se completará una vez que haya obtenido una vista

previa de cada candidato y lo haya fusionado o lo haya marcado como no coincidente. Utilice la API para cerrar el flujo de trabajo de la fusión.

La API utiliza el método PUT.

URL de solicitud

La URL para la tarea completada tiene el siguiente formato:

```
http://<host>:<port>/<context>/<database ID>/task/<taskId>?action=complete
```

Haga la siguiente solicitud HTTP PUT a la URL para la tarea completada:

```
PUT http://<host>:<port>/<context>/<database ID>/task/<taskId>?action=complete
```

Agregue el encabezado Content-Type para especificar el tipo de medio de los datos que desea enviar con la solicitud:

```
PUT http://<host>:<port>/<context>/<database ID>/task/<taskId>?action=complete
Content-Type: application/<json/xml>
```

Cuerpo de la solicitud

Envíe los detalles de la tarea en el cuerpo de la solicitud. Utilice la API para la lectura de tareas para obtener los detalles de la tarea.

En la siguiente tabla se describen los parámetros de tarea del cuerpo de la solicitud:

Parámetro	Descripción
taskType	Conjunto de acciones que puede realizar en un registro. Utilice el atributo name para especificar el tipo de tarea. Para obtener más información sobre tipos de tarea, consulte <i>la Guía de implementación de Multidomain MDM Data Director</i> .
taskId	ID de la tarea.
owner	Usuario que realiza la tarea.
title	Breve descripción de la tarea.
comments	Comentarios de la tarea.
dueDate	La fecha en la que el propietario debe completar la tarea.
status	Estado de la tarea en el flujo de trabajo. Utilice uno de los siguientes dos valores: - Abierto: la tarea no se ha iniciado o está en curso. - Cerrado: la tarea ha finalizado o se ha cancelado.
priority	Nivel de importancia de la tarea.
creator	Usuario que crea la tarea.
createDate	La fecha en la que se creó la tarea.
updatedBy	Usuario que actualiza la tarea.
lastUpdateDate	Fecha de la última actualización de la tarea.

Parámetro	Descripción
orsId	El ID de ORS como está registrado en la herramienta de bases de datos de la Consola del concentrador.
processId	ID del proceso de flujo de trabajo que contiene la tarea.
taskRecord	El registro raíz o el registro de referencia cruzada asociado a la tarea. Utilice el ID de fila o el sistema de origen y la clave de fuente para especificar el registro.
businessEntity name	Nombre de la entidad de negocio al que pertenece taskRecord.

En el siguiente código de ejemplo se utiliza el ID de fila (rowId) para especificar el registro de tarea (taskRecord):

```
taskRecord: [{
  businessEntity: {
    name: 'Person',
    key: {
      rowid: '233',
      systemName: '',
      sourceKey: ''
    }
  }
}]
```

TEMAS RELACIONADOS

- [“Formatos de fecha y hora UTC” en la página 36](#)

Ejemplo de solicitud de API

La siguiente solicitud de ejemplo completa el flujo de trabajo de fusión:

```
PUT http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/task/urn:b4p2:20210?
action=complete

{
  "taskType": {"name": "Merge"},
  "taskId": "urn:b4p2:20210",
  "owner": "admin",
  "dueDate": "2015-08-14T17:00:00-07:00",
  "status": "OPEN",
  "priority": "NORMAL",
  "creator": "admin",
  "createDate": "2015-06-15T00:00:00",
  "updatedBy": "admin",
  "lastUpdateDate": "2015-06-15T00:00:00",
  "businessEntity": "Person",
  "orsId": "localhost-orcl-DS_UI1",
  "processId": "20208",
  "taskRecord": [{
    "businessEntity": {
      "name": "Person",
      "key": {
        "rowid": "233",
        "systemName": "",
        "sourceKey": ""
      }
    }
  ]
}
```

Ejemplo de respuesta de API

La API devuelve una respuesta de 200 OK al completar correctamente un flujo de trabajo de tareas. El cuerpo de la respuesta está vacío.

Ejecutar acción de tarea

La API de REST para la ejecución de acciones de tarea restablece una tarea en el flujo de trabajo para procesarla más adelante. Cada tipo de tarea tiene un conjunto de acciones de tarea y un flujo de trabajo que especifica la secuencia de tareas. Al ejecutar una acción de tarea, la tarea se desplaza al siguiente paso del flujo de trabajo. Si a una acción de tarea no le sigue ninguna tarea después, el flujo de trabajo finalizará cuando ejecute la acción de tarea.

La API utiliza el método POST para realizar acciones como, por ejemplo, aprobar, escalar o cancelar una tarea.

URL de solicitud

La siguiente URL especifica el formato de la URL para la ejecución de una acción de tarea:

```
http://<host>:<port>/<context>/<database ID>/task/<taskId>?action=<taskAction>
```

Nota: Utilice la API para la obtención de una lista de tareas para obtener el ID de la tarea.

Haga la siguiente solicitud HTTP POST a la URL para la ejecución de una acción de tarea:

```
POST http://<host>:<port>/<context>/<database ID>/task/<taskId>?action=<taskAction>
```

Si desea editar la tarea antes de ejecutar la acción de la tarea, agregue el encabezado Content-Type para especificar el tipo de medio de los datos de la solicitud:

```
POST http://<host>:<port>/<context>/<database ID>/task/<taskId>?action=<taskAction>  
Content-Type: application/<json/xml>
```

Cuerpo de la solicitud

Proporcione los datos de la tarea en el cuerpo de la solicitud si desea cambiar los detalles de la tarea antes de ejecutar la acción de esta.

En la tabla siguiente se describen los parámetros del cuerpo de la solicitud:

Parámetro	Descripción
taskType	Conjunto de acciones que puede realizar en un registro. Utilice el atributo name para especificar el tipo de tarea. Para obtener más información sobre tipos de tarea, consulte <i>la Guía de implementación de Multidomain MDM Data Director</i> .
taskId	ID de la tarea.
owner	Usuario que realiza la tarea.
title	Breve descripción de la tarea.
comments	Comentarios de la tarea.
adjuntos	Adjuntos de la tarea.
dueDate	La fecha en la que el propietario debe completar la tarea.

Parámetro	Descripción
status	Estado de la tarea en el flujo de trabajo. Utilice uno de los siguientes dos valores: - Abierto: la tarea no se ha iniciado o está en curso. - Cerrado: la tarea ha finalizado o se ha cancelado.
priority	Nivel de importancia de la tarea. Utilice uno de los siguientes valores: alto, normal y bajo.
creator	Usuario que crea la tarea.
createDate	La fecha en la que se creó la tarea.
updatedBy	Usuario que actualiza la tarea.
lastUpdateDate	Fecha de la última actualización de la tarea.
businessEntity	Nombre de la entidad de negocio.
orsId	El ID de ORS como está registrado en la herramienta de bases de datos de la Consola del concentrador.
processId	ID del proceso de flujo de trabajo que contiene la tarea.
taskRecord	El registro raíz o el registro de referencia cruzada asociado a la tarea. Utilice el ID de fila o el sistema de origen y la clave de fuente para especificar el registro.
businessEntity name	Nombre de la entidad de negocio al que pertenece taskRecord.

En el siguiente código de ejemplo se utiliza el ID de fila (rowId) para especificar el registro de tarea (taskRecord):

```
taskRecord: [{
  businessEntity: {
    name: 'Person',
    key: {
      rowid: '233',
      systemName: '',
      sourceKey: ''
    }
  }
}]
```

TEMAS RELACIONADOS

- [“Formatos de fecha y hora UTC” en la página 36](#)

Ejemplo de solicitud de API

La siguiente solicitud de ejemplo cancela una tarea y finaliza el flujo de trabajo:

```
POST http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/task/urn:b4p2:15934?
taskAction=Cancel

{
  taskType: {
    name: "UpdateWithApprovalWorkflow",
    taskAction: [{name: "Cancel"}]
  },
  taskId: "urn:b4p2:15934",
  owner: "manager",
  title: "Smoke test task 222",
```



```

    comments: "Smoke testing",
    attachments: [
      {
        id: "TEMP_SVR1.1VDVS"
      }
    ],
    dueDate: "2015-06-15T00:00:00",
    status: "OPEN",
    priority: "NORMAL",
    creator: "admin",
    createDate: "2015-06-15T00:00:00",
    updatedBy: "admin",
    lastUpdateDate: "2015-06-15T00:00:00",
    businessEntity: "Person",
    orsId: "localhost-orcl-DS_UI1",
    processId: '3685',
    taskRecord: [{
      businessEntity: {
        name: 'Person',
        key: {
          rowid: '123',
          systemName: '',
          sourceKey: ''
        }
      }
    }
  ]
}

```

Ejemplo de respuesta de API

La API devuelve un código de respuesta de 200 OK al ejecutar correctamente una acción de tarea. El cuerpo de la respuesta está vacío.

Lista de usuarios asignables

La API de REST para la obtención de una lista de usuarios asignables devuelve una lista de los usuarios a los que se les pueden asignar las tareas que pertenecen a un tipo de tarea. Utilice la API para obtener los usuarios de destino de una tarea.

La API utiliza el método GET.

URL de solicitud

La URL para la obtención de una lista de usuarios asignables tiene el siguiente formato:

```

http://<host>:<port>/<context>/<database ID>/user?taskType=<task
type>&businessEntity=<business entity name>

```

Haga la siguiente solicitud HTTP GET a la URL para la obtención de una lista de usuarios asignables:

```

GET http://<host>:<port>/<context>/<database ID>/user?taskType=<task
type>&businessEntity=<business entity name>

```

Parámetros de consulta

En la siguiente tabla se enumeran los parámetros obligatorios en la dirección URL:

Parámetro	Descripción
taskType	Conjunto de acciones que puede realizar en un registro. Los tipos de tarea incluyen la actualización con aprobación, la actualización con aprobación opcional, la fusión, la anulación de fusión, la revisión sin aprobación, la revisión final y la actualización del registro rechazado.
businessEntity	Nombre de la entidad de negocio.

Ejemplo de solicitud de API

La siguiente solicitud de ejemplo recupera una lista de los usuarios asignables:

```
GET http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/user.json?
taskType=ReviewNoApprove&businessEntity=Person
```

Ejemplo de respuesta de API

La siguiente respuesta de ejemplo muestra la lista de usuarios asignables para el tipo de tarea ReviewNoApprove:

```
{
  "users": {
    "user": [
      {
        "userName": "admin"
      }
    ]
  },
  "roles": {}
}
```

Reclamación masiva de tareas

La API de REST de reclamación masiva de tareas reclama varias tareas. Úsela para reclamar tareas para las cuales sea un posible propietario.

La API usa el método POST.

Nota: Puede especificar un máximo de 100 tareas en una solicitud.

URL de solicitud

La dirección URL de reclamación masiva de tareas tiene el siguiente formato:

```
http://<host>:<port>/<context>/<database ID>
```

Haga la siguiente solicitud HTTP POST a la URL de reclamación masiva de tareas:

```
POST http://<host>:<port>/<context>/<database ID>
```

Cuerpo de la solicitud

Especifique las tareas que quiera reclamar en el cuerpo de la solicitud.

En la tabla siguiente se describen los parámetros del cuerpo de la solicitud:

Parámetro	Descripción
taskId	ID de la tarea.
optype	Operaciones que quiere realizar en las tareas. Especifique <code>Reclamar</code> para reclamar las tareas.

Ejemplo de solicitud de API

La siguiente solicitud de ejemplo reclama varias tareas:

```
POST http://localhost:8080/cmx/task/operations/mdmdb3-MDM_SAMPLE
{
  "tasks":
  [
    {
      "taskId": "urn:b4p2:2315"
    },
    {
      "taskId": "urn:b4p2:2312"
    },
    {
      "taskId": "urn:b4p2:2309"
    },
    {
      "taskId": "urn:b4p2:2306"
    },
    {
      "taskId": "urn:b4p2:2303"
    }
  ],
  "opType": "Claim"
}
```

Ejemplo de respuesta de API

La siguiente respuesta de ejemplo muestra una lista de tareas reclamadas satisfactoriamente:

```
{
  "tasks":
  [
    {
      "taskId": "urn:b4p2:2315",
      "status": "Completed"
    },
    {
      "taskId": "urn:b4p2:2312",
      "status": "Completed"
    },
    {
      "taskId": "urn:b4p2:2309",
      "status": "Completed"
    },
    {
      "taskId": "urn:b4p2:2306",
      "status": "Completed"
    },
    {
      "taskId": "urn:b4p2:2303",
      "status": "Completed"
    }
  ]
}
```

```
}
}
```

Tareas de liberación masiva

La API de REST de tareas de liberación masiva libera varias tareas en el grupo de tareas no asignadas.

La API usa el método POST.

Nota: Puede especificar un máximo de 100 tareas en una solicitud.

URL de solicitud

La dirección URL de liberación masiva de tareas tiene el siguiente formato:

```
http://<host>:<port>/<context>/<database ID>
```

Haga la siguiente solicitud HTTP POST a la URL de liberación masiva de tareas:

```
POST http://<host>:<port>/<context>/<database ID>
```

Cuerpo de la solicitud

Especifique las tareas que quiera liberar al grupo en el cuerpo de la solicitud.

En la tabla siguiente se describen los parámetros del cuerpo de la solicitud:

Parámetro	Descripción
taskId	ID de la tarea.
comments	Opcional. Comentarios de las tareas.
optype	Operaciones que quiere realizar en las tareas. Especifique <code>Lanzar</code> para lanzar las tareas hacia el grupo de tareas disponibles.

Ejemplo de solicitud de API

La siguiente solicitud de ejemplo lanza varias tareas y agrega el mismo comentario para cada una:

```
POST http://localhost:8080/cmx/task/operations/mdmdb3-MDM_SAMPLE
{
  "tasks":
  [
    {
      "taskId": "urn:b4p2:2318"
    },
    {
      "taskId": "urn:b4p2:2209"
    }
  ],
  "comments": "John is on vacation. Releasing these tasks back to the pool of
available tasks.",
  "opType": "Release"
}
```

La siguiente solicitud de ejemplo lanza varias tareas y agrega un comentario diferente para cada una:

```
POST http://localhost:8080/cmx/task/operations/mdmdb3-MDM_SAMPLE
{
  "tasks":
  [
    {
```

```

        "taskId": "urn:b4p2:2318",
        "comments": "Jane is on vacation."
    },
    {
        "taskId": "urn:b4p2:2209"
        "comments": "Joe is on vacation."
    }
],
"opType": "Release"
}

```

Ejemplo de respuesta de API

La siguiente respuesta de ejemplo muestra una lista de tareas lanzadas satisfactoriamente:

```

{
  "tasks":
  [
    {
      "taskId": "urn:b4p2:2318",
      "status": "Completed"
    },
    {
      "taskId": "urn:b4p2:2209",
      "status": "Completed"
    }
  ]
}

```

Asignación masiva de tareas

La API de REST de asignación masiva de tareas permite asignar o reasignar tareas a los usuarios. Use la API para asignar o reasignar tareas a cualquier usuario que sea un posible propietario de la tarea.

Para determinar los posibles propietarios de tareas, use la AP para enumerar posibles propietarios de tareas. Para obtener más información, consulte [“Enumerar posibles propietarios de tareas” en la página 116](#).

La API usa el método POST.

Nota: Puede especificar un máximo de 100 tareas en una solicitud.

URL de solicitud

La dirección URL de asignación masiva de tareas tiene el siguiente formato:

```
http://<host>:<port>/<context>/<database ID>
```

Haga la siguiente solicitud HTTP POST a la URL de asignación masiva de tareas:

```
POST http://<host>:<port>/<context>/<database ID>
```

Cuerpo de la solicitud

Especifique las tareas y los usuarios que quiera asignar a las tareas en el cuerpo de la solicitud.

En la tabla siguiente se describen los parámetros del cuerpo de la solicitud:

Parámetro	Descripción
taskId	ID de la tarea.
comments	Opcional. Comentarios de las tareas.

Parámetro	Descripción
assignTo	Usuario responsable de la tarea. Nota: Solo puede asignar tareas a usuarios que sean posibles propietario de ellas. Para determinar los posibles propietarios de tareas, use la AP para enumerar posibles propietarios de tareas. Para obtener más información, consulte "Enumerar posibles propietarios de tareas" en la página 116.
optype	Operaciones que quiere realizar en las tareas. Especifique <code>Asignar</code> para asignar las tareas.

Ejemplo de solicitud de API

La siguiente solicitud de ejemplo asigna un usuario a varias tareas y agrega el mismo comentario para cada una:

```
POST http://localhost:8080/cmx/task/operations/mdmdb3-MDM_SAMPLE
{
  "tasks":
  [
    {
      "taskId": "urn:b4p2:5351"
    },
    {
      "taskId": "urn:b4p2:5352"
    }
  ],
  "assignTo": "srmngr1",
  "comments": "Please take a look at these tasks instead.",
  "opType": "Assign"
}
```

La siguiente solicitud de ejemplo asigna las tareas a diferentes usuarios y agrega un comentario distinto a cada una:

```
POST http://localhost:8080/cmx/task/operations/mdmdb3-MDM_SAMPLE
{
  "tasks":
  [
    {
      "taskId": "urn:b4p2:5351",
      "assignTo": "srmngr1",
      "comments": "Please help John take a look at this task."
    },
    {
      "taskId": "urn:b4p2:5352",
      "assignTo": "srmngr2",
      "comments": "Please help Jane take a look at this task."
    }
  ],
  "opType": "Assign"
}
```

Ejemplo de respuesta de API

La siguiente respuesta de ejemplo muestra una lista de tareas asignadas satisfactoriamente:

```
{
  "tasks":
  [
    {
      "taskId": "urn:b4p2:5351",
      "status": "Completed"
    },
    {
      "taskId": "urn:b4p2:5352",
      "status": "Completed"
    }
  ]
}
```

```

    }
  ]
}

```

Edición masiva de tareas

La API de REST de edición masiva de tareas permite editar los detalles de varias tareas.

La API usa el método POST para actualizar algunos campos de tarea.

Nota: Puede especificar un máximo de 100 tareas en una solicitud.

URL de solicitud

La dirección URL de edición masiva de tareas tiene el siguiente formato:

```
http://<host>:<port>/<context>/<database ID>
```

Haga la siguiente solicitud HTTP POST a la URL de edición masiva de tareas:

```
POST http://<host>:<port>/<context>/<database ID>
```

Cuerpo de la solicitud

Especifique las tareas que quiera editar y proporcione los datos de la tarea en el cuerpo de la solicitud.

En la tabla siguiente se describen los parámetros del cuerpo de la solicitud:

Parámetro	Descripción
taskId	ID de la tarea.
dueDate	Opcional. La fecha en la que el propietario debe completar la tarea.
comments	Opcional. Comentarios de la tarea.
título	Opcional. Breve descripción de la tarea.
priority	Opcional. Nivel de importancia de la tarea. Utilice uno de los siguientes valores: alto, normal y bajo.
assignTo	Opcional. Usuario responsable de la tarea.
fileId	Opcional. El ID del archivo que quiera adjuntar a la tarea. Para adjuntar un archivo a una tarea, debe cargar el archivo en un almacenamiento temporario. Para obtener más información, consulte Apéndice B, "Uso de las API de REST para cargar archivos" en la página 342 .
optype	Operaciones que quiere realizar en las tareas. Especifique <code>Editar</code> para editar las tareas.

Ejemplo de solicitud de API

La siguiente solicitud de ejemplo edita varias tareas mediante el uso de detalles de la misma tarea:

```

POST http://localhost:8080/cmx/task/operations/mdmdb3-MDM_SAMPLE
{
  "tasks":
  [
    {
      "taskId": "urn:b4p2:4383"
    },
  ]
}

```

```

        {
            "taskId": "urn:b4p2:4384"
        }
    ],
    "comments": "Please work on this task ASAP. There's more information in the
attachment.",
    "priority": "High",
    "title": "Test Demo Review Person",
    "dueDate": "2019-05-01 00:00:00",
    "fileId": "TEMP_SVR1.1IOPK",
    "opType": "Edit"
}

```

La siguiente solicitud de ejemplo edita varias tareas mediante el uso de detalles de tareas diferentes:

```

POST http://localhost:8080/cmxc/task/operations/mdmdb3-MDM_SAMPLE
{
    "tasks":
    [
        {
            "taskId": "urn:b4p2:4383",
            "comments": "Added a screenshot to help clarify.",
            "priority": "High",
            "title": "Test Demo Review Person",
            "dueDate": "2019-05-01 00:00:00",
            "fileId": "TEMP_SVR1.1IOPK"
        },
        {
            "taskId": "urn:b4p2:4384",
            "priority": "High"
        }
    ],
    "opType": "Edit"
}

```

Ejemplo de respuesta de API

La siguiente respuesta de ejemplo muestra una lista de tareas editadas satisfactoriamente:

```

{
    "tasks":
    [
        {
            "taskId": "urn:b4p2:4383",
            "status": "Completed"
        },
        {
            "taskId": "urn:b4p2:4384",
            "status": "Completed"
        }
    ]
}

```

Acción masiva de tareas

La API de REST de acción masiva de tareas ejecuta acciones de tarea en varias tareas. Al ejecutar una acción de tarea, la tarea se desplaza al siguiente paso del flujo de trabajo. Si a una acción de tarea no le sigue ninguna tarea después, el flujo de trabajo terminará cuando se ejecute la acción de tarea.

La API utiliza el método POST para realizar acciones como, por ejemplo, aprobar, escalar o rechazar una tarea.

Para determinar las acciones de tarea que puede realizar en una tarea, use la API de obtención de acciones de tarea. Para obtener más información, consulte ["Obtener acciones de tarea" en la página 114](#).

Nota: Puede especificar un máximo de 100 tareas en una solicitud.

URL de solicitud

La dirección URL de acción masiva de tareas tiene el siguiente formato:

```
http://<host>:<port>/<context>/<database ID>
```

Haga la siguiente solicitud HTTP POST a la URL de acción masiva de tareas:

```
POST http://<host>:<port>/<context>/<database ID>
```

Cuerpo de la solicitud

Especifique las tareas sobre las que quiera ejecutar una acción de tarea y las acciones que quiera ejecutar.

En la tabla siguiente se describen los parámetros del cuerpo de la solicitud:

Parámetro	Descripción
taskId	ID de la tarea.
actionName	La acción de tarea que quiera ejecutar. Use la API de obtención de acciones de tarea para determinar las acciones que quiera realizar en las tareas.
optype	Operaciones que quiere realizar en las tareas. Especifique <i>Acción</i> para ejecutar una acción de tarea.

Ejemplo de solicitud de API

La siguiente solicitud de ejemplo aprueba varias tareas:

```
POST http://localhost:8080/cmx/task/operations/mdmdb3-MDM_SAMPLE
{
  "tasks":
  [
    {
      "taskId": "urn:b4p2:5351"
    },
    {
      "taskId": "urn:b4p2:5352"
    }
  ],
  "actionName": "Approve",
  "opType": "Action"
}
```

La siguiente solicitud de ejemplo aprueba una tarea y escala otra:

```
POST http://localhost:8080/cmx/task/operations/mdmdb3-MDM_SAMPLE
{
  "tasks":
  [
    {
      "actionName": "Approve",
      "taskId": "urn:b4p2:5353"
    },
    {
      "actionName": "Escalate",
      "taskId": "urn:b4p2:5354"
    }
  ],
  "opType": "Action"
}
```

Ejemplo de respuesta de API

La siguiente respuesta de ejemplo muestra una lista de tareas editadas satisfactoriamente:

```
{
  "tasks":
  [
    {
      "taskId": "urn:b4p2:5351",
      "status": "Completed"
    },
    {
      "taskId": "urn:b4p2:5352",
      "status": "Completed"
    }
  ]
}
```

Obtener acciones de tarea

La API de REST de obtención de acciones de tarea recupera una lista de las acciones de tarea que puede realizar en las tareas.

La API usa el método POST para recuperar una lista de las acciones de tarea.

Nota: Puede especificar un máximo de 100 tareas en una solicitud.

URL de solicitud

La dirección URL de obtención de acciones de tarea tiene el siguiente formato:

```
http://<host>:<port>/<context>/<database ID>
```

Haga la siguiente solicitud HTTP POST a la URL de obtención de acciones de tarea:

```
POST http://<host>:<port>/<context>/<database ID>
```

Cuerpo de la solicitud

Especifique las tareas para las que quiera conocer las acciones de tarea que puede realizar.

En la tabla siguiente se describen los parámetros del cuerpo de la solicitud:

Parámetro	Descripción
taskId	ID de la tarea.
optype	Operaciones que quiere realizar en las tareas. Especifique <code>GetTaskActions</code> para recuperar una lista de acciones de tarea que quiera realizar en la tarea.

Ejemplo de solicitud de API

La siguiente solicitud de ejemplo recupera las acciones de tarea que puede realizar en las tareas:

```
POST http://localhost:8080/cmx/task/operations/mdmdb3-MDM_SAMPLE
{
  "tasks":
  [
    {
      "taskId": "urn:b4p2:5351"
    },
    {
      "taskId": "urn:b4p2:5352"
    }
  ]
}
```

```

    },
    "opType": "GetTaskActions"
}

```

Ejemplo de respuesta de API

La siguiente respuesta de ejemplo muestra una lista de tareas y las acciones de tarea que pueden realizarse en cada una de ellas:

```

{
  "tasks":
  [
    {
      "taskId": "urn:b4p2:5351",
      "status": "Completed",
      "taskActions":
      [
        {
          "name": "Approve",
          "displayName": "Approve",
          "description": null,
          "manualReassign": false,
          "closeTaskView": true,
          "cancelTask": false,
          "nextTaskType": "AVOSBeNotification"
        },
        {
          "name": "Reject",
          "displayName": "Reject",
          "description": null,
          "manualReassign": false,
          "closeTaskView": true,
          "cancelTask": false,
          "nextTaskType": "AVOSBeUpdate"
        },
        {
          "name": "Disclaim",
          "displayName": "Disclaim",
          "description": null,
          "manualReassign": true,
          "closeTaskView": true,
          "cancelTask": false,
          "nextTaskType": "AVOSBeFinalReview"
        }
      ]
    },
    {
      "taskId": "urn:b4p2:5352",
      "status": "Completed",
      "taskActions":
      [
        {
          "name": "Approve",
          "displayName": "Approve",
          "description": null,
          "manualReassign": false,
          "closeTaskView": true,
          "cancelTask": false,
          "nextTaskType": "AVOSBeNotification"
        },
        {
          "name": "Reject",
          "displayName": "Reject",
          "description": null,
          "manualReassign": false,
          "closeTaskView": true,
          "cancelTask": false,
          "nextTaskType": "AVOSBeUpdate"
        }
      ]
    }
  ]
}

```

```

    },
    {
      "name": "Disclaim",
      "displayName": "Disclaim",
      "description": null,
      "manualReassign": true,
      "closeTaskView": true,
      "cancelTask": false,
      "nextTaskType": "AVOSBeFinalReview"
    }
  ]
}

```

Enumerar posibles propietarios de tareas

La API de REST de enumeración de posibles propietarios de tareas recupera los posibles propietarios de varias tareas.

La API usa el método GET.

Nota: Puede especificar un máximo de 100 tareas en una solicitud.

URL de solicitud

La dirección URL de enumeración de posibles propietarios de tarea para las tareas tiene el siguiente formato:

```
http://<host>:<port>/<context>/<database ID>
```

Haga la siguiente solicitud HTTP GET a la URL de enumeración de posibles propietarios de tarea para las tareas:

```
GET http://<host>:<port>/<context>/<database ID>
```

Cuerpo de la solicitud

Especifique las tareas para las que quiera recuperar los posibles propietarios.

En la tabla siguiente se describen los parámetros del cuerpo de la solicitud:

Parámetro	Descripción
taskId	ID de la tarea.
optype	Operaciones que quiere realizar en las tareas. Especifique <code>PotentialOwners</code> para obtener una lista de los posibles propietarios de las tareas:

Ejemplo de solicitud de API

La siguiente respuesta de ejemplo recupera los posibles propietarios para varias tareas:

```

GET http://localhost:8080/cmx/task/operations/mdmdb3-MDM_SAMPLE
{
  "tasks":
  [
    {
      "taskId": "urn:b4p2:4383"
    },
    {
      "taskId": "urn:b4p2:4382"
    }
  ]
}

```

```

    ],
    "opType": "PotentialOwners"
  }
}

```

Ejemplo de respuesta de API

La siguiente respuesta de ejemplo muestra los posibles propietarios para cada tarea:

```

{
  "tasks":
  [
    {
      "taskId": "urn:b4p2:4383",
      "status": "Completed",
      "users":
      [
        [
          "srmngr2",
          "srmngr 2"
        ],
        [
          "srmngr1",
          "senior manager"
        ]
      ]
    },
    {
      "taskId": "urn:b4p2:4382",
      "status": "Completed",
      "users":
      [
        [
          "srmngr2",
          "srmngr 2"
        ],
        [
          "srmngr1",
          "senior manager"
        ]
      ]
    }
  ]
}

```

Enumerar posibles propietarios de una tarea

La API de REST de enumeración de posibles propietarios de una tarea recupera los posibles propietarios de una tarea.

La API usa el método GET.

URL de solicitud

La dirección URL de enumeración de posibles propietarios de tarea para una tarea tiene el siguiente formato:

```
http://<host>:<port>/<context>/list/owners/<database ID>/<taskId>
```

Haga la siguiente solicitud HTTP GET a la URL de enumeración de posibles propietarios de tarea para una tarea:

```
GET http://<host>:<port>/<context>/list/owners/<database ID>
```

Cuerpo de la solicitud

Especifique la tarea para la que quiera recuperar los posibles propietarios.

En la tabla siguiente se describen los parámetros del cuerpo de la solicitud:

Parámetro	Descripción
taskId	ID de la tarea.
optype	Operaciones que quiere realizar en la tarea. Especifique <code>PotentialOwners</code> para obtener una lista de los posibles propietarios de la tarea.

Ejemplo de solicitud de API

La siguiente respuesta de ejemplo recupera los posibles propietarios para una tarea:

```
GET http://localhost:8080/cmx/task/operations/list/owners/mdmdb3-MDM_SAMPLE
{
  "tasks":
  [
    {
      "taskId": "urn:b4p2:4383"
    },
    {
      "taskId": "urn:b4p2:4382"
    }
  ],
  "opType": "PotentialOwners"
}
```

Ejemplo de respuesta de API

La siguiente respuesta de ejemplo muestra los posibles propietarios para una tarea:

```
{
  "tasks":
  [
    {
      "taskId": "urn:b4p2:4383",
      "status": "Completed",
      "users":
      [
        [
          "srmngr2",
          "srmngr 2"
        ],
        [
          "srmngr1",
          "senior manager"
        ]
      ]
    },
    {
      "taskId": "urn:b4p2:4382",
      "status": "Completed",
      "users":
      [
        [
          "srmngr2",
          "srmngr 2"
        ]
      ],
    }
  ]
}
```

```

    [
      "srmngr1",
      "senior manager"
    ]
  ]
}

```

Enumerar metadatos de archivo

La API de REST de enumeración de metadatos de archivo devuelve una lista de los metadatos de archivo en un almacenamiento.

Utilice la API de REST de enumeración de metadatos de archivo con un almacenamiento del tipo BPM o TEMP.

La API usa el método GET.

URL de solicitud

La dirección URL de enumeración de metadatos de archivo tiene el siguiente formato:

```
http://<host>:<port>/<context>/<database ID>/<storage>
```

Haga la siguiente solicitud HTTP GET a la dirección URL de enumeración de metadatos de archivo:

```
GET http://<host>:<port>/<context>/<database ID>/<storage>
```

Ejemplo de solicitud de API

La siguiente solicitud de ejemplo recupera una lista de los metadatos de archivo en un almacenamiento TEMP:

```
GET http://localhost:8080/cmx/file/localhost-orcl-MDM_SAMPLE/TEMP
```

Ejemplo de respuesta de API

La siguiente respuesta de ejemplo muestra una lista de los metadatos de archivo:

```

{
  files: [
    {
      "fileId": "TEMP_SVR1.1VDVS",
      "fileName": "file1.txt",
      "fileType": "text",
      "fileContentType": "text/plain",
    },
    {
      "fileId": "TEMP_SVR1.2ESDS",
      "fileName": "image1.png",
      "fileType": "image",
      "fileContentType": "image/png",
    },
    ...
  ]
}

```

Crear metadatos de archivo

La API de REST de creación de metadatos de archivo crea metadatos para un archivo y devuelve un ID de archivo para el archivo.

El ID de archivo se puede usar para cargar el archivo, adjuntarlo, actualizarlo, descargarlo y eliminarlo.

Utilice la API de REST de creación de metadatos de archivo con un almacenamiento del tipo DB o TEMP.

La API usa el método POST.

URL de solicitud

La dirección URL de creación de metadatos de archivo tiene el siguiente formato:

```
http://<host>:<port>/<context>/<database ID>/<storage>
```

Haga la siguiente solicitud HTTP POST a la dirección URL de creación de metadatos de archivo:

```
POST http://<host>:<port>/<context>/<database ID>/<storage>
```

Cuerpo de la solicitud

Especifique los metadatos del archivo.

En la tabla siguiente se describen los parámetros de los metadatos de archivo del cuerpo de la solicitud:

Parámetro	Descripción
fileName	Nombre del archivo. Por ejemplo, <code>file.txt</code> .
fileType	Categoría del tipo de archivo. Por ejemplo, <code>text</code> o <code>image</code> .
fileContentType	Tipo de contenido del archivo. El tipo de contenido consta de un tipo y un subtipo separados por <code>/</code> . Por ejemplo, <code>image/png</code> .

Nota: Los parámetros que se requieren para la solicitud de la API de REST de creación de metadatos de archivo son específicos del almacenamiento.

Ejemplo de solicitud de API

La siguiente solicitud de ejemplo crea los metadatos de un archivo en un almacenamiento TEMP:

```
POST http://localhost:8080/cmx/file/localhost-orcl-MDM_SAMPLE/TEMP

{
  "fileName": "file1.txt",
  "fileType": "text",
  "fileContentType": "text/plain"
}
```

Ejemplo de respuesta de API

El ejemplo siguiente muestra la respuesta al crear correctamente los metadatos de un archivo en un almacenamiento TEMP. La API devuelve el ID de archivo para el archivo.

```
TEMP_SVR1.1VDVS
```

Nota: El formato del ID de archivo es `<storage type>_<uniqueID>`.

Obtener metadatos de archivo

La API de REST de obtención de metadatos de archivo devuelve los metadatos de un archivo que está asociado a un ID de archivo.

Utilice la API de REST de obtención de metadatos de archivo con un almacenamiento del tipo BPM, BUNDLE, DB o TEMP.

La API usa el método GET.

URL de solicitud

La dirección URL de obtención de metadatos de archivo tiene el siguiente formato:

```
http://<host>:<port>/<context>/<database ID>/<storage>/<fileId>
```

Haga la siguiente solicitud HTTP GET a la dirección URL de obtención de metadatos de archivo:

```
GET http://<host>:<port>/<context>/<database ID>/<storage>/<fileId>
```

Ejemplo de solicitud de API

La siguiente solicitud de ejemplo devuelve los metadatos de un archivo con el ID de archivo TEMP_SVR1.1VDVS en el almacenamiento TEMP:

```
GET http://localhost:8080/cmx/file/localhost-orcl-MDM_SAMPLE/TEMP/TEMP_SVR1.1VDVS
```

La siguiente solicitud de ejemplo devuelve los metadatos del archivo de paquete de recursos con el ID de archivo besMetadata en el almacenamiento BUNDLE:

```
GET http://localhost:8080/cmx/file/localhost-orcl-MDM_SAMPLE/BUNDLE/besMetadata
```

Ejemplo de respuesta de API

La siguiente respuesta de ejemplo muestra los metadatos de un archivo con el ID de archivo TEMP_SVR1.1VDVS en un almacenamiento TEMP:

```
{
  "fileName": "file1.txt",
  "fileType": "text",
  "fileContentType": "text/plain"
}
```

La siguiente respuesta de ejemplo muestra los metadatos del archivo de paquete de recursos con el ID de archivo besMetadata en un almacenamiento BUNDLE:

```
{
  "fileName": "besMetadata.zip",
  "fileType": "BES Metadata Bundle",
  "fileContentType": "application/zip",
  "digest": "a08c5d97da7e6a780ed7c427ff14a8d2d396438cd65b654ad67424e226f64a41"
}
```

Actualizar metadatos de archivo

La API de REST de actualización de metadatos de archivo actualiza los metadatos de un archivo que está asociado a un ID de archivo.

Utilice la API de REST de actualización de metadatos de archivo con un almacenamiento del tipo DB o TEMP.

La API utiliza el método PUT.

URL de solicitud

La dirección URL de actualización de metadatos de archivo tiene el siguiente formato:

```
http://<host>:<port>/<context>/<database ID>/<storage>/<fileId>
```

Haga la siguiente solicitud HTTP PUT a la dirección URL de actualización de metadatos de archivo:

```
PUT
http://<host>:<port>/<context>/<database ID>/<storage>/<fileId>
```

Ejemplo de solicitud de API

La siguiente solicitud de ejemplo actualiza los metadatos de un archivo con el ID de archivo TEMP_SVR1.1VDVS en un almacenamiento TEMP:

```
PUT http://localhost:8080/cmx/file/localhost-orcl-MDM_SAMPLE/TEMP/TEMP_SVR1.1VDVS
{
  "fileName": "file2.txt",
  "fileType": "text",
  "fileContentType": "text/plain"
}
```

La siguiente solicitud de ejemplo actualiza los metadatos de un archivo con el ID de archivo DB_SVR1.OJU1 en un almacenamiento DB:

```
PUT http://localhost:8080/cmx/file/localhost-orcl-MDM_SAMPLE/DB/DB_SVR1.OJU1
{
  "fileName": "Document_2.pdf",
  "fileType": "pdf",
  "fileContentType": "application/pdf"
}
```

Ejemplo de respuesta de API

La API devuelve un código de respuesta de 200 OK al actualizar correctamente los metadatos de archivo. El cuerpo de la respuesta está vacío.

Cargar contenido de archivo

La API de REST de carga de contenido de archivo carga el contenido de un archivo que está asociado a un ID de archivo.

Utilice la API de REST de carga de contenido de archivo con un almacenamiento del tipo BUNDLE, DB o TEMP.

La API utiliza el método PUT.

URL de solicitud

La dirección URL de carga de contenido de archivo tiene el siguiente formato:

```
http://<host>:<port>/<context>/<database ID>/<storage>/<fileId>/content
```

Haga la siguiente solicitud HTTP PUT a la dirección URL de carga de contenido de archivo:

```
PUT http://<host>:<port>/<context>/<database ID>/<storage>/<fileId>/content
```

Ejemplo de solicitud de API

La siguiente solicitud de ejemplo carga el contenido de un archivo con el ID de archivo `TEMP_SVR1.1VDVS` en un almacenamiento `TEMP`:

```
PUT http://localhost:8080/cmx/file/localhost-orcl-MDM_SAMPLE/TEMP/TEMP_SVR1.1VDVS/content
Test attachment content: file 1
```

La siguiente solicitud de ejemplo carga el contenido de un archivo con el ID de archivo `DB_SVR1.OJU1` en un almacenamiento `DB`:

```
PUT http://localhost:8080/cmx/file/localhost-orcl-MDM_SAMPLE/DB/DB_SVR1.OJU1/content
Content-Type: application/octet-stream
<file object (upload using REST client)>
```

La siguiente solicitud de ejemplo carga el contenido de un archivo de paquete de recursos con el ID de archivo `besMetadata` en un almacenamiento `BUNDLE`:

```
PUT http://localhost:8080/cmx/file/localhost-orcl-MDM_SAMPLE/BUNDLE/besMetadata/content
Content-Type: application/octet-stream
Body: binary stream - zip file with besMetadata bundle
```

Ejemplo de respuesta de API

La API devuelve un código de respuesta de 200 OK al cargar correctamente el contenido de un archivo. El cuerpo de la respuesta está vacío.

Obtener contenido de archivo

La API de REST de obtención de contenido de archivo devuelve el contenido de un archivo que está asociado a un ID de archivo.

Utilice la API de REST de obtención de contenido de archivo con un almacenamiento del tipo `BPM`, `BUNDLE`, `DB` o `TEMP`.

La API usa el método `GET`.

URL de solicitud

La dirección URL de obtención de contenido de archivo tiene el siguiente formato:

```
http://<host>:<port>/<context>/<database ID>/<storage>/<fileId>/content
```

Haga la siguiente solicitud HTTP `GET` a la dirección URL de obtención de contenido de archivo:

```
GET http://<host>:<port>/<context>/<database ID>/<storage>/<fileId>/content
```

Ejemplo de solicitud de API

La siguiente solicitud de ejemplo devuelve el contenido de un archivo con el ID de archivo `urn:b4p2:22203::file1.txt` en un almacenamiento `BPM`:

```
GET http://localhost:8080/cmx/file/localhost-orcl-MDM_SAMPLE/BPM/
urn:b4p2:22203::file1.txt/content
```

Nota: Utilice la API de REST de lectura de tareas para recuperar el ID de archivo de un adjunto de tarea en un almacenamiento `BPM`.

La siguiente solicitud de ejemplo devuelve el contenido de un archivo con el ID de archivo `DB_SVR1.OJU1` en un almacenamiento `DB`:

```
GET http://localhost:8080/cmx/file/localhost-orcl-MDM_SAMPLE/DB/DB_SVR1.OJU1/content
```

Nota: Utilice la API de REST de lectura de registros para recuperar el ID del archivo que se adjunta a un registro.

La siguiente solicitud de ejemplo devuelve el contenido del archivo de paquete de recursos con el ID de archivo `besMetadata` en un almacenamiento BUNDLE:

```
GET http://localhost:8080/cmx/file/localhost-orcl-MDM_SAMPLE/BUNDLE/besMetadata/content
```

Ejemplo de respuesta de API

La siguiente respuesta de ejemplo muestra el contenido de un archivo TXT en un almacenamiento BPM:

```
Test attachment content: file 1
```

La siguiente respuesta de ejemplo muestra el contenido del archivo de paquete de recursos en un almacenamiento BUNDLE:

```
Content-Disposition → attachment; filename=besMetadata.zip  
Content-Type → application/octet-stream  
Transfer-Encoding → chunked
```

Suprimir archivo

La API de REST de eliminación de archivos elimina el archivo que está asociado a un ID de archivo, el cual incluye el contenido y los metadatos del archivo.

Utilice la API de REST de eliminación de archivos con un almacenamiento del tipo BUNDLE, DB o TEMP.

La API usa el método DELETE.

URL de solicitud

La dirección URL de eliminación de archivo tiene el siguiente formato:

```
http://<host>:<port>/<context>/<database ID>/<storage>/<fileId>
```

Haga la siguiente solicitud HTTP DELETE a la dirección URL de eliminación de archivo:

```
DELETE http://<host>:<port>/<context>/<database ID>/<storage>/<fileId>
```

Ejemplo de solicitud de API

La siguiente solicitud de ejemplo elimina el archivo que está asociado al ID de archivo `TEMP_SVR1.1VDVS` en un almacenamiento TEMP, incluidos el contenido y los metadatos de archivo:

```
DELETE http://localhost:8080/cmx/file/localhost-orcl-MDM_SAMPLE/TEMP/TEMP_SVR1.1VDVS
```

La siguiente solicitud de ejemplo elimina el archivo que está asociado al ID de archivo `DB_SVR1.OJU1` en un almacenamiento DB, incluidos el contenido y los metadatos de archivo:

```
DELETE http://localhost:8080/cmx/file/localhost-orcl-MDM_SAMPLE/DB/DB_SVR1.OJU1
```

La siguiente solicitud de ejemplo elimina el archivo de paquete de recursos con el ID de archivo `besMetadata` en un almacenamiento BUNDLE, incluidos el contenido y los metadatos de archivo:

```
DELETE http://localhost:8080/cmx/file/localhost-orcl-MDM_SAMPLE/BUNDLE/besMetadata
```

Ejemplo de respuesta de API

La API devuelve un código de respuesta de 200 OK al eliminar correctamente un archivo. El cuerpo de la respuesta está vacío.

Vista previa de informes reemplazados

La API de REST de vista previa de informes reemplazados devuelve los registros de entidad de negocio que se editan y reemplazan con valores nuevos.

La API usa el método POST.

URL de solicitud

La dirección URL de vista previa de los registros reemplazados tiene el siguiente formato:

```
http://<host>:<port>/find-replace/<orsId>/<entity>/<preview{?
firstRecord,order,recordsToReturn,returnTotal}>
```

Haga la siguiente solicitud HTTP POST a la dirección URL de vista previa de los registros reemplazados:

```
POST http://<host>:<port>/find-replace/<orsId>/<entity>/<preview{?
firstRecord,order,recordsToReturn,returnTotal}>
```

Parámetros

Especifique los parámetros para devolver los registros de entidad de negocio que se editan y reemplazan con valores nuevos.

En la tabla siguiente, se describen los parámetros en el cuerpo de la solicitud.

Parámetro	Tipo	Descripción
entity	Ruta de acceso	El nombre de los registros de la entidad de negocio por devolver.
orsId	Ruta de acceso	ID del almacén de referencia operativa de la base de datos para devolver los registros de la entidad de negocio reemplazada.
firstRecord	Consulta	Opcional. La fila de la lista de registros.
order	Consulta	Opcional. Lista de nombres de campo separados por comas con el prefijo opcional + o -. El prefijo + indica la ordenación ascendente de los resultados y el prefijo - indica la ordenación descendente de los resultados. El valor predeterminado es +. Si especifica más de un parámetro, el conjunto de resultados se ordena por el primer parámetro de la lista, seguido del segundo.
recordsToReturn	Consulta	Opcional. Especifica el número de filas que se van a devolver.
returnTotal	Consulta	Opcional. Si se establece en <code>true</code> , devuelve el número de registros en el resultado. El valor predeterminado es <code>false</code> .

Ejemplo de solicitud de API

La siguiente solicitud de ejemplo devuelve los registros de entidad de negocio que se editan y reemplazan con valores nuevos:

```
POST /cmx/find-replace/localhost-orcl-MDM_SAMPLE/Person/preview?returnTotal=true HTTP/1.1
Content-Type: application/json
Host: localhost:8080
Content-Length: 234
```

```

{
  "scope" : {
    "type" : "query",
    "filter" : "lastName='jones'"
  },
  "findReplace" : {
    "field" : "Emails.electAddrTypeCd",
    "find" : null,
    "replacement" : {
      "electronicType" : "EMAIL"
    }
  }
}

```

Ejemplo de respuesta de API

La siguiente respuesta de ejemplo muestra los registros de entidad de negocio:

```

HTTP/1.1 200 OK
Server: JBoss-EAP/7
Date: Mon, 13 Jan 2020 18:07:02 GMT
X-Powered-By: Undertow/1
Content-Type: application/json;charset=UTF-8
Content-Length: 4891

{
  "link" : [ ],
  "firstRecord" : 1,
  "recordCount" : 6,
  "pageSize" : 10,
  "foundRecords" : 2,
  "item" : [ {
    "origin" : {
      "key" : {
        "rowid" : "401924"
      },
      "name" : "Person",
      "label" : "jones, alice",
      "object" : {
        "key" : {
          "rowid" : "426"
        },
        "name" : "Emails",
        "label" : "Emails"
      }
    },
    "found" : false,
    "record" : {
      "rowidObject" : "426",
      "creator" : "admin",
      "createDate" : "2020-01-13T12:40:17.811-05:00",
      "updatedBy" : "admin",
      "lastUpdateDate" : "2020-01-13T12:40:18.583-05:00",
      "consolidationInd" : 4,
      "lastRowidSystem" : "SYS0",
      "hubStateInd" : 1,
      "label" : "Emails",
      "electronicAddress" : "alice@mail.com",
      "electAddrTypeCd" : {
        "electronicType" : "EMAIL"
      }
    }
  }, {
    "origin" : {
      "key" : {
        "rowid" : "401925"
      },
      "name" : "Person",
      "label" : "jones, bob",

```

```

    "object" : {
      "key" : {
        "rowid" : "427"
      },
      "name" : "Emails",
      "label" : "Emails"
    }
  },
  "found" : false,
  "record" : {
    "rowidObject" : "427",
    "creator" : "admin",
    "createDate" : "2020-01-13T12:40:18.006-05:00",
    "updatedBy" : "admin",
    "lastUpdateDate" : "2020-01-13T12:40:18.655-05:00",
    "consolidationInd" : 4,
    "lastRowidSystem" : "SYS0",
    "hubStateInd" : 1,
    "label" : "Emails",
    "electronicAddress" : "bob@mail.com",
    "electAddrTypeCd" : {
      "electronicType" : "EMAIL"
    }
  }
}, {
  "origin" : {
    "key" : {
      "rowid" : "401926"
    },
    "name" : "Person",
    "label" : "jones, charlie",
    "object" : {
      "key" : {
        "rowid" : "428"
      },
      "name" : "Emails",
      "label" : "Emails"
    }
  },
  "found" : false,
  "record" : {
    "rowidObject" : "428",
    "creator" : "admin",
    "createDate" : "2020-01-13T12:40:18.216-05:00",
    "updatedBy" : "admin",
    "lastUpdateDate" : "2020-01-13T12:40:18.216-05:00",
    "consolidationInd" : 4,
    "lastRowidSystem" : "SYS0",
    "hubStateInd" : 1,
    "label" : "Emails",
    "electronicAddress" : "charlie@mail.com",
    "electAddrTypeCd" : {
      "electronicType" : "EMAIL"
    }
  }
}, {
  "origin" : {
    "key" : {
      "rowid" : "401935"
    },
    "name" : "Person",
    "label" : "jones, alice",
    "object" : {
      "key" : {
        "rowid" : "437"
      },
      "name" : "Emails",
      "label" : "Emails"
    }
  },
  "found" : true,

```

```

"record" : {
  "rowidObject" : "437",
  "creator" : "admin",
  "createDate" : "2020-01-13T13:07:02.187-05:00",
  "updatedBy" : "admin",
  "lastUpdateDate" : "2020-01-13T13:07:02.187-05:00",
  "consolidationInd" : 4,
  "lastRowidSystem" : "SYS0",
  "hubStateInd" : 1,
  "label" : "Emails",
  "electronicAddress" : "alice@mail.com"
}
}, {
"origin" : {
  "key" : {
    "rowid" : "401936"
  },
  "name" : "Person",
  "label" : "jones, bob",
  "object" : {
    "key" : {
      "rowid" : "438"
    },
    "name" : "Emails",
    "label" : "Emails"
  }
},
"found" : true,
"record" : {
  "rowidObject" : "438",
  "creator" : "admin",
  "createDate" : "2020-01-13T13:07:02.389-05:00",
  "updatedBy" : "admin",
  "lastUpdateDate" : "2020-01-13T13:07:02.390-05:00",
  "consolidationInd" : 4,
  "lastRowidSystem" : "SYS0",
  "hubStateInd" : 1,
  "label" : "Emails",
  "electronicAddress" : "bob@mail.com"
}
}, {
"origin" : {
  "key" : {
    "rowid" : "401937"
  },
  "name" : "Person",
  "label" : "jones, charlie",
  "object" : {
    "key" : {
      "rowid" : "439"
    },
    "name" : "Emails",
    "label" : "Emails"
  }
},
"found" : false,
"record" : {
  "rowidObject" : "439",
  "creator" : "admin",
  "createDate" : "2020-01-13T13:07:02.587-05:00",
  "updatedBy" : "admin",
  "lastUpdateDate" : "2020-01-13T13:07:02.587-05:00",
  "consolidationInd" : 4,
  "lastRowidSystem" : "SYS0",
  "hubStateInd" : 1,
  "label" : "Emails",
  "electronicAddress" : "charlie@mail.com",
  "electAddrTypeCd" : {
    "electronicType" : "EMAIL"
  }
}
}
}

```



```
} ]  
}
```

Actualizar Buscar y reemplazar

Las API de REST de actualización de informes reemplazados devuelven registros filtrados para las entidades de negocio que quiera editar y reemplazar con valores nuevos.

La API usa el método POST.

URL de solicitud

La dirección URL de actualización de los registros reemplazados tiene el siguiente formato:

```
http://<host>:<port>/find-replace/<orsId>/<entity>/{?systemName}
```

Haga la siguiente solicitud HTTP POST a la dirección URL de actualización de los registros reemplazados:

```
POST http://<host>:<port>/find-replace/<orsId>/<entity>/{?systemName}
```

Parámetros

Especifique los parámetros para actualizar los registros filtrados para las entidades de negocio.

En la siguiente tabla se describen los parámetros:

Parámetro	Tipo	Descripción
entity	Ruta de acceso	El nombre de los registros de la entidad de negocio por devolver.
orsId	Ruta de acceso	ID del almacén de referencia operativa de la base de datos para devolver los registros de la entidad de negocio reemplazada.
systemName	Consulta	Opcional. Nombre del sistema de origen.
If-Unmodified-Since	Encabezado	Opcional. Permite omitir registros reemplazados previamente.

Ejemplo de solicitud de API

La siguiente solicitud de ejemplo actualiza registros filtrados para las entidades de negocio que quiera editar y reemplazar con valores nuevos:

```
POST /cmx/find-replace/localhost-orcl-MDM_SAMPLE/Person?systemName=Admin HTTP/1.1  
Content-Type: application/json  
Host: localhost:8080  
Content-Length: 234  
  
{  
  "scope" : {  
    "type" : "query",  
    "filter" : "lastName='jones'"  
  },  
  "findReplace" : {  
    "field" : "Emails.selectAddrTypeCd",  
    "find" : null,  
    "replacement" : {  
      "electronicType" : "EMAIL"  
    }  
  }  
}
```

```
}  
}
```

Ejemplo de respuesta de API

La siguiente respuesta de ejemplo muestra los registros filtrados actualizados para las entidades de negocio:

```
HTTP/1.1 200 OK  
Server: JBoss-EAP/7  
X-Powered-By: Undertow/1  
Content-Type: application/json;charset=UTF-8  
Content-Length: 1339
```

```
{  
  "successful" : 2,  
  "failed" : 0,  
  "skipped" : 0,  
  "item" : [ {  
    "object" : {  
      "Person" : {  
        "key" : {  
          "rowid" : "401935"      "  
        },  
        "rowidObject" : "401935"  "  
      },  
      "Emails" : {  
        "link" : [ ],  
        "item" : [ {  
          "key" : {  
            "rowid" : "437",  
            "sourceKey" : "480000041000"  
          },  
          "rowidObject" : "437",  
          "electAddrTypeCd" : {  
            "key" : {  
              "rowid" : "1"      "  
            },  
            "rowidObject" : "1"  "  
          }  
        } ]  
      }  
    }  
  },  
  "parameters" : { }  
}, {  
  "object" : {  
    "Person" : {  
      "key" : {  
        "rowid" : "401936"      "  
      },  
      "rowidObject" : "401936"  "  
    },  
    "Emails" : {  
      "link" : [ ],  
      "item" : [ {  
        "key" : {  
          "rowid" : "438",  
          "sourceKey" : "480000043000"  
        },  
        "rowidObject" : "438",  
        "electAddrTypeCd" : {  
          "key" : {  
            "rowid" : "1"      "  
          },  
          "rowidObject" : "1"  "  
        }  
      } ]  
    }  
  }  
},  
}
```

```

    "parameters" : { }
  } ]
}

```

Importar archivo nuevo

La API de REST de importación de archivos nuevos comienza con un nuevo trabajo de importación de archivo e importa las entidades de negocio desde un archivo.

La API usa el método POST.

URL de solicitud

La dirección URL de importación de archivos nuevos tiene el siguiente formato:

```
http://<host>:<port>/<context>/<orsId>/<job>
```

Use la siguiente solicitud HTTP POST para la dirección URL de importación de archivos nuevos:

```
POST http://<host>:<port>/<context>/<orsId>/<job>
```

Parámetros

Especifique los siguientes parámetros para una nueva importación de archivo en el cuerpo de la solicitud.

En la siguiente tabla se describen los parámetros:

Parámetro	Tipo	Descripción
orsID	Ruta de acceso	ID de la base de datos del almacén de referencia operativa en la que importar el nuevo archivo.
fileID	Consulta	Opcional. ID de archivo del archivo importado. Use esta opción si está actualizando registros existentes.
mappingID	Consulta	Opcional. ID de asignación del archivo importado. Use esta opción si está actualizando un archivo importado previamente.
systemName	Consulta	Opcional. Nombre del usuario del sistema que importa el archivo.

Ejemplo de solicitud de API

La siguiente solicitud de ejemplo inicia un nuevo trabajo de importación de archivo e importa entidades de negocio:

```

POST /cmx/beimport/localhost-orcl-MDM_SAMPLE/job HTTP/1.1
Content-Type: application/json
Host: localhost:8080
Content-Length: 94

{
  "fileId" : "TEMP_SVR1.1G7UW",
  "mappingId" : "SVR1.1G7UX",
  "systemName" : "Admin"
}

```

Ejemplo de respuesta de API

La siguiente respuesta de ejemplo muestra el nuevo archivo importado satisfactoriamente:

```
HTTP/1.1 201 Created
Server: JBoss-EAP/7
X-Powered-By: Undertow/1
Location: http://localhost:8080/cmz/jobcontrol/localhost-orcl-MDM_SAMPLE/group/SVR1.1G7UY
```

Importar archivo coincidente

La API de REST de importación de archivos coincidentes hace coincidir entidades de negocio nuevas con entidades de negocio existentes antes de importar el archivo. Después de que se complete la operación de importación, las entidades de negocio coincidentes se actualizan con los datos nuevos.

Puede importar entidades de negocio duplicadas, entidades de negocio únicas o entidades de negocio únicas y duplicadas.

La API usa el método POST.

URL de solicitud

La dirección URL de importación de archivos con coincidencia tiene el siguiente formato:

```
http://<host>:<port>/<context>/<orsId>/<aftermatch>
```

Haga la siguiente solicitud HTTP POST a la dirección URL de importación de archivos con coincidencia:

```
POST http://<host>:<port>/<context>/<orsId>/<aftermatch>
```

Parámetros

Especifique los parámetros para hacer coincidir entidades de negocio nuevas con negocios existentes antes de importar el archivo.

En la siguiente tabla se describen los parámetros:

Parámetro	Tipo	Descripción
orsId	Ruta de acceso	ID de la base de datos del almacén de referencia operativa para importar el archivo después de la coincidencia con registros existentes.
filter	Consulta	Opcional. Utilice los siguientes valores de filtro: withMatches: introduzca <code>true</code> si los registros coinciden con registros existentes. withoutMatches: introduzca <code>false</code> si los registros no coinciden con registros existentes.
jobGroupControllId	Consulta	Opcional. ID de grupo de trabajos de coincidencia.
systemName	Consulta	Opcional. Nombre del sistema de origen.

Ejemplo de solicitud de API

La siguiente solicitud de ejemplo hace coincidir entidades de negocio nuevas con existentes antes de importar el archivo:

```
POST /cmz/beimport/{ors}/aftermatch
{
```

```

    jobGroupControlId: "...",
    systemName: "SFA",
    filter: {
      withMatches: true,
      withoutMatches: false
    }
  }
}

```

Ejemplo de respuesta de API

La API devuelve un código de respuesta 200 OK después de la coincidencia exitosa de nuevas entidades de negocio con las existentes y de la importación del archivo. El cuerpo de la respuesta está vacío.

Obtener propiedades de archivo

La API de REST de obtención de propiedades de archivo devuelve las propiedades de archivo del archivo importado. Por ejemplo, la API devuelve las propiedades de archivo de separador decimal, calificador de texto y delimitador si las selecciona cuando importa el archivo.

La API usa el método GET.

Nota: La API devuelve las propiedades de archivo existentes solo con delimitadores si existen cinco filas o más.

URL de solicitud

La dirección URL de obtención de propiedades de archivo tiene el siguiente formato:

```
http://<host>:<port>/<context>/<orsId>/<fileId>/<properties{query}>
```

Haga la siguiente solicitud HTTP GET a la dirección URL de obtención de propiedades de archivo:

```
GET http://<host>:<port>/<context>/<orsId>/<fileId>/<properties{query}>
```

Parámetros

Especifique los parámetros para obtener las propiedades de archivo del archivo importado.

En la siguiente tabla se describen los parámetros:

Parámetro	Tipo	Descripción
orsId	Ruta de acceso	ID de la base de datos del almacén de referencia operativa.
fileId	Ruta de acceso	ID de archivo del archivo importado.
suggest	Consulta	Opcional. Si se establece en true, la solicitud devuelve las propiedades del archivo según el archivo importado. De lo contrario, la solicitud devuelve propiedades de archivo sugeridas según los datos en el archivo importado.

Ejemplo de solicitud de API

La siguiente solicitud de ejemplo detecta y devuelve las propiedades de archivo del archivo importado:

```
GET /cmx/file/parser/localhost-orcl-MDM_SAMPLE/TEMP_SVR1.1G7UW/properties?suggest=true
HTTP/1.1
```

```
Content-Type: application/json
Host: localhost:8080
```

Ejemplo de respuesta de API

La siguiente respuesta de ejemplo muestra el encabezado y cuerpo después de devolver las propiedades de archivo del archivo importado:

```
HTTP/1.1 200 OK
Server: JBoss-EAP/7
X-Powered-By: Undertow/1
Content-Type: application/json;charset=UTF-8
Content-Length: 370

{
  "confidence" : "HIGH",
  "type" : "CSV",
  "properties" : {
    "headerRow" : 1,
    "dataStartsAtRow" : 2,
    "regionalSettings" : {
      "locale" : "en-CA",
      "datePattern" : "MM-dd-yyyy",
      "decimalSeparator" : ".",
      "thousandsSeparator" : ","
    },
    "delimiter" : ",",
    "encoding" : "UTF-8",
    "textQualifier" : "\""
  }
}
```

Guardar propiedades de archivo

La API de REST para guardar propiedades de archivo analiza y guarda las propiedades de los archivos importados.

La API utiliza el método PUT.

URL de solicitud

La dirección URL de guardado de propiedades de archivo tiene el siguiente formato:

```
http://<host>:<port>/<context>/<orsId>/<fileId>/<properties>
```

Haga la siguiente solicitud HTTP PUT a la dirección URL de guardado de propiedades de archivo:

```
PUT http://<host>:<port>/<context>/<orsId>/<fileId>/<properties>
```

Parámetros

Especifique los parámetros para guardar las propiedades de archivo del archivo importado.

En la tabla siguiente se describen los parámetros del cuerpo de la solicitud:

Parámetro	Tipo	Descripción
orsId	Ruta de acceso	ID de la base de datos del almacén de referencia operativa.
fileId	Ruta de acceso	ID de archivo del archivo importado.

Parámetro	Tipo	Descripción
headerRow	Consulta	Opcional. Número de encabezado de columna que quiera para una columna.
dataStartAtRow	Consulta	Opcional. Fila del archivo de origen desde el que quiere importar datos.
locale	Consulta	Opcional. Idioma de localización.
dataPattern	Consulta	Opcional. Formato de fecha para los campos de fecha en el campo de importación.
decimalSeparator	Consulta	Opcional. Separador decimal. Introduzca una coma o un punto.
thousandsSeparator	Consulta	Opcional. Separador de miles. Introduzca una coma o un punto.
delimitador	Consulta	Opcional. Carácter que representa la delimitación de los valores de datos en el archivo de importación. Introduzca un delimitador predefinido o defina un delimitador personalizado.
codificación	Consulta	Opcional. Codificación Unicode estándar.
textQualifier	Consulta	Opcional. Símbolos usados en el archivo para encerrar una cadena.

Ejemplo de solicitud de API

La siguiente solicitud de ejemplo analiza y guarda las propiedades del archivo:

```
PUT /cmx/file/parser/localhost-orcl-MDM_SAMPLE/TEMP_SVR1.1G7UW/properties HTTP/1.1
Content-Type: application/json
Host: localhost:8080
Content-Length: 344

{
  "type" : "CSV",
  "properties" : {
    "headerRow" : 1,
    "dataStartsAtRow" : 2,
    "regionalSettings" : {
      "locale" : "en-CA",
      "datePattern" : "MM-dd-yyyy",
      "decimalSeparator" : ".",
      "thousandsSeparator" : ","
    },
    "delimiter" : ",",
    "encoding" : "UTF-8",
    "textQualifier" : "\""
  }
}
```

Ejemplo de respuesta de API

La siguiente respuesta de ejemplo muestra que las propiedades de los archivos se guardaron.

```
HTTP/1.1 200 OK
Server: JBoss-EAP/7
X-Powered-By: Undertow/1
```

Devolver propiedades de archivo

La API de REST de devolución de propiedades de archivo devuelve las propiedades de los archivos persistentes.

La API usa el método GET.

URL de solicitud

La dirección URL de devolución de propiedades de archivo tiene el siguiente formato:

```
http://<host>:<port>/<context>/<orsId>/<fileId>/<properties>
```

Haga la siguiente solicitud HTTP GET a la dirección URL de devolución de propiedades de archivo:

```
GET http://<host>:<port>/<context>/<orsId>/<fileId>/<properties>
```

Parámetros

Especifique los parámetros para devolver las propiedades de los archivos persistentes.

En la siguiente tabla se describen los parámetros:

Parámetro	Tipo	Descripción
orsId	Ruta de acceso	ID del almacén de referencias operativas desde el cual recuperar las propiedades del archivo.
fileId	Ruta de acceso	ID de archivo del archivo importado.
suggest	Consulta	Opcional. Si se establece en true, la solicitud devuelve propiedades de archivos que se persisten. De lo contrario, la solicitud devuelve todas las propiedades de archivo.

Ejemplo de solicitud de API

La siguiente solicitud de ejemplo devuelve las propiedades de los archivos persistentes.

```
GET /cmx/file/parser/localhost-orcl-MDM_SAMPLE/TEMP_SVR1.1G7UW/properties?suggest=true
HTTP/1.1
Content-Type: application/json
Host: localhost:8080
```

Ejemplo de respuesta de API

La siguiente respuesta de ejemplo muestra las propiedades de los archivos persistentes:

```
HTTP/1.1 200 OK
Server: JBoss-EAP/7
X-Powered-By: Undertow/1
Content-Type: application/json;charset=UTF-8
Content-Length: 370

{
  "confidence" : "HIGH",
  "type" : "CSV",
  "properties" : {
    "headerRow" : 1,
    "dataStartsAtRow" : 2,
    "regionalSettings" : {
      "locale" : "en-CA",
      "datePattern" : "MM-dd-yyyy",
```



```

        "decimalSeparator" : ".",
        "thousandsSeparator" : ",",
    },
    "delimiter" : ",",
    "encoding" : "UTF-8",
    "textQualifier" : "\""
}
}

```

Vista previa de archivos analizados

La API de vista previa de archivos analizados analiza las propiedades de archivo y devuelve registros sin formato según las propiedades proporcionadas en el cuerpo de la solicitud. La API analiza archivos Excel y CSV.

La API usa el método POST.

URL de solicitud

La dirección URL de la API de vista previa de archivos analizados tiene el siguiente formato:

```
http://<host>:<port>/<context>/<orsId>/<fileId>/preview{?pageSize}
```

Haga la siguiente solicitud HTTP POST a la URL de la API de vista previa de archivos analizados:

```
POST http://<host>:<port>/<context>/<orsId>/<fileId>/preview{?pageSize}
```

Parámetros

Especifique los parámetros para analizar las propiedades de archivo, y se devuelven registros sin formato según las propiedades proporcionadas en el cuerpo de la solicitud.

En la tabla siguiente se describen los parámetros del cuerpo de la solicitud:

Parámetro	Tipo	Descripción
orsId	Ruta de acceso	ID de la base de datos del almacén de referencia operativa del archivo almacenado en la base de datos.
fileId	Ruta de acceso	ID de archivo del archivo importado.
pagesize	Consulta	Opcional. Número de filas de registro para devolver.

Ejemplo de solicitud de API

La siguiente solicitud de ejemplo analiza propiedades de archivo y devuelve registros sin formato según las propiedades proporcionadas en el cuerpo de la solicitud:

```

POST /cmx/file/parser/localhost-orcl-MDM_SAMPLE/TEMP_SVR1.1G7UW/preview HTTP/1.1
Content-Type: application/json
Host: localhost:8080
Content-Length: 345

{
  "type" : "CSV",
  "properties" : {
    "headerRow" : 1,
    "dataStartsAtRow" : 2,
    "regionalSettings" : {
      "locale" : "en-CA",
      "datePattern" : "d MMM, yyyy",

```

```

        "decimalSeparator" : ".",
        "thousandsSeparator" : ",",
    },
    "delimiter" : ",",
    "encoding" : "UTF-8",
    "textQualifier" : "\""
}
}

```

Ejemplo de respuesta de API

La siguiente respuesta de ejemplo muestra los registros sin formato recuperados en función de las propiedades proporcionadas en el cuerpo de la solicitud:

```

HTTP/1.1 200 OK
Server: JBoss-EAP/7
X-Powered-By: Undertow/1
Content-Type: application/json;charset=UTF-8
Content-Length: 381

{
  "header" : [ "first_name", "last_name", "job_title", "gender", "birth_date" ],
  "data" : [ [ "alice", "smith", "developer", "F", "01-01-1950" ], [ "bob", "smith",
"tester", "M", "02-02-1960" ], [ "charlie", "smith", "manager", "X", "03-03-1970" ],
[ "dave", "smith", "accountant", "M", "ten years ago" ], [ "eve", "smith", "", "F",
"04-04-1990" ] ],
  "numberOfRows" : 5
}

```

Obtener archivos analizados

La API de REST de obtención de archivos analizados analiza las propiedades de archivo y devuelve registros sin formato según las propiedades. La API analiza archivos Excel y CSV.

La API usa el método GET.

URL de solicitud

La dirección URL de obtención de propiedades de archivo analizado tiene el siguiente formato:

```
http://<host>:<port>/<context>/<orsId>/<fileId>/preview{?pageSize}
```

Haga la siguiente solicitud HTTP GET a la dirección URL de obtención de propiedades de archivo analizado:

```
GET http://<host>:<port>/<context>/<orsId>/<fileId>/preview{?pageSize}
```

Parámetros

Especifique los parámetros para analizar las propiedades de archivo y devuelva registros sin formato según las propiedades del archivo.

En la siguiente tabla se describen los parámetros:

Parámetro	Tipo	Descripción
orsId	Ruta de acceso	ID de la base de datos del almacén de referencia operativa.
fileId	Ruta de acceso	ID de archivo del archivo importado.
pagesize	Consulta	Opcional. Número de filas de registro para devolver.

Ejemplo de solicitud de API

La siguiente solicitud de ejemplo analiza archivos y devuelve archivos sin formato según las propiedades asociadas con un archivo:

```
GET /cmx/file/parser/localhost-orcl-MDM_SAMPLE/TEMP_SVR1.1G7UW/preview HTTP/1.1
Content-Type: application/json
Host: localhost:8080
```

Ejemplo de respuesta de API

La siguiente respuesta de ejemplo muestra los registros sin formato analizados en función de las propiedades asociadas con un archivo:

```
HTTP/1.1 200 OK
Server: JBoss-EAP/7
X-Powered-By: Undertow/1
Content-Type: application/json;charset=UTF-8
Content-Length: 381

{
  "header" : [ "first_name", "last_name", "job_title", "gender", "birth_date" ],
  "data" : [ [ "alice", "smith", "developer", "F", "01-01-1950" ], [ "bob", "smith",
"tester", "M", "02-02-1960" ], [ "charlie", "smith", "manager", "X", "03-03-1970" ],
[ "dave", "smith", "accountant", "M", "ten years ago" ], [ "eve", "smith", "", "F",
"04-04-1990" ] ],
  "numberOfRows" : 5
}
```

Errores de análisis al importar archivos

La API de REST de errores de análisis al importar archivos devuelve errores de análisis de la operación de importación de archivos en un formato de archivo CSV.

La API usa el método GET.

URL de solicitud

La dirección URL de errores de análisis al importar archivos tiene el siguiente formato:

```
http://<host>:<port>/<context>/<orsId>/<parse-errors>/{jobGroupControlId}{?entity}
```

Haga la siguiente solicitud HTTP GET a la dirección URL de errores de análisis al importar archivos:

```
GET http://<host>:<port>/<context>/<orsId>/<parse-errors>/{jobGroupControlId}{?entity}
```

Parámetros

Especifique los parámetros para devolver errores de análisis de la operación de importación de archivo en un archivo CSV.

En la tabla siguiente se describen los parámetros del cuerpo de la solicitud:

Parámetro	Tipo	Descripción
orsId	Ruta de acceso	ID de la base de datos del almacén de referencia operativa para devolver errores de análisis desde la operación de importación de archivo.
jobGroupControlId	Consulta	Opcional. ID de grupo de trabajos en lotes.
entity	Consulta	Opcional. Nombre de la entidad de negocio.

Ejemplo de solicitud de API

La siguiente solicitud de ejemplo devuelve los errores de análisis de la operación de importación:

```
GET /cmx/beimport/localhost-orcl-MDM_SAMPLE/parse-errors/SVR1.1G7UY HTTP/1.1
Host: localhost:8080
```

Ejemplo de respuesta de API

La siguiente respuesta de ejemplo muestra los errores de análisis de la operación de importación de archivo:

```
HTTP/1.1 200 OK
Server: JBoss-EAP/7
X-Powered-By: Undertow/1
Content-Type: text/csv
Content-Length: 200

first_name,last_name,job_title,gender,birth_date,ROW,COLUMN,ERROR,ENTITY
dave,smith,accountant,M,ten years ago,3,birth_date,Failed to convert value [ten years ago] for field [birthdate].,PersonView
```

Errores de carga al importar archivos

La API de REST de errores de carga al importar archivos devuelve errores de carga de la operación de importación de archivos en un formato de archivo CSV.

La API usa el método GET.

URL de solicitud

La dirección URL de errores de carga al importar archivos tiene el siguiente formato:

```
http://<host>:<port>/<context>/<orsId>/<load-errors>/<jobGroupControlId>{?entity}
```

Haga la siguiente solicitud HTTP GET a la dirección URL de errores de carga al importar archivos:

```
GET http://<host>:<port>/<context>/<orsId>/<load-errors>/<jobGroupControlId>{?entity}
```

Parámetros

Especifique los parámetros para devolver errores de carga de la operación de importación de archivo en un archivo CSV.

En la tabla siguiente, se describen los parámetros en el cuerpo de la solicitud.

Parámetro	Tipo	Descripción
orsld	Ruta de acceso	ID de la base de datos del almacén de referencia operativa para devolver errores de carga desde la operación de importación de archivo.
jobGroupControllId	Consulta	Opcional. ID de grupo de trabajos en lotes.
entity	Consulta	Opcional. Nombre de la entidad de negocio.

Ejemplo de solicitud de API

La siguiente solicitud de ejemplo devuelve los errores de carga de la operación de importación de archivo:

```
GET /cmx/beimport/localhost-orcl-MDM_SAMPLE/load-errors/SVR1.1G7UY HTTP/1.1
Host: localhost:8080
```

Ejemplo de respuesta de API

La siguiente respuesta de ejemplo muestra los errores de carga de la operación de importación:

```
HTTP/1.1 200 OK
Server: JBoss-EAP/7
X-Powered-By: Undertow/1
Content-Type: text/csv
Content-Length: 257

first_name,last_name,job_title,gender,birth_date,ROW,COLUMN,ERROR,ENTITY
charlie,smith,manager,X,03-03-1970,2,,SIP-50112: Could not run WriteCO business entity
service. A record for [genderCd] could not be found in business entity
[LUGender].,PersonView
```

Vista previa de promoción

La API de REST de vista previa de promoción devuelve una vista previa de un registro resultante si promueve los cambios pendientes.

La API utiliza el método GET. Puede ver qué aspecto tendría un registro si aplica los cambios pendientes en el registro. La respuesta de la API contiene el registro con los valores nuevos y un resumen de cambios con los valores anteriores. La API no devolverá información sobre los datos que elimine. Proporcione el ID de interacción de los cambios pendientes en la URL.

URL de solicitud

La URL para la vista previa de la promoción tiene el siguiente formato:

```
http://<host>:<port>/<context>/<database ID><business entity>/<rowId>?
action=previewPromote&interactionID=<interaction ID>
```

Haga la siguiente solicitud HTTP GET a la URL para la vista previa de la promoción:

```
GET http://<host>:<port>/<context>/<database ID><business entity>/<rowId>?
action=previewPromote&interactionID=<interaction ID>
```

Parámetros de consulta

El ID de interacción de los cambios pendientes es un parámetro obligatorio en la URL.

En la tabla siguiente se muestran los parámetros de la consulta:

Parámetro	Descripción
contentMetadata	Metadatos para la vista previa de fusión. Proporcione una lista de valores separados por comas. Puede usar los siguientes valores: <ul style="list-style-type: none">- BVT. Especifique el valor de rowid del registro que contiene el valor de máxima confianza para su uso en la vista previa de fusión. Devuelve información sobre el registro de referencias cruzadas y el ID de registro original.- MERGE. Especifique los valores de rowid de los registros que se van a fusionar. Devuelve información sobre cómo se han fusionado los registros descendientes.
interactionId	El ID de interacción de los cambios pendientes.
effectiveDate	Opcional. Fecha en la que desea obtener una vista previa de los cambios. Utilice el parámetro para objetos base que tengan habilitada la línea temporal.

TEMAS RELACIONADOS

- [“Formatos de fecha y hora UTC” en la página 36](#)

Ejemplo de solicitud de API

La siguiente solicitud de ejemplo crea una vista previa de un registro raíz en la entidad de negocio de persona:

```
GET http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/Person/233?
action=previewPromote&interactionId=72300000001000
```

Ejemplo de respuesta de API

La siguiente respuesta de ejemplo devuelve una vista previa del registro con los valores nuevos y un resumen de cambios de los valores anteriores:

```
{
  "rowidObject": "233",
  "creator": "admin",
  "createDate": "2008-08-12T02:15:02-07:00",
  "updatedBy": "admin",
  "lastUpdateDate": "2015-07-14T03:42:38.778-07:00",
  "consolidationInd": "1",
  "lastRowidSystem": "SYS0",
  "dirtyIndicator": "0",
  "interactionId": "72300000001000",
  "hubStateInd": "1",
  "label": "LLOYD,BOB",
  "partyType": "Person",
  "lastName": "LLOYD",
  "firstName": "BOB",
  "displayName": "BOB LLOYD",
  "preferredPhone": {
    "rowidObject": "164",
    "$original": {
      "rowidObject": "164"
    }
  },
  "$original": {
    "label": "DUNN,LLOYD",
    "lastName": "DUNN",
  }
}
```

```
        "firstName": "LLOYD",
        "displayName": "LLOYD DUNN"
    }
}
```

Promover

La API de REST de promoción promueve todos los cambios pendientes que se realicen en un registro en función del ID de interacción de la solicitud de cambio.

La API utiliza el método POST. Proporcione el ID de interacción como un parámetro de consulta.

URL de solicitud

La dirección URL para la promoción tiene el siguiente formato:

```
http://<host>:<port>/<context>/<database ID>/<business entity>/<rowId of the root record>?action=promote&interactionId=<interaction ID>
```

Haga la siguiente solicitud HTTP POST a la URL para la promoción:

```
POST http://<host>:<port>/<context>/<database ID>/<business entity>/<rowId of the root record>?action=promote&interactionId=<interaction ID>
```

Parámetro de consulta

El ID de interacción de los cambios pendientes es un parámetro obligatorio. La API utiliza el ID de interacción para buscar todos los registros relacionados con los cambios pendientes.

Ejemplo de solicitud de API

La siguiente solicitud de ejemplo promueve todos los cambios pendientes según el ID de interacción de la solicitud de cambio:

```
POST http://localhost:8080/cmxc/cs/localhost-orcl-DS_UI1/Person/1038246?
action=promote&interactionId=69120000294000
```

Ejemplo de respuesta de API

La siguiente respuesta de ejemplo contiene el ID de fila del registro después de promover los cambios pendientes:

```
{
    Person: {
        rowidObject: "1038246"
    }
}
```

Eliminar pendiente

La API de REST de eliminación de pendientes elimina todos los cambios pendientes que realice en un registro en función del ID de interacción de la solicitud de cambio.

La API utiliza el método DELETE y devuelve el ID de fila del registro.

URL de solicitud

La URL para la eliminación de pendiente tiene el siguiente formato:

```
http://<host>:<port>/<context>/<database ID>/<business entity>/<rowid>?
action=deletePending&interactionId=<interaction ID>
```

Haga la siguiente solicitud DELETE a la URL para la eliminación de pendiente:

```
DELETE http://<host>:<port>/<context>/<database ID>/<business entity>/<rowid>?
action=deletePending&interactionId=<interaction ID>
```

Parámetro de consulta

Proporcione el ID de interacción de los cambios pendientes que desee eliminar.

Ejemplo de solicitud de API

La siguiente solicitud de ejemplo elimina todos los cambios pendientes según el ID de interacción de la solicitud de cambio:

```
DELETE http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/Person/233?
action=deletePending&interactionId=72300000001000
```

Ejemplo de respuesta de API

La siguiente respuesta de ejemplo contiene el ID de fila del registro después de eliminar los cambios pendientes:

```
{
  Person: {
    rowidObject: "233"
  }
}
```

Vista previa de fusión

La API de REST de vista previa de fusión devuelve una vista previa de un registro raíz consolidado si fusiona dos o más registros raíz.

La API utiliza el método POST y acepta una lista de registros raíz y reemplazos a nivel de campo para devolver una vista previa del registro fusionado. El ID de fila del registro de destino es un parámetro obligatorio.

URL de solicitud

La URL para la vista previa de la fusión tiene el siguiente formato:

```
http://<host>:<port>/<context>/<database ID>/<business entity>/<rowid of the record>?
action=previewMerge
```

El siguiente formato de la URL para la vista previa de la fusión especifica el número de niveles secundarios que se van a devolver:

```
http://<host>:<port>/<context>/<database ID>/<business entity>/<rowid of the record>?
action=previewMerge&depth=2
```

Haga la siguiente solicitud HTTP POST a la URL para la vista previa de la fusión:

```
POST http://<host>:<port>/<context>/<database ID>/<business entity>/<rowid of the
record>?action=previewMerge
```

Nota: En el cuerpo de la solicitud, añada la propiedad de las claves y especifique los registros raíz que desea fusionar con el registro de destino.

Para reemplazar las coincidencias para los registros secundarios, añada el parámetro contentMetadata:

```
POST http://<host>:<port>/<context>/<database ID>/<business entity>/<rowid of the record>?action=previewMerge&contentMetadata=MERGE/<json/xml>
```

Nota: En el cuerpo de la solicitud, añada la propiedad overrides y especifique los reemplazos de fusión.

Para especificar el tipo de medio de los datos que desea enviar con la solicitud, añada el encabezado Content-Type:

```
POST http://<host>:<port>/<context>/<database ID>/<business entity>/<rowid of the record>?action=previewMerge
Content-Type: application/<json/xml>
```

Parámetros de consulta

El ID de fila del registro de destino es un parámetro obligatorio.

Puede utilizar los siguientes parámetros de consulta:

Parámetro	Descripción
contentMetadata	Metadatos para la vista previa de fusión. Proporcione una lista de valores separados por comas. Puede usar los siguientes valores: <ul style="list-style-type: none">- BVT. Devuelve información sobre el registro de referencias cruzadas prevalente y el ID de registro original.- MERGE. Devuelve información sobre cómo se han fusionado los registros descendientes.
depth	El número de niveles secundarios que se devolverán.
effectiveDate	Fecha en la que desea generar la vista previa.

TEMAS RELACIONADOS

- [“Formatos de fecha y hora UTC” en la página 36](#)

Cuerpo de la solicitud

Antes de empezar, use la API de lectura de registros coincidentes para determinar qué registros coincidentes puede fusionar con el registro raíz original. Envíe la lista de registros en el cuerpo de la solicitud para la API de vista previa de fusión.

Puede reemplazar los valores de campo en el registro raíz. Por ejemplo, si ninguno de los registros raíz coincidentes contiene la ortografía correcta del nombre, puede especificar el nombre correcto en el cuerpo de la solicitud. Asimismo, puede eliminar los registros coincidentes o especificar otros.

Use las siguientes propiedades en el cuerpo de la solicitud:

Propiedades/elementos	Tipo	Descripción
keys	array	Obligatorio. Una lista ordenada de los registros raíz coincidentes que desea fusionar. Puede identificar los registros por el ID de fila o mediante una combinación del sistema de origen y la clave de origen.
overrides	object	Reemplaza los valores de campo de un registro raíz y las coincidencias de los registros secundarios.
MERGE	object	Reemplaza los valores de campo de los registros secundarios que quiere fusionar. Añada el tipo de registro secundario al objeto <code>overrides</code> y, a continuación, añada el objeto <code>MERGE</code> .

En el siguiente ejemplo de código JSON se identifica un registro raíz que se va a fusionar con el registro raíz de destino:

```
{
  keys: [
    {
      rowid: "P2"
    }
  ]
}
```

En el siguiente código se muestra cómo reemplazar un campo del registro raíz de parte y cómo reemplazar los candidatos de fusión para los registros secundarios de teléfono:

```
{
  keys: [
    {
      rowid: "P2"
    }
  ]
  overrides: {
    Party: {
      rowidObject: "P1",
      firstName: "Serge", //override the value for the first name
      Telephone: { // override which Telephone child records to merge
        item:[
          {
            rowidObject: "T1",
            MERGE: {
              item: [ // to remove the original merge candidates, specify null
                null,
                null
              ],
              $original: {
                item: [
                  {key:{rowid: "T2"}},
                  {key:{rowid: "T3"}}
                ]
              }
            }
          },
          {
            rowidObject: "T4",
            MERGE: {
              item: [ // to add or change merge candidates, specify matched records
                {key:{rowid: "T2"}}
              ],
              $original: {
                item: [
                  null
                ]
              }
            }
          }
        ]
      }
    }
  }
}
```

```
}
  }
}
]
```

Ejemplo de solicitud de API

La siguiente solicitud de ejemplo devuelve la vista previa de un registro consolidado:

```
POST http://localhost:8080/cmxcsllocalhost-orcl-DS_UI1/Person/2478245?
action=previewMerge

{
  keys: [
    {
      rowid: "2478246"
    },
    {
      rowid: "2478230"
    }
  ],
  overrides: {
    Person: {
      firstName: "Charlie"
    }
  }
}
```

Ejemplo de respuesta de API

La siguiente respuesta de ejemplo muestra la vista previa del registro consolidado:

```
{
  "Person": {
    "rowidObject": "2478245",
    "partyType": "Person",
    "lastName": "Smith",
    "firstName": "Charlie",
    "displayName": "ALICE SMITH"
  }
}
```

Actualizar fusión pendiente

Para guardar los cambios realizados en los registros que forman parte de una tarea de fusión pendiente, como reemplazos de valores de registro, utilice la API de REST de actualización de fusión pendiente. La API guarda los cambios en función del ID de interacción de los registros.

La API usa el método POST.

URL de solicitud

El componente de ruta de la URL de solicitud debe incluir el ID de fila del registro de destino.

La dirección URL para la actualización de fusión pendiente tiene el siguiente formato:

```
http://<host>:<port>/<context>/<database ID>/<business entity>
/<rowid of the root record>?
action=updatePendingMerge&interactionId=<interaction ID>
```

Haga la siguiente solicitud HTTP POST a la URL para la actualización de fusión pendiente:

```
POST http://<host>:<port>/<context>/<database ID>/<business entity>
/<rowid of the root record>?
action=updatePendingMerge&interactionId=<interaction ID>
```

En el cuerpo de la solicitud, añada las claves y especifique los registros raíz que desea fusionar con el registro raíz de destino. También puede especificar los registros secundarios en los que desea reemplazar las coincidencias.

Parámetro de consulta

En la siguiente tabla se describen los parámetros de consulta que puede usar en la URL:

Parámetro	Descripción
action	<p>Obligatorio. Guarda los cambios, como reemplazos a nivel de campo, en una tarea de fusión pendiente. Establézcalo en <code>updatePendingMerge</code> y use el parámetro con el parámetro <code>interactionId</code>.</p> <p>Por ejemplo, use la siguiente consulta para guardar los cambios realizados en los registros de entidad de negocio de persona que están pendientes de la acción de fusión:</p> <pre>Person?action=updatePendingMerge&interactionId</pre>
interactionId	Obligatorio. ID de interacción de la tarea de fusión pendiente.

Cuerpo de la solicitud

Antes de utilizar la API de actualización de fusión pendiente, utilice la API de REST de lectura de registros coincidentes para determinar cuáles son los registros coincidentes que se pueden fusionar con el registro raíz de destino. Envíe la lista de registros en el cuerpo de la solicitud para la API de actualización de fusión pendiente.

Puede reemplazar los valores de campo en el registro raíz. Por ejemplo, si ninguno de los registros raíz coincidentes contiene la ortografía correcta del nombre, puede especificar el nombre correcto en el cuerpo de la solicitud. Asimismo, puede eliminar los registros coincidentes o especificar otros.

Use las siguientes propiedades en el cuerpo de la solicitud:

Propiedades/ elementos	Tipo	Descripción
keys	array	Obligatorio. Una lista ordenada de los registros raíz coincidentes que desea fusionar. Puede identificar los registros por el ID de fila o mediante una combinación del sistema de origen y la clave de origen.
overrides	object	Reemplaza los valores de campo de un registro raíz y las coincidencias de los registros secundarios.
MERGE	object	Reemplaza los valores de campo de los registros secundarios que quiere fusionar. Añada el tipo de registro secundario al objeto <code>overrides</code> y, a continuación, añada el objeto <code>MERGE</code> .

En el siguiente ejemplo de código JSON se identifican dos registros raíz que se van a fusionar con el registro raíz de destino:

```
{
  keys: [
    {rowid: "2478246"},
    {rowid: "2478230"}
  ]
}
```

En el siguiente cuerpo de solicitud de ejemplo se muestra cómo reemplazar un campo del registro raíz de parte y cómo reemplazar los registros coincidentes para los registros secundarios de teléfono:

```
{
  keys: [
    {
      rowid: "2478246"
    }
  ]
  overrides: {
    Party: {
      rowidObject: "2478230",
      firstName: "Charlie", //Override the value for the first name
      Telephone: { // Specifies the Telephone child records to merge
        item:[
          {
            rowidObject: "2511",
            MERGE: {
              item: [ // To remove the original merge candidates, specify null
                null,
                null
              ],
              $original: {
                item: [
                  {key:{rowid: "2822"}},
                  {key:{rowid: "2733"}}
                ]
              }
            }
          },
          {
            rowidObject: "2644",
            MERGE: {
              item: [ // To add or change merge candidates, specify matched records
                {key:{rowid: "2822"}}
              ],
              $original: {
                item: [
                  null
                ]
              }
            }
          }
        ]
      }
    }
  ]
}
}
```

Ejemplo de solicitud de API

Con la siguiente solicitud de ejemplo se reemplaza el campo de nombre del registro de destino con el valor Charlie:

```
POST http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/Person/2478245?
action=updatePendingMerge&interactionId=3982462873645
```

```
{
  keys: [
    {
      rowid: "2478246"
    }
  ]
}
```

```

    },
    {
      rowid: "2478230"
    }
  ],
  overrides: {
    Person: {
      firstName: "Charlie"
    }
  }
}

```

Ejemplo de respuesta de API

La siguiente respuesta de ejemplo contiene el ID de fila del registro de destino para el que está pendiente la acción de fusión:

```

{
  "Person": {
    "key": {
      "rowid": "2478245"
    },
    "rowidObject": "2478245"
  }
}

```

Fusión pendiente

La API de REST de fusión pendiente actualiza todas las tareas de fusión pendiente que se realizan en un registro en función del ID de interacción de la solicitud de cambio. Con Fusión pendiente, las operaciones de fusión se pueden aplazar hasta que el proceso de flujo de trabajo conceda la aprobación para todas las tareas de fusión.

La API utiliza el método POST y devuelve el ID de fila del registro.

URL de solicitud

La dirección URL para la fusión pendiente tiene el siguiente formato:

```

http://<host>:<port>/<context>/<database ID>/<business entity>/<rowid>?
action=PendingMerge&interactionId=<interaction ID>

```

Haga la siguiente solicitud HTTP POST a la URL para la fusión pendiente:

```

POST http://<host>:<port>/<context>/<database ID>/<business entity>/<rowid of the
record>?action=Merge

```

Parámetro de consulta

El ID de interacción de la fusión pendiente es un parámetro obligatorio.

Ejemplo de solicitud de API

El siguiente ejemplo de solicitud actualiza todas las tareas de fusión pendiente que están asociadas con el ID de interacción:

```

POST /Person/123?action=pendingMerge&interactionId=123

```

Ejemplo de respuesta de API

El siguiente ejemplo de respuesta contiene los ID de fila de los objetos base raíz afectados:

```
{
  keys: [{rowid: "456"}, {rowid: "789"}],
  overrides: {...}
}
```

Promover fusión

La API de REST para promover la fusión ejecuta todas las tareas de fusión pendiente que hay asociadas con el ID de interacción de la solicitud de cambio.

La API utiliza el método POST y devuelve el ID de fila del registro ganador.

URL de solicitud

La dirección URL para la promoción de la fusión tiene el siguiente formato:

```
http://<host>:<port>/<context>/<database ID>/<business entity>/<rowid>?
action=PromoteMerge&interactionId=<interaction ID>
```

Haga la siguiente solicitud HTTP POST a la URL para promover la fusión:

```
POST http://<host>:<port>/<context>/<database ID>/<business entity>/<rowid of the
record>?action=Merge
```

Parámetro de consulta

El ID de interacción de la tarea de fusión pendiente es un parámetro obligatorio. La API utiliza el ID de interacción para buscar todas las tareas de fusión pendiente y ejecuta las fusiones.

Ejemplo de solicitud de API

El siguiente ejemplo de solicitud promueve todas las tareas de fusión pendiente que están asociadas con el ID de interacción:

```
POST /Person/123?action=promoteMerge&interactionId=123
```

Ejemplo de respuesta de API

El siguiente ejemplo de respuesta contiene los ID de fila de los registros después de promover las tareas de fusión pendiente:

```
POST /Person/123?action=promoteMerge&interactionId=123
```

Transformación de archivos

La API de transformación de archivos recupera la lista de entidades de negocio del archivo importado antes de asignar las entidades de negocio a los campos de destino.

La API usa el método POST.

URL de solicitud

La API de transformación de archivos tiene el siguiente formato:

```
http://<host>:<port>/cmx/flat2be/{orsId}/preview{?entity,fileId,pageSize}
```

Haga la siguiente solicitud HTTP POST a la URL de la API de transformación de archivos:

```
POST /cmx/flat2be/{orsId}/preview{?entity,fileId,pageSize}
```

Parámetros

Especifique las propiedades para obtener una vista previa de la lista de entidades de negocio asignadas desde el archivo importado antes de asignarlas a los campos de destino.

En la siguiente tabla, se describen los parámetros.

Parámetro	Descripción
fileId	ID de archivo que se usará para generar la vista previa.
orsId	ID de almacén de datos de referencia operativa de la base de datos desde la cual obtener la vista previa de la asignación.
entity	Opcional. El número de entidades de negocio o relaciones de las cuales obtener una vista previa.
pageSize	Opcional. Número de páginas por devolver.

Ejemplo de solicitud de API

La siguiente solicitud de ejemplo recupera la lista de entidades de negocio asignadas desde el archivo importado antes de asignarlas:

```
POST /cmx/flat2be/localhost-orcl-MDM_SAMPLE/preview?entity=Person&fileId=TEMP_SVR1.1G7UW
HTTP/1.1
Content-Type: application/json
Host: localhost:8080
Content-Length: 431

{
  "regionalSettings" : {
    "locale" : "en-CA",
    "datePattern" : "d MMM, yyyy",
    "decimalSeparator" : ".",
    "thousandsSeparator" : ",",
  },
  "objects" : [ {
    "name" : "Person",
    "fields" : [ {
      "name" : "firstName",
      "fileColumn" : "first_name",
      "skipNulls" : false
    }, {
      "name" : "lastName",
      "fileColumn" : "last_name",
      "skipNulls" : false
    } ]
  } ]
}
```

Ejemplo de respuesta de API

La siguiente respuesta de ejemplo muestra la lista de entidades de negocio asignadas desde el archivo importado antes de asignarlas a los campos de destino.

```
HTTP/1.1 200 OK
Server: JBoss-EAP/7
X-Powered-By: Undertow/1
Content-Type: application/json
```


Content-Length: 825

```
{
  "link" : [ ],
  "item" : [ {
    "object" : {
      "Person" : {
        "label" : "smith, alice",
        "lastName" : "smith",
        "firstName" : "alice"
      }
    }
  }, {
    "object" : {
      "Person" : {
        "label" : "smith, bob",
        "lastName" : "smith",
        "firstName" : "bob"
      }
    }
  }, {
    "object" : {
      "Person" : {
        "label" : "smith, charlie",
        "lastName" : "smith",
        "firstName" : "charlie"
      }
    }
  }, {
    "object" : {
      "Person" : {
        "label" : "smith, dave",
        "lastName" : "smith",
        "firstName" : "dave"
      }
    }
  }, {
    "object" : {
      "Person" : {
        "label" : "smith, eve",
        "lastName" : "smith",
        "firstName" : "eve"
      }
    }
  } ]
}
```

Sugerir asignación

La API de REST de sugerencia de asignación sugiere asignaciones para las entidades de negocio o relaciones en el archivo importado.

La API usa el método GET.

URL de solicitud

La API de sugerencia de asignación tiene el siguiente formato:

```
http://<host>:<port>/<context>/<orsId>/<suggest{?entity,fileId,purpose}>
```

Haga la siguiente solicitud HTTP GET a la URL de la API de sugerencia de asignación:

```
GET http://<host>:<port>/<context>/<orsId>/<suggest{?entity,fileId,purpose}>
```

Cuerpo de la solicitud

Especifique los parámetros para sugerir asignación para las relaciones o entidades de negocio en el archivo importado.

En la tabla siguiente, se describen los parámetros en el cuerpo de la solicitud.

Parámetro	Tipo	Descripción
orsld	Ruta de acceso	ID de la base de datos del almacén de referencias operativas desde la que se obtiene una vista previa de la asignación.
fileId	Consulta	Opcional. ID de archivo del archivo importado.
purpose	Consulta	Opcional. El propósito de la asignación y cuándo se usará.
entity	Consulta	Opcional. El número de entidades de negocio o relaciones de las cuales obtener una vista previa. Si especifica el parámetro de la entidad, la API detecta y devuelve la asignación para la entidad especificada solamente.

Ejemplo de solicitud de API

La siguiente solicitud de ejemplo devuelve la asignación sugerida para las entidades de negocio o relaciones en el archivo importado:

```
GET /cmx/flat2be/localhost-orcl-MDM_SAMPLE/suggest?fileId=TEMP_SVR1.1G7UW HTTP/1.1
Host: localhost:8080
```

Ejemplo de respuesta de API

La siguiente respuesta de ejemplo muestra la asignación sugerida para las entidades de negocio o relaciones en el archivo importado:

```
HTTP/1.1 200 OK
Server: JBoss-EAP/7
X-Powered-By: Undertow/1
Content-Type: application/json;charset=UTF-8
Content-Length: 1315
```

```
{
  "name" : "my_mapping",
  "systemName" : "Admin",
  "mapping" : {
    "confidence" : "HIGH",
    "regionalSettings" : {
      "locale" : "en-CA",
      "datePattern" : "MM-dd-yyyy",
      "decimalSeparator" : ".",
      "thousandsSeparator" : ","
    },
  },
  "objects" : [ {
    "confidence" : "HIGH",
    "name" : "PersonView",
    "fields" : [ {
      "confidence" : "HIGH",
      "name" : "firstName",
      "fileColumn" : "first_name",
      "skipNulls" : false
    }, {
      "confidence" : "HIGH",
      "name" : "jobTitle",
      "fileColumn" : "job_title",
      "skipNulls" : false
    }
  ]
}
```

```

    }, {
      "confidence" : "HIGH",
      "name" : "lastName",
      "fileColumn" : "last_name",
      "skipNulls" : false
    }, {
      "confidence" : "HIGH",
      "name" : "birthdate",
      "fileColumn" : "birth_date",
      "skipNulls" : false
    }, {
      "confidence" : "HIGH",
      "name" : "genderCd",
      "fileColumn" : "gender",
      "skipNulls" : false
    } ],
    "children" : [ ]
  } ],
  "purpose" : "IMPORT"
},
"creator" : "admin",
"createDate" : "2020-01-13T12:39:37.186-05:00",
"updatedBy" : "admin",
"lastUpdateDate" : "2020-01-13T12:39:37.186-05:00",
"system" : false
}

```

Fusionar registros

La API de REST para la fusión de registros fusiona dos o más registros raíz para crear un único registro consolidado. El ID de fila del registro consolidado es el ID de fila del registro en el que se fusionan los demás registros.

La API utiliza el método POST. Puede especificar los reemplazos a nivel de campo para el registro fusionado en el cuerpo de la solicitud.

URL de solicitud

La URL para la fusión de registros tiene el siguiente formato:

```

http://<host>:<port>/<context>/<database ID>/<business entity>/<rowid of the record>?
action=merge

```

Haga la siguiente solicitud HTTP POST a la URL para la fusión de registros:

```

POST http://<host>:<port>/<context>/<database ID>/<business entity>/<rowid of the
record>?action=merge

```

Agregue el encabezado Content-Type para especificar el tipo de medio de los datos que desea enviar con la solicitud:

```

POST http://<host>:<port>/<context>/<database ID>/<business entity>/<rowid of the
record>?action=merge
Content-Type: application/<json/xml>

```

Parámetros de consulta

La siguiente tabla muestra los parámetros que puede usar en la URL:

Parámetro	Descripción
taskComment	Añada un comentario a la tarea de flujo de trabajo que ha activado la API.
taskAttachments	Si los adjuntos de tarea están habilitados, adjunte un archivo a la tarea de flujo de trabajo que ha activado la API.

Cuerpo de la solicitud

Antes de empezar, use la API de vista previa de fusión para obtener una vista previa de los resultados de la fusión de los registros raíz seleccionados. Si considera que la vista previa es satisfactoria, use las mismas propiedades en el cuerpo de la solicitud para la API de fusión de registros.

Puede reemplazar los valores de campo en el registro raíz. Por ejemplo, si ninguno de los registros raíz coincidentes contiene la ortografía correcta del nombre, puede especificar el nombre correcto en el cuerpo de la solicitud. Asimismo, puede eliminar los registros coincidentes o especificar otros.

Use las siguientes propiedades en el cuerpo de la solicitud:

Propiedades/elementos	Tipo	Descripción
keys	array	Obligatorio. Una lista ordenada de los registros raíz coincidentes que desea fusionar. Puede identificar los registros por el ID de fila o mediante una combinación del sistema de origen y la clave de origen.
overrides	object	Reemplaza los valores de campo de un registro raíz y las coincidencias de los registros secundarios.
MERGE	object	Reemplaza los valores de campo de los registros secundarios que quiere fusionar. Añada el tipo de registro secundario al objeto <code>overrides</code> y, a continuación, añada el objeto <code>MERGE</code> .

En el siguiente ejemplo de código JSON se identifican dos registros raíz que se van a fusionar con el registro raíz de destino:

```
{
  keys: [
    {rowid: "2478246"},
    {rowid: "2478269"}
  ]
}
```

Para ver un ejemplo de uso de las propiedades `overrides` y `MERGE` con la API de fusión de registros, consulte el cuerpo de la solicitud para la API de vista previa de fusión.

Ejemplo de solicitud de API

La siguiente solicitud de ejemplo fusiona registros para formar un registro consolidado. La solicitud añade un comentario y un adjunto a la tarea de flujo de trabajo que ha activado la API.

```
POST http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/Person/2478245?
action=merge&taskComment=Read my comment&taskAttachments=TEMP_SVR1.290T8,TEMP_SVR1.290T9
Content-Type: application/<json/xml>

{
  keys: [
    {
      rowid: "2478246"
    }
  ],
  overrides: {
    Person: {
      firstName: "Charlie"
    }
  }
}
```

Ejemplo de respuesta de API

La siguiente respuesta de ejemplo muestra el registro consolidado:

```
{
  "Person": {
    "link": [
      {
        "href": "http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/Person/
2478245",
        "rel": "self"
      }
    ],
    "key": {
      "rowid": "2478245"
    },
    "rowidObject": "2478245"
  }
}
```

Anular fusión de registros

La API de REST para la anulación de fusión de registros anula la fusión de un registro raíz en los registros raíz individuales que existían antes de que se fusionaran los registros.

La API utiliza el método POST.

URL de solicitud

La URL para la anulación de la fusión de registros tiene el siguiente formato:

```
http://<host>:<port>/<context>/<database ID>/<business entity>/<rowId>?action=unmerge
```

Haga la siguiente solicitud HTTP POST a la URL para la anulación de la fusión de registros:

```
POST http://<host>:<port>/<context>/<database ID>/<business entity>/<rowId>?
action=unmerge
```

Agregue el encabezado Content-Type para especificar el tipo de medio de los datos que desea enviar con la solicitud:

```
POST http://<host>:<port>/<context>/<database ID>/<business entity>/<rowId>?
action=unmerge
Content-Type: application/<json/xml>
```

Parámetros de consulta

La siguiente tabla muestra los parámetros que puede usar en la URL:

Parámetro	Descripción
taskComment	Añada un comentario a la tarea de flujo de trabajo que ha activado la API.
taskAttachments	Si los adjuntos de tarea están habilitados, adjunte un archivo a la tarea de flujo de trabajo que ha activado la API.

Cuerpo de la solicitud

Envíe la lista de registros cuya fusión del registro consolidado desea anular en el cuerpo de la solicitud. Utilice el ID de fila xref o el sistema de origen y la clave de fuente para especificar los registros.

Utilice la API de lectura de registros para obtener el ID de fila (rowId) de XREF del registro cuya fusión se va a anular. En la siguiente solicitud de ejemplo se recuperan los metadatos de XREF de un registro:

```
GET http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/Person/2638243?
contentMetadata=XREF
```

Ejemplo de solicitud de API

La siguiente solicitud de ejemplo anula la fusión de un registro del registro consolidado. La solicitud añade un comentario y un adjunto a la tarea de flujo de trabajo que ha activado la API.

```
POST http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/Person/2478248?
action=unmerge&taskComment=Read my
comment&taskAttachments=TEMP_SVR1.29OT8,TEMP_SVR1.29OT9

{
  rowid: "4880369"
}
```

Ejemplo de respuesta de API

La siguiente respuesta de ejemplo muestra el registro cuya fusión ha anulado del registro consolidado:

```
{
  "Person": {
    "link": [
      {
        "href": "http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/Person/
2478249",
        "rel": "self"
      }
    ],
    "key": {
      "rowid": "2478249"
    },
    "rowidObject": "2478249"
  }
}
```

Crear asignación

La API de REST de creación de asignación crea una nueva asignación al importar un nuevo archivo en la base de datos.

La API usa el método POST.

URL de solicitud

La dirección URL de creación de la asignación tiene el siguiente formato:

```
http://<host>:<port>/<context>/<orsId>/<mapping>
```

Haga la siguiente solicitud POST HTTP a la URL para la creación de asignaciones:

```
POST //<host>:<port>/<context>/<orsId>/<mapping>
```

Parámetros

Especifique los parámetros para crear una nueva asignación en el cuerpo de la solicitud. Use el formato JSON o XML para crear una asignación en la solicitud.

En la tabla siguiente se describen los parámetros del cuerpo de la solicitud:

Parámetro	Tipo	Descripción
orsID	Ruta de acceso	ID de la base de datos del almacén de referencias operativas para crear una nueva asignación.
mapping	Consulta	Opcional. Introduzca las propiedades de la asignación.
confidence	Consulta	Opcional. Nivel de confianza de la asignación. Use cualquiera de los siguientes valores: alto, medio o bajo.
creator	Consulta	Opcional. El nombre de usuario.
createDate	Consulta	Opcional. La fecha en la que se crea la asignación.
UpdatedBy	Consulta	Opcional. Si está actualizando una asignación existente, introduzca su nombre de usuario.
lastUpdateDate	Consulta	Opcional. Fecha en la que el usuario actualiza la asignación.
modified	Consulta	Opcional. Use los siguientes valores: true si está modificando una asignación existente o false si está creando una nueva asignación.
name	Consulta	Opcional. Nombre de la asignación.
score	Consulta	Opcional. Puntuación de coincidencia de la asignación.
system	Consulta	Opcional. Use los siguientes valores: True: si el sistema crea la asignación. False: si el usuario manualmente crea la asignación.
systemName	Consulta	Opcional. Nombre del sistema que crea la asignación.
objects	Consulta	Opcional. Introduzca los campos que quiera para el objeto base de la entidad de negocio o relación.

Ejemplo de solicitud de API

La siguiente solicitud de ejemplo crea una nueva asignación:

```
POST /cmx/flat2be/localhost-orcl-MDM_SAMPLE/mapping HTTP/1.1
Content-Type: application/json
Host: localhost:8080
Content-Length: 737
```

```

{
  "mapping" : {
    "regionalSettings" : {
      "locale" : "en-CA",
      "datePattern" : "dd-MM-yyyy",
      "decimalSeparator" : ".",
      "thousandsSeparator" : ","
    },
    "objects" : [ {
      "name" : "PersonView",
      "fields" : [ {
        "name" : "firstName",
        "fileColumn" : "first_name",
        "skipNulls" : false
      }, {
        "name" : "jobTitle",
        "fileColumn" : "job_title",
        "skipNulls" : false
      }, {
        "name" : "lastName",
        "fileColumn" : "last_name",
        "skipNulls" : false
      }, {
        "name" : "birthdate",
        "fileColumn" : "birth_date",
        "skipNulls" : false
      } ]
    } ],
    "purpose" : "IMPORT"
  }
}

```

Ejemplo de respuesta de API

La siguiente respuesta de ejemplo muestra la nueva asignación creada en la base de datos.

```

HTTP/1.1 201 Created
Server: JBoss-EAP/7
X-Powered-By: Undertow/1
Location: http://localhost:8080/cmx/flat2be/localhost-orcl-MDM_SAMPLE/mapping/SVR1.1G7UX
Content-Type: application/json;charset=UTF-8
Content-Length: 27

{
  "id" : "SVR1.1G7UX"
}

```

Vista previa de asignación

La API de REST de vista previa de asignación recupera la lista de las entidades de negocio asignadas desde el archivo importado.

La API usa el método GET.

URL de solicitud

La dirección URL de vista previa de asignación tiene el siguiente formato:

```
http://<host>:<port>/<context>/<orsId>/<preview{?entity,fileId,mapping,pageSize}>
```

Haga la siguiente solicitud HTTP GET a la URL de la API de vista previa de asignación:

```
GET http://<host>:<port>/<context>/<orsId>/<preview{?entity,fileId,mapping,pageSize}>
```


Parámetros

Especifique los parámetros para obtener la vista previa de la lista de entidades de negocio asignadas desde el archivo importado.

En la tabla siguiente se describen los parámetros del cuerpo de la solicitud:

Parámetro	Tipo	Descripción
orsld	Ruta de acceso	ID de almacén de datos de referencia operativa de la base de datos desde la cual obtener la vista previa de la asignación.
fileId	Consulta	ID de archivo. Si se especifica, la API recupera las entidades de negocio asignadas desde el ID de archivo especificado.
mappingID	Consulta	Opcional. Introduzca el ID de asignación del archivo importado.
entity	Consulta	Opcional. Introduzca el número de entidades de negocio o relaciones de las cuales obtener vista previa.
pageSize	Consulta	Opcional. Introduzca el número de páginas o los registros por devolver.

Ejemplo de solicitud de API

La siguiente solicitud de ejemplo proporciona una vista previa de las entidades de negocio asignadas desde el archivo importado:

```
GET /cmx/flat2be/localhost-orcl-MDM_SAMPLE/preview?
mapping=SVR1.1G7UX&entity=PersonView&fileId=TEMP_SVR1.1G7UW HTTP/1.1
Host: localhost:8080
```

Ejemplo de respuesta de API

La siguiente respuesta de ejemplo muestra una vista previa de las entidades de negocio asignadas desde el archivo importado:

```
HTTP/1.1 200 OK
Server: JBoss-EAP/7
X-Powered-By: Undertow/1
Content-Type: application/json
Content-Length: 1751
```

```
{
  "link" : [ ],
  "item" : [ {
    "object" : {
      "PersonView" : {
        "firstName" : "alice",
        "lastName" : "smith",
        "birthdate" : "1950-01-01T00:00:00-05:00",
        "genderCd" : {
          "genderCode" : "F",
          "genderDisp" : "F"
        },
        "jobTitle" : "developer"
      }
    }
  }, {
    "object" : {
```

```

    "PersonView" : {
      "firstName" : "bob",
      "lastName" : "smith",
      "birthdate" : "1960-02-02T00:00:00-05:00",
      "genderCd" : {
        "genderCode" : "M",
        "genderDisp" : "M"
      },
      "jobTitle" : "tester"
    }
  }, {
    "object" : {
      "PersonView" : {
        "firstName" : "charlie",
        "lastName" : "smith",
        "birthdate" : "1970-03-03T00:00:00-05:00",
        "genderCd" : {
          "genderCode" : "X",
          "genderDisp" : "X"
        },
        "jobTitle" : "manager"
      }
    }
  }, {
    "object" : {
      "PersonView" : {
        "firstName" : "dave",
        "lastName" : "smith",
        "genderCd" : {
          "genderCode" : "M",
          "genderDisp" : "M"
        },
        "jobTitle" : "accountant"
      }
    }
  },
  "errors" : [ {
    "code" : "SIP-55206",
    "message" : "Failed to convert value [ten years ago] for field [birthdate].",
    "row" : 3,
    "column" : "birth_date",
    "field" : "birthdate"
  } ]
}, {
  "object" : {
    "PersonView" : {
      "firstName" : "eve",
      "lastName" : "smith",
      "birthdate" : "1990-04-04T00:00:00-04:00",
      "genderCd" : {
        "genderCode" : "F",
        "genderDisp" : "F"
      },
      "jobTitle" : ""
    }
  }
} ]
}

```

Buscar asignación

La API de REST de búsqueda de asignación encuentra la asignación definida por el usuario o sistema existente por ID de asignación.

La API usa el método GET.

URL de solicitud

La dirección URL de búsqueda de la asignación tiene el siguiente formato:

```
http://<host>:<port>/<context>/<orsId>/<mapping{?
  createdBy, fileId, firstRecord, purpose, recordsToReturn, returnTotal}>
```

Haga la siguiente solicitud HTTP GET a la URL para buscar la asignación:

```
GET http://<host>:<port>/<context>/<orsId>/<mapping {?
  createdBy, fileId, firstRecord, purpose, recordsToReturn, returnTotal}>
```

Parámetros

Especifique los parámetros para encontrar la asignación definida por el usuario o sistema existente por ID de asignación.

En la tabla siguiente se describen los parámetros del cuerpo de la solicitud:

Parámetro	Ruta de acceso	Descripción
orsId	Ruta de acceso	ID de la base de datos del almacén de referencias operativas para encontrar la asignación.
fileId	Consulta	ID de archivo. Si se especifica, la API recupera la asignación desde el ID de archivo especificado.
createdBy	Consulta	Introduzca el usuario que creó la asignación.
firstRecord	Consulta	Opcional. Introdúzcala solo para devolver la primera fila del registro.
purpose	Consulta	Opcional. Use los siguientes valores: Import: si crea una nueva asignación. Match: si realiza la coincidencia con una asignación existente.
recordsToReturn	Consulta	Opcional. Introduzca el número de registros por devolver.
returnTotal	Consulta	Opcional. Introduzca el número total de registros.

Ejemplo de solicitud de API

La siguiente solicitud de ejemplo encuentra la asignación definida por el usuario o sistema existente por ID de asignación:

```
GET http://<host>:<port>/cmx/flat2be/{orsId}/mapping{?
  createdBy, fileId, firstRecord, purpose, recordsToReturn, returnTotal}
```

Ejemplo de respuesta de API

La siguiente respuesta de ejemplo muestra la asignación definida por el usuario o sistema existente por ID de asignación:

```
{
  mappings: [
    {
      id: 123, name: "...", ...
    },
    ...
  ]
}
```

```
} ]
```

Leer asignación

La API de REST para lectura de asignaciones devuelve la asignación de las relaciones o entidades de negocio existentes por ID de asignación.

La API usa el método GET.

URL de solicitud

La dirección URL de lectura de asignación tiene el siguiente formato:

```
http://<host>:<port>/<context>/<orsId>/<mapping>/<id><fileId>
```

Haga la siguiente solicitud HTTP GET a la URL para lectura de asignación:

```
GET http://<host>:<port>/<context>/<orsId>/<mapping>/<id><fileId>
```

Parámetros

Especifique los parámetros para devolver las relaciones o entidades de negocio asignadas desde el ID del archivo importado.

En la tabla siguiente, se describen los parámetros en el cuerpo de la solicitud.

Parámetro	Tipo	Descripción
mappingID	Ruta de acceso	Nombre de la asignación dado por el usuario.
orsId	Ruta de acceso	ID de la base de datos del almacén de referencias operativas desde la que se recupera la asignación.
fileId	Consulta	Opcional. Especifique el ID de archivo asociado con la asignación existente.

Ejemplo de solicitud de API

La siguiente solicitud de ejemplo devuelve las relaciones o entidades de negocio asignadas por ID de asignación:

```
GET /cmx/flat2be/localhost-orcl-MDM_SAMPLE/mapping/SVR1.1G7UX HTTP/1.1
Content-Type: application/json
Host: localhost:8080
```

Ejemplo de respuesta de API

La siguiente respuesta de ejemplo devuelve las relaciones o entidades de negocio existentes por ID de asignación:

```
HTTP/1.1 200 OK
Server: JBoss-EAP/7
Content-Disposition: inline;filename=f.txt
X-Powered-By: Undertow/1
Content-Type: application/json;charset=UTF-8
Content-Length: 993
```

```
{
  "id" : "SVR1.1G7UX",
```

```

"systemName" : "Admin",
"mapping" : {
  "regionalSettings" : {
    "locale" : "en-CA",
    "datePattern" : "dd-MM-yyyy",
    "decimalSeparator" : ".",
    "thousandsSeparator" : ",",
  },
  "objects" : [ {
    "name" : "PersonView",
    "fields" : [ {
      "name" : "firstName",
      "fileColumn" : "first_name",
      "skipNulls" : false
    }, {
      "name" : "jobTitle",
      "fileColumn" : "job_title",
      "skipNulls" : false
    }, {
      "name" : "lastName",
      "fileColumn" : "last_name",
      "skipNulls" : false
    }, {
      "name" : "birthdate",
      "fileColumn" : "birth_date",
      "skipNulls" : false
    } ],
    "children" : [ ]
  } ],
  "purpose" : "IMPORT"
},
"creator" : "admin",
"createDate" : "2020-01-13T13:06:26.153-05:00",
"updatedBy" : "admin",
"lastUpdateDate" : "2020-01-13T13:06:26.153-05:00",
"system" : true
}

```

Actualizar asignación

La API de REST de actualización de la asignación actualiza la asignación existente con datos nuevos.

La API utiliza el método PUT.

URL de solicitud

La dirección URL de actualización de la asignación tiene el siguiente formato:

```
http://<host>:<port>/<context>/<orsId>/<mapping>/<mappingId>
```

Haga la siguiente solicitud HTTP PUT a la dirección URL de actualización de la asignación:

```
PUT http://<host>:<port>/<context>/<orsId>/<mapping>/<mappingId>
```

Parámetros

Especifique los parámetros para actualizar la asignación definida por el usuario en el cuerpo de la solicitud. Use el formato JSON o XML para crear una asignación en la solicitud.

En la tabla siguiente, se describen los parámetros en el cuerpo de la solicitud.

Parámetro	Tipo	Descripción
mappingID	Ruta de acceso	ID de asignación de los datos de usuario.
orsId	Ruta de acceso	ID de la base de datos del almacén de referencia operativa.
mapping	Consulta	Opcional. Introduzca las propiedades de la asignación.
confidence	Consulta	Opcional. Nivel de confianza de la asignación. Use cualquiera de los siguientes valores: alto, medio o bajo.
creator	Consulta	Opcional. El nombre de usuario.
createDate	Consulta	Opcional. La fecha en la que se crea la asignación.
UpdatedBy	Consulta	Opcional. Si está actualizando una asignación existente, introduzca su nombre de usuario.
lastUpdateDate	Consulta	Opcional. Fecha en la que el usuario actualiza la asignación.
modified	Consulta	Opcional. Use los siguientes valores: true si está modificando una asignación existente o false si está creando una nueva asignación.
name	Consulta	Opcional. Nombre de la asignación.
score	Consulta	Opcional. Puntuación de coincidencia de la asignación.
system	Consulta	Opcional. Use los siguientes valores: True: si el sistema crea la asignación. False: si el usuario manualmente crea la asignación.
systemName	Consulta	Opcional. Muestra el nombre del sistema que crea la asignación.
objects	Consulta	Opcional. Introduzca los campos que quiera para el objeto base de la entidad de negocio o relación.
purpose	Consulta	Opcional. Use los siguientes valores: Import: Si crea una nueva asignación. Match: si realiza la coincidencia con una asignación existente.

Ejemplo de solicitud de API

La siguiente solicitud de ejemplo actualiza la asignación existente con datos nuevos.

```
PUT /cmx/flat2be/localhost-orcl-MDM_SAMPLE/mapping/SVR1.1G7UX HTTP/1.1
Content-Type: application/json
Host: localhost:8080
Content-Length: 889

{
  "system" : false,
  "name" : "my_mapping",
```

```

"mapping" : {
  "regionalSettings" : {
    "locale" : "en-CA",
    "datePattern" : "dd-MM-yyyy",
    "decimalSeparator" : ".",
    "thousandsSeparator" : ",",
  },
  "objects" : [ {
    "name" : "PersonView",
    "fields" : [ {
      "name" : "firstName",
      "fileColumn" : "first_name",
      "skipNulls" : false
    }, {
      "name" : "jobTitle",
      "fileColumn" : "job_title",
      "skipNulls" : false
    }, {
      "name" : "lastName",
      "fileColumn" : "last_name",
      "skipNulls" : false
    }, {
      "name" : "birthdate",
      "fileColumn" : "birth_date",
      "skipNulls" : false
    }, {
      "name" : "genderCd",
      "fileColumn" : "gender",
      "skipNulls" : false
    } ]
  } ],
  "purpose" : "IMPORT"
}

```

Ejemplo de respuesta de API

La siguiente es la respuesta de ejemplo después de actualizar satisfactoriamente la asignación existente con datos nuevos:

```

HTTP/1.1 204 No Content
Server: JBoss-EAP/7
X-Powered-By: Undertow/1

```

Transformación de archivos

La API de transformación de archivos recupera la lista de entidades de negocio del archivo importado antes de asignar las entidades de negocio a los campos de destino.

La API usa el método POST.

URL de solicitud

La API de transformación de archivos tiene el siguiente formato:

```

http://<host>:<port>/cmx/fla2be/{orsId}/preview{?entity,fileId,pageSize}

```

Haga la siguiente solicitud HTTP POST a la URL de la API de transformación de archivos:

```

POST /cmx/fla2be/{orsId}/preview{?entity,fileId,pageSize}

```

Parámetros

Especifique las propiedades para obtener una vista previa de la lista de entidades de negocio asignadas desde el archivo importado antes de asignarlas a los campos de destino.

En la siguiente tabla, se describen los parámetros.

Parámetro	Descripción
fileId	ID de archivo que se usará para generar la vista previa.
orsId	ID de almacén de datos de referencia operativa de la base de datos desde la cual obtener la vista previa de la asignación.
entity	Opcional. El número de entidades de negocio o relaciones de las cuales obtener una vista previa.
pageSize	Opcional. Número de páginas por devolver.

Ejemplo de solicitud de API

La siguiente solicitud de ejemplo recupera la lista de entidades de negocio asignadas desde el archivo importado antes de asignarlas:

```
POST /cmx/flat2be/localhost-orcl-MDM_SAMPLE/preview?entity=Person&fileId=TEMP_SVR1.1G7UW
HTTP/1.1
Content-Type: application/json
Host: localhost:8080
Content-Length: 431

{
  "regionalSettings" : {
    "locale" : "en-CA",
    "datePattern" : "d MMM, yyyy",
    "decimalSeparator" : ".",
    "thousandsSeparator" : ",",
  },
  "objects" : [ {
    "name" : "Person",
    "fields" : [ {
      "name" : "firstName",
      "fileColumn" : "first_name",
      "skipNulls" : false
    }, {
      "name" : "lastName",
      "fileColumn" : "last_name",
      "skipNulls" : false
    } ]
  } ]
}
```

Ejemplo de respuesta de API

La siguiente respuesta de ejemplo muestra la lista de entidades de negocio asignadas desde el archivo importado antes de asignarlas a los campos de destino.

```
HTTP/1.1 200 OK
Server: JBoss-EAP/7
X-Powered-By: Undertow/1
Content-Type: application/json
Content-Length: 825

{
  "link" : [ ],
  "item" : [ {
```



```

    "object" : {
      "Person" : {
        "label" : "smith, alice",
        "lastName" : "smith",
        "firstName" : "alice"
      }
    }, {
      "object" : {
        "Person" : {
          "label" : "smith, bob",
          "lastName" : "smith",
          "firstName" : "bob"
        }
      }
    }, {
      "object" : {
        "Person" : {
          "label" : "smith, charlie",
          "lastName" : "smith",
          "firstName" : "charlie"
        }
      }
    }, {
      "object" : {
        "Person" : {
          "label" : "smith, dave",
          "lastName" : "smith",
          "firstName" : "dave"
        }
      }
    }, {
      "object" : {
        "Person" : {
          "label" : "smith, eve",
          "lastName" : "smith",
          "firstName" : "eve"
        }
      }
    }
  ]
}

```

Leer una relación

La API de REST de lectura de relaciones devuelve los detalles de un registro de relación, como los tipos de parte, los valores de rowID y los nombres para mostrar de los dos registros.

La API usa el método GET.

URL de solicitud

La dirección URL de lectura de relaciones tiene el siguiente formato:

```
http://<host>:<port>/<context>/<database ID>/<relationship>/<row ID of the relationship record>
```

Haga la siguiente solicitud HTTP GET a la URL:

```
GET http://<host>:<port>/<context>/<database ID>/<relationship>/<row ID of the relationship record>
```

Parámetros de consulta

En la tabla siguiente se muestran los parámetros de la consulta:

Parámetro	Descripción
suppressLinks	Opcional. Indica si los vínculos entre elementos primarios y secundarios se pueden ver en la respuesta de la API. Establezca el parámetro en true para suprimir todos los vínculos entre elementos primarios y secundarios en la respuesta. Establezca el parámetro en false para no mostrar los vínculos en la respuesta de la API. El valor predeterminado es false.
depth	Opcional. El número de niveles secundarios que se devolverán.

Ejemplo de solicitud de API

En la siguiente solicitud de ejemplo se devuelven los detalles del registro de relación con el ID de fila 85, lo que se corresponde con el tipo de relación `ProductGroupProductGroupIsParentOfProductProducts`:

```
GET http://localhost:8080/cm/cs/localhost-orcl-DS_UI1/
ProductGroupProductGroupIsParentOfProductProducts/85
```

En la siguiente solicitud de ejemplo se devuelven detalles con una profundidad igual a 2:

```
GET http://localhost:8080/cm/cs/localhost-orcl-DS_UI1/
ProductGroupProductGroupIsParentOfProductProducts/85?depth=2
```

Ejemplo de respuesta de API

En el siguiente ejemplo se muestran los detalles del registro de relación con el ID de fila 85, que es el tipo de relación `ProductGroupProductGroupIsParentOfProductProducts`:

```
{
  "link": [
    {
      "href": "http://localhost:8080/cm/cs/localhost-orcl-DS_UI1/
ProductGroupProductGroupIsParentOfProductProducts/85.json",
      "rel": "self"
    },
    {
      "href": "http://localhost:8080/cm/cs/localhost-orcl-DS_UI1/
ProductGroupProductGroupIsParentOfProductProducts/85.json?depth=2",
      "rel": "children"
    }
  ],
  "rowidObject": "85",
  "label": "Product Group Product Group is parent of Product Products",
  "rowidRelType": "9",
  "rowidHierarchy": "3",
  "from": {
    "link": [
      {
        "href": "http://localhost:8080/cm/cs/localhost-orcl-DS_UI1/
ProductGroupProductGroupIsParentOfProductProducts/85.json",
        "rel": "parent"
      },
      {
        "href": "http://localhost:8080/cm/cs/localhost-orcl-DS_UI1/
ProductGroupProductGroupIsParentOfProductProducts/85/from/86.json",
        "rel": "self"
      },
      {
        "href": "http://localhost:8080/cm/cs/localhost-orcl-DS_UI1/
ProductGroupProductGroupIsParentOfProductProducts/85/from/86.json?depth=2",
        "rel": "children"
      }
    ]
  }
}
```

```

    }
  ],
  "rowidObject": "86",
  "label": "Product Group",
  "productType": "Product Group",
  "productNumber": "Presenter2",
  "productName": "Presenter",
  "productDesc": "Presenter Family",
  "productTypeCd": {
    "link": [
      {
        "href": "http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/
ProductGroupProductGroupIsParentOfProductProducts/85/from/86.json",
        "rel": "parent"
      },
      {
        "href": "http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/
ProductGroupProductGroupIsParentOfProductProducts/85/from/86/productTypeCd.json?depth=2",
        "rel": "children"
      },
      {
        "href": "http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/
ProductGroupProductGroupIsParentOfProductProducts/85/from/86/productTypeCd.json",
        "rel": "self"
      }
    ]
  },
  "productType": "Product Group"
}
},
"to": {
  "link": [
    {
      "href": "http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/
ProductGroupProductGroupIsParentOfProductProducts/85/to/66.json",
      "rel": "self"
    },
    {
      "href": "http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/
ProductGroupProductGroupIsParentOfProductProducts/85/to/66.json?depth=2",
      "rel": "children"
    },
    {
      "href": "http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/
ProductGroupProductGroupIsParentOfProductProducts/85.json",
      "rel": "parent"
    }
  ]
},
"rowidObject": "66",
"label": "Products",
"productType": "Product",
"productNumber": "931307-0403",
"productName": "2.4 GHz Cordless Presenter",
"productDesc": "A cordless presenter to streamline your delivery.",
"productTypeCd": {
  "link": [
    {
      "href": "http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/
ProductGroupProductGroupIsParentOfProductProducts/85/to/66/productTypeCd.json",
      "rel": "self"
    },
    {
      "href": "http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/
ProductGroupProductGroupIsParentOfProductProducts/85/to/66.json",
      "rel": "parent"
    },
    {
      "href": "http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/
ProductGroupProductGroupIsParentOfProductProducts/85/to/66/productTypeCd.json?depth=2",
      "rel": "children"
    }
  ]
}
],

```

```
        "productType": "Presenter"
    }
}
```

Crear una relación

La API de REST de creación de relaciones permite crear una relación entre los registros especificados. Para crear una relación entre registros, deben existir relaciones entre las entidades de negocio a las que pertenecen los registros. Por ejemplo, si desea especificar una relación entre Informática y John Smith, debe existir una relación entre las entidades de negocio Persona y Organización. Debe enviar los datos de la relación en el cuerpo de la solicitud.

La API usa los métodos PUT y POST.

URL de solicitud

La dirección URL de REST de creación de relaciones tiene el siguiente formato:

```
http://<host>:<port>/<context>/<database ID>/<relationship>?systemName=<name of the source system>
```

Nota: El nombre del sistema de origen es un parámetro obligatorio en la URL.

Use la siguiente solicitud HTTP POST o PUT para la dirección URL:

```
POST http://<host>:<port>/<context>/<database ID>/<relationship>?systemName=<name of the source system>
```

Agregue el encabezado Content-Type para especificar el tipo de medio de los datos que desea enviar con la solicitud:

```
Content-Type: application/<json/xml>
```

Parámetros de URL

El nombre del sistema de origen es un parámetro obligatorio en la URL de solicitud.

Cuerpo de la solicitud

Envíe los datos para el registro de relación en el cuerpo de la solicitud REST. Utilice el formato JSON o el formato XML para enviar los datos. Especifique los valores de los parámetros obligatorios en el cuerpo de la solicitud.

Ejemplo de solicitud de API

La siguiente solicitud de ejemplo crea la relación `OrganizationEmploysPerson` entre una entidad de negocio de organización con el ID de fila 101 y una entidad de negocio de persona con el ID de fila 1101:

```
POST http://localhost:8080/cmxcs/localhost-orcl-DS_UI1/OrganizationEmploysPerson?
systemName=SFA
Content-Type: application/json

{
  "from": {
    "rowidObject": "101"
  },
  "to": {
    "rowidObject": "1101"
  },
}
```

```

    "relName": "Documentation",
    "relDesc": "Writer"
  }

```

La relación `OrganizationEmploysPerson` define una relación de una entidad de negocio de organización con una entidad de negocio de persona. El elemento `from` especifica el registro en el que se origina la relación y el elemento `to` especifica el registro en el que finaliza la relación.

Ejemplo de respuesta de API

La siguiente respuesta de ejemplo muestra el encabezado y el cuerpo de la respuesta después de crear correctamente una relación entre una entidad de negocio de organización con el ID de fila 101 y una entidad de negocio de persona con el ID de fila 1101:

```

{
  "OrganizationEmploysPerson": {
    "key": {
      "rowid": "414721"
      "sourceKey": "SVR1.1E7UW"
    }-
  }-
  "rowidObject": "414721"
  "from": {
    "key": {
      "rowid": "101 "
    }-
    "rowidObject": "101 "
  }-
  "to": {
    "key": {
      "rowid": "1101 "
    }-
  }-
  "rowidObject": "1101 "
}

```

Actualizar una relación

La API de REST de actualización de relaciones actualiza la relación entre dos registros. La API actualiza los atributos adicionales definidos para la relación.

La API usa los métodos POST y PUT.

URL de solicitud

La dirección URL de actualización de relaciones tiene el siguiente formato:

```

http://<host>:<port>/<context>/<database ID>/<relationship>/<row ID>?systemName=<name of the source system>

```

Nota: El nombre del sistema de origen es un parámetro obligatorio en la URL.

Use la siguiente llamada a HTTP POST o PUT para la dirección URL:

```

http://<host>:<port>/<context>/<database ID>/<relationship>/<row ID>?systemName=<name of the source system>

```

Agregue el encabezado `Content-Type` para especificar el tipo de medio de los datos que desea enviar con la solicitud:

Cuerpo de la solicitud

Envíe las actualizaciones al registro de relación en el cuerpo de la solicitud. Utilice el formato JSON o el formato XML para enviar los datos. Especifique los valores de los parámetros obligatorios en el cuerpo de la solicitud.

Ejemplo de solicitud de API

La relación con el ID de fila 414721 es una relación OrganizationEmploysPerson entre una entidad de organización con el ID de fila 101 y una entidad de persona con el ID de fila 1101.

La siguiente solicitud de ejemplo actualiza el registro de relación con el ID de fila 414721:

```
POST http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/OrganizationEmploysPerson/414721?
systemName=SFA
Content-Type: application/json
{
  "from": {
    "rowidObject": "101"
  },
  "to": {
    "rowidObject": "1101"
  },
  "relName": "Development",
  "relDesc": "Software Engineer",
  "$original": {
    "relName": "Documentation",
    "relDesc": "Writer"
  }
}
```

Ejemplo de respuesta de API

La siguiente respuesta de ejemplo se recibe una vez que se ha actualizado correctamente la relación con el ID de fila 414721:

```
{
  "OrganizationEmploysPerson": {
    "key": {
      "rowid": "414721"
      "sourceKey": "SVR1.1E7UW"
    }-
    "rowidObject": "414721"
    "from": {
      "key": {
        "rowid": "101"
      }-
      "rowidObject": "101"
    }-
    "to": {
      "key": {
        "rowid": "1101"
      }-
      "rowidObject": "1101"
    }-
  }-
}
```

Eliminar una relación

La API de REST de eliminación de relaciones elimina la relación entre dos registros.

La API usa el método DELETE.

URL de solicitud

La dirección URL de eliminación de relaciones tiene el siguiente formato:

```
http://<host>:<port>/<context>/<database ID>/<relationship>/<rowID of the relationship record>?systemName=<name of the source system>
```

Nota: El nombre del sistema de origen es un parámetro obligatorio en la URL.

Haga la siguiente solicitud HTTP DELETE a la dirección URL de eliminación:

```
DELETE http://<host>:<port>/<context>/<database ID>/<relationship>/<rowID of the relationship record>?systemName=<name of the source system>
```

Parámetro de consulta

El nombre del sistema de origen es un parámetro de URL obligatorio. Use el parámetro `systemName` para especificar el sistema de origen.

Ejemplo de solicitud de API

En la siguiente solicitud de ejemplo se elimina un registro de relación con el ID de fila 414721:

```
DELETE http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/OrganizationEmploysPerson/414721?systemName=SFA
```

Ejemplo de respuesta de API

La siguiente respuesta de ejemplo muestra la respuesta después de eliminar correctamente el registro de relación con el ID de fila 414721:

```
{
  "OrganizationEmploysPerson": {
    "key": {
      "rowid": "414721"
    }-
  "rowidObject": "414721"
  }-
}
```

Obtener registros relacionados

La API de REST de obtención de registros relacionados devuelve una lista de registros relacionados con un registro raíz especificado y basado en las relaciones que ha configurado. La API devuelve también los detalles de las relaciones.

La API usa el método GET.

URL de solicitud

La URL para la obtención de registros relacionados tiene el siguiente formato:

```
http://<host>:<port>/<context>/<database ID>/<business entity>/<rowId>?action=related
```

Haga la siguiente solicitud HTTP GET a la URL para la obtención de registros relacionados:

```
GET http://<host>:<port>/<context>/<database ID>/<business entity>/<rowId>?action=related
```

Parámetros de consulta

Puede anexar los parámetros de consulta a la dirección URL de la solicitud.

En la siguiente tabla se indican los parámetros de consulta que puede usar:

Parámetro	Descripción
recordsToReturn	Número de registros para los elementos secundarios <i>many</i> que desea leer.
searchToken	Token de búsqueda para obtener las páginas siguientes del conjunto de resultados.
returnTotal	Devuelve el número de registros en el conjunto de resultados. Se establece en <i>true</i> para obtener el número de registros de la lista de resultados. El valor predeterminado es <i>false</i> .

Parámetros de filtro

Puede anexar parámetros a la dirección URL para filtrar los registros relacionados.

En la siguiente tabla se indican los parámetros de filtro que puede usar:

Parámetro	Descripción
recordStates	Lista de estados de registro separados por comas que desea recuperar. Los estados de registro admitidos son <i>ACTIVE</i> , <i>PENDING</i> y <i>DELETED</i> . El valor predeterminado es <i>ACTIVE</i> . Por ejemplo, la consulta <code>/Party/123?action=related&recordStates=ACTIVE,PENDING</code> devuelve registros activos o pendientes.
entityLabel	Etiqueta de la entidad.
relationshipLabel	Etiqueta de la relación.
entityType	Lista de tipos de entidad separados por comas. Por ejemplo, la lista <code>entityType=Person,Organization</code> devuelve registros relacionados de los tipos de entidad <i>Person</i> y <i>Organization</i> .
relationshipType	Lista de tipos de relación separados por comas. Por ejemplo, la lista <code>relationshipType=Employee,Employer</code> devuelve registros relacionados de los tipos de relación <i>Employee</i> y <i>Employer</i> .

Nota: Si especifica varias condiciones de filtro, el resultado contiene todos los registros que cumplen la condición *AND*.

Cuerpo de la respuesta

El cuerpo de la respuesta contiene la lista de registros relacionados, los detalles de las relaciones y los registros relacionados, y un token de búsqueda. Utilice el token de búsqueda para obtener las páginas posteriores del resultado.

Ejemplo de solicitud de API

En la siguiente solicitud de ejemplo se obtienen los registros relacionados y las relaciones configuradas para la entidad de negocio de organización con el ID de fila 101:

```
GET http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/Organization/101?action=related
```

Ejemplo de respuesta de API

En la siguiente respuesta de ejemplo se muestran los registros relacionados y las relaciones para el tipo de entidad de negocio de organización con el ID de fila 101:

```
{
  "link": [],
  "firstRecord": 1,
  "pageSize": 10,
  "searchToken": "SVR1.1H7YB",
  "relatedEntity": [
    {
      "businessEntity": {
        "SecurePerson": {
          "link": [
            {
              "href": "http://10.21.43.42:8080/cmx/cs/localhost-orcl-ds_ui1/SecurePerson/1101",
              "rel": "self"
            }
          ]
        },
        "rowidObject": "1101",
        "creator": "admin",
        "createDate": "2008-11-11T21:22:20-08:00",
        "updatedBy": "admin",
        "lastUpdateDate": "2012-03-29T19:03:19-07:00",
        "consolidationInd": "1",
        "lastRowidSystem": "SYS0",
        "dirtyIndicator": "0",
        "interactionId": "20003000",
        "hubStateInd": "1",
        "partyType": "Person",
        "lastName": "Obama",
        "firstName": "Barack"
      }
    },
    {
      "entityLabel": "Obama,Barack",
      "relationshipLabel": "Organization employes SecurePerson",
      "relationship": {
        "rowidObject": "414721",
        "creator": "admin",
        "createDate": "2016-10-17T01:58:12.436-07:00",
        "updatedBy": "admin",
        "lastUpdateDate": "2016-10-19T01:40:28.830-07:00",
        "consolidationInd": "4",
        "lastRowidSystem": "SFA",
        "interactionId": "1476866426786",
        "hubStateInd": "1",
        "rowidRelType": "101",
        "rowidHierarchy": "1",
        "relName": "Documentation",
        "relDesc": "Writer"
      },
      "entityType": "SecurePerson",
      "relationshipType": "OrganizationEmployesSecurePerson"
    }
  ],
  {
    "businessEntity": {
      "SecurePerson": {
        "link": [
          {
            "href": "http://10.21.43.42:8080/cmx/cs/localhost-orcl-ds_ui1/SecurePerson/114",

```

```

        "rel": "self"
    }
},
"rowidObject": "114",
"creator": "admin",
"createDate": "2008-08-11T23:00:55-07:00",
"updatedBy": "Admin",
"lastUpdateDate": "2008-08-12T02:59:17-07:00",
"consolidationInd": "1",
"lastRowidSystem": "Legacy",
"dirtyIndicator": "0",
"hubStateInd": "1",
"partyType": "Person",
"lastName": "HERNANDEZ",
"displayName": "ALEJANDRO HERNANDEZ",
"firstName": "ALEJANDRO"
},
"entityLabel": "HERNANDEZ,ALEJANDRO",
"relationshipLabel": "Organization employes SecurePerson",
"relationship": {
    "rowidObject": "434721",
    "creator": "admin",
    "createDate": "2016-10-19T01:49:03.415-07:00",
    "updatedBy": "admin",
    "lastUpdateDate": "2016-10-19T01:49:03.415-07:00",
    "consolidationInd": "4",
    "lastRowidSystem": "SFA",
    "hubStateInd": "1",
    "rowidRelType": "101",
    "rowidHierarchy": "1",
    "relName": "Documentation",
    "relDesc": "Writer"
},
"entityType": "SecurePerson",
"relationshipType": "OrganizationEmployesSecurePerson"
},
{
    "businessEntity": {
        "Person": {
            "link": [
                {
                    "href": "http://10.21.43.42:8080/cmxcs/localhost-orcl-ds_ui1/Person/1101",
                    "rel": "self"
                }
            ]
        }
    },
    "rowidObject": "1101",
    "creator": "admin",
    "createDate": "2008-11-11T21:22:20-08:00",
    "updatedBy": "admin",
    "lastUpdateDate": "2012-03-29T19:03:19-07:00",
    "consolidationInd": "1",
    "lastRowidSystem": "SYS0",
    "dirtyIndicator": "0",
    "interactionId": "20003000",
    "hubStateInd": "1",
    "partyType": "Person",
    "lastName": "Obama",
    "firstName": "Barack"
},
"entityLabel": "Obama,Barack",
"relationshipLabel": "Organization employes Person",
"relationship": {
    "rowidObject": "414721",
    "creator": "admin",
    "createDate": "2016-10-17T01:58:12.436-07:00",
    "updatedBy": "admin",
    "lastUpdateDate": "2016-10-19T01:40:28.830-07:00",
    "consolidationInd": "4",
    "lastRowidSystem": "SFA",

```

```

        "interactionId": "1476866426786",
        "hubStateInd": "1",
        "rowidRelType": "101",
        "rowidHierarchy": "1",
        "relName": "Documentation",
        "relDesc": "Writer"
    },
    "entityType": "Person",
    "relationshipType": "OrganizationEmploYESPerson"
},
{
    "businessEntity": {
        "Person": {
            "link": [
                {
                    "href": "http://10.21.43.42:8080/cmx/cs/localhost-orcl-ds_uil/Person/114",
                    "rel": "self"
                }
            ]
        },
        "rowidObject": "114",
        "creator": "admin",
        "createDate": "2008-08-11T23:00:55-07:00",
        "updatedBy": "Admin",
        "lastUpdateDate": "2008-08-12T02:59:17-07:00",
        "consolidationInd": "1",
        "lastRowidSystem": "Legacy",
        "dirtyIndicator": "0",
        "hubStateInd": "1",
        "partyType": "Person",
        "lastName": "HERNANDEZ",
        "displayName": "ALEJANDRO HERNANDEZ",
        "statusCd": "A",
        "firstName": "ALEJANDRO"
    }
},
"entityLabel": "HERNANDEZ,ALEJANDRO",
"relationshipLabel": "Organization employes Person",
"relationship": {
    "rowidObject": "434721",
    "creator": "admin",
    "createDate": "2016-10-19T01:49:03.415-07:00",
    "updatedBy": "admin",
    "lastUpdateDate": "2016-10-19T01:49:03.415-07:00",
    "consolidationInd": "4",
    "lastRowidSystem": "SFA",
    "hubStateInd": "1",
    "rowidRelType": "101",
    "rowidHierarchy": "1",
    "relName": "Documentation",
    "relDesc": "Writer"
},
"entityType": "Person",
"relationshipType": "OrganizationEmploYESPerson"
}
]
}

```

Exportar entidades de negocio relacionadas

Escriba aquí una breve descripción del concepto (obligatorio).

Este es el inicio del concepto.

URL de solicitud

Escriba aquí una breve descripción del concepto (obligatorio).

Este es el inicio del concepto.

Ejemplo de solicitud de API

Escriba aquí una breve descripción del concepto (obligatorio).

Este es el inicio del concepto.

Ejemplo de respuesta de API

Escriba aquí una breve descripción del concepto (obligatorio).

Este es el inicio del concepto.

Lista Jerarquías

La API de REST de enumeración de jerarquías devuelve todas las jerarquías. También puede usar la API para devolver las jerarquías que contienen una entidad de negocio especificada en cualquier nivel.

La API usa el método GET.

URL de solicitud

La dirección URL de enumeración de jerarquías puede tener los siguientes formatos:

Dirección URL para devolver todas las jerarquías

Use la siguiente dirección URL de enumeración de todas las jerarquías:

```
http://<host>:<port>/<context>/<database ID>/metadata/hierarchy
```

Dirección URL para devolver todas las jerarquías que contienen una entidad de negocio especificada

Use la siguiente dirección URL para enumerar todas las jerarquías que contengan una entidad de negocio especificada en cualquier nivel:

```
http://<host>:<port>/<context>/<database ID>/metadata/hierarchy?entityName=<entity name>
```

Ejemplo de solicitud de API

La siguiente solicitud de ejemplo enumera todas las jerarquías:

```
GET http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/metadata/hierarchy
```

La siguiente solicitud de ejemplo enumera todas las jerarquías que contienen la entidad de negocio Product:

```
GET http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/metadata/hierarchy?entityName=Product
```

Ejemplo de respuesta de API

La siguiente respuesta de ejemplo muestra las jerarquías que contienen la entidad de negocio Product:

```
{
  "link": [],
  "item": [
    {
      "operations": {
        "read": {
          "allowed": true
        },
        "update": {
          "allowed": true
        }
      }
    },
    "root": "ProductGroup",
  ]
}
```

```

        "name": "Product"
    }
]
}

```

Obtener metadatos de jerarquía

La API de REST de obtención de metadatos de jerarquía devuelve los metadatos de una jerarquía.

La API usa el método GET.

URL de solicitud

La dirección URL de obtención de metadatos de jerarquía tiene el siguiente formato:

```
http://<host>:<port>/<context>/<database ID>/metadata/hierarchy/<hierarchy name>
```

Use la siguiente solicitud HTTP GET para la dirección URL de obtención de metadatos de jerarquía:

```
GET http://<host>:<port>/<context>/<database ID>/metadata/hierarchy/<hierarchy name>
```

Ejemplo de solicitud de API

La siguiente solicitud de ejemplo muestra los metadatos para una jerarquía:

```
GET http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/metadata/hierarchy/ODI
```

Ejemplo de respuesta de API

La siguiente respuesta de ejemplo muestra los metadatos para una jerarquía:

```

{
  "relationship": [
    {
      "field": [
        {
          "allowedValues": [
            "2"
          ],
          "name": "rowidRelType",
          "label": "Rowid Rel Type",
          "dataType": "String",
          "length": 14,
          "readOnly": false,
          "required": false,
          "system": false,
          "trust": false,
          "applyNullValues": false,
          "displayFormat": "DATETIME_LONG_FORMAT",
          "filterable": true,
          "sortable": true,
          "lookup": {
            "link": [
              {
                "href": "http://localhost:8080/cmx/lookup/localhost-orcl-MDM_SAMPLE/id-label/IsOdiParentOf/rowidRelType",
                "rel": "lookup"
              },
              {
                "href": "http://localhost:8080/cmx/lookup/localhost-orcl-MDM_SAMPLE/object/IsOdiParentOf/rowidRelType",
                "rel": "list"
              }
            ]
          },
          "object": "RboRelType",
          "key": "rowidObject",

```

```

        "value":"displayName"
    }
},
{
    "allowedValues":[
        "1"
    ],
    "name":"rowidHierarchy",
    "label":"Rowid Hierarchy",
    "dataType":"String",
    "length":14,
    "readOnly":false,
    "required":false,
    "system":false,
    "trust":false,
    "applyNullValues":false,
    "displayFormat":"DATETIME_LONG_FORMAT",
    "filterable":true,
    "sortable":true,
    "lookup":{
        "link":[
            {
                "href":"http://localhost:8080/cm/lookup/localhost-orcl-MDM_SAMPLE/id-
label/IsOdiParentOf/rowidHierarchy",
                "rel":"lookup"
            },
            {
                "href":"http://localhost:8080/cm/lookup/localhost-orcl-MDM_SAMPLE/
object/IsOdiParentOf/rowidHierarchy",
                "rel":"list"
            }
        ],
        "object":"RboHierarchy",
        "key":"rowidObject",
        "value":"displayName"
    }
},
{
    "name":"relName",
    "label":"Rel Name",
    "dataType":"String",
    "length":50,
    "trust":false,
    "applyNullValues":false,
    "displayFormat":"DATETIME_LONG_FORMAT",
    "filterable":true,
    "sortable":true
},
{
    "name":"relDesc",
    "label":"Rel Desc",
    "dataType":"String",
    "length":200,
    "trust":false,
    "applyNullValues":false,
    "displayFormat":"DATETIME_LONG_FORMAT",
    "filterable":true,
    "sortable":true
},
{
    "name":"consolidationInd",
    "label":"Consolidation Ind",
    "dataType":"Integer",
    "length":38,
    "readOnly":true,
    "system":true,
    "trust":false,
    "applyNullValues":false,
    "filterable":true,
    "sortable":true
},

```

```

{
  "name": "creator",
  "label": "Creator",
  "dataType": "String",
  "length": 50,
  "readOnly": true,
  "system": true,
  "trust": false,
  "applyNullValues": false,
  "filterable": true,
  "sortable": true
},
{
  "name": "interactionId",
  "label": "Interaction Id",
  "dataType": "Integer",
  "length": 38,
  "readOnly": true,
  "system": true,
  "trust": false,
  "applyNullValues": false,
  "filterable": true,
  "sortable": true
},
{
  "name": "updatedBy",
  "label": "Updated By",
  "dataType": "String",
  "length": 50,
  "readOnly": true,
  "system": true,
  "trust": false,
  "applyNullValues": false,
  "filterable": true,
  "sortable": true
},
{
  "name": "lastUpdateDate",
  "label": "Last Update Date",
  "dataType": "Date",
  "readOnly": true,
  "system": true,
  "trust": false,
  "applyNullValues": false,
  "filterable": true,
  "sortable": true
},
{
  "name": "lastRowidSystem",
  "label": "Last Rowid System",
  "dataType": "String",
  "length": 14,
  "readOnly": true,
  "system": true,
  "trust": false,
  "applyNullValues": false,
  "filterable": true,
  "sortable": true
},
{
  "name": "dirtyIndicator",
  "label": "Dirty Indicator",
  "dataType": "Integer",
  "length": 38,
  "readOnly": true,
  "system": true,
  "trust": false,
  "applyNullValues": false,
  "filterable": true,
  "sortable": true
},
}

```

```

{
  "name": "deletedBy",
  "label": "Deleted By",
  "dataType": "String",
  "length": 50,
  "readOnly": true,
  "system": true,
  "trust": false,
  "applyNullValues": false,
  "filterable": true,
  "sortable": true
},
{
  "name": "deletedInd",
  "label": "Deleted Indicator",
  "dataType": "Integer",
  "length": 38,
  "readOnly": true,
  "system": true,
  "trust": false,
  "applyNullValues": false,
  "filterable": true,
  "sortable": true
},
{
  "name": "hubStateInd",
  "label": "Hub State Ind",
  "dataType": "Integer",
  "length": 38,
  "readOnly": true,
  "system": true,
  "trust": false,
  "applyNullValues": false,
  "filterable": true,
  "sortable": true
},
{
  "name": "deletedDate",
  "label": "Deleted Date",
  "dataType": "Date",
  "readOnly": true,
  "system": true,
  "trust": false,
  "applyNullValues": false,
  "filterable": true,
  "sortable": true
},
{
  "name": "rowidObject",
  "label": "Rowid Object",
  "dataType": "String",
  "length": 14,
  "readOnly": true,
  "system": true,
  "trust": false,
  "applyNullValues": false,
  "filterable": true,
  "sortable": true
},
{
  "name": "cmDirtyInd",
  "label": "Content metadata dirty Ind",
  "dataType": "Integer",
  "length": 38,
  "readOnly": true,
  "system": true,
  "trust": false,
  "applyNullValues": false,
  "filterable": true,
  "sortable": true
},
}

```



```

    {
      "name": "createDate",
      "label": "Create Date",
      "dataType": "Date",
      "readOnly": true,
      "system": true,
      "trust": false,
      "applyNullValues": false,
      "filterable": true,
      "sortable": true
    }
  ],
  "contentMetadata": [
    {
      "operations": {
        "read": {
          "allowed": true
        },
        "create": {
          "allowed": true
        },
        "update": {
          "allowed": true
        },
        "delete": {
          "allowed": true
        }
      },
      "name": "XREF"
    }
  ],
  "name": "IsOdiParentOf",
  "label": "is ODI parent of",
  "color": "#990066",
  "direction": "ENTITY_1_TO_ENTITY_2",
  "bidirectional": false,
  "from": {
    "dataType": "BusinessEntity",
    "required": true,
    "lookup": {
      "object": "Organization"
    }
  },
  "to": {
    "dataType": "BusinessEntity",
    "required": true,
    "lookup": {
      "object": "Organization"
    }
  },
  "hierarchy": "ODI"
}
],
"operations": {
  "read": {
    "allowed": true
  },
  "update": {
    "allowed": true
  }
},
"root": "Organization",
"name": "ODI"
}

```

Obtener ruta de jerarquía

La API de REST de obtención de rutas de jerarquía devuelve las rutas del registro de una entidad de negocio a la raíz de jerarquía.

La API usa el método GET.

URL de solicitud

La dirección URL de obtención de rutas de jerarquía tiene el siguiente formato:

```
http://<host>:<port>/<context>/<database ID>/hierarchy/<hierarchy name>/entity/<entity name>/<entity ID>/path
```

Haga la siguiente solicitud HTTP GET a la dirección URL de obtención de rutas de jerarquía:

```
GET http://<host>:<port>/<context>/<database ID>/hierarchy/<hierarchy name>/entity/<entity name>/<entity ID>/path
```

Parámetros de consulta

Para filtrar las relaciones, puede anexar los parámetros de consulta a la URL de solicitud.

En la tabla siguiente, se describen los parámetros de la consulta:

Parámetro	Descripción
effectiveDate	La fecha para la cual desea recuperar las relaciones.
interactionId	ID de la interacción.
readLabelsOnly	Indica si desea ejecutar solo etiquetas en el resultado. Los valores son <code>true</code> o <code>false</code> .
readSystemFields	Indica si desea devolver los campos del sistema en el resultado. Los valores son <code>true</code> o <code>false</code> .
recordsToReturn	Especifica el número de filas que se van a devolver.
rejectInteractionId	ID de la interacción para rechazar cambios.
relationshipType	Lista de tipos de relación separados por comas. Por ejemplo, la lista <code>relationshipType=Employee,Employer</code> devuelve registros relacionados de los tipos de relación <code>Employee</code> y <code>Employer</code> .
returnTotal	Devuelve el número de registros en el conjunto de resultados. Se establece en <code>true</code> para obtener el número de registros de la lista de resultados.
rootId	ID de raíz de jerarquía.

Ejemplo de solicitud de API

La siguiente solicitud de ejemplo muestra las rutas desde el registro de una unidad de negocio hacia la raíz de la jerarquía:

```
GET http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/hierarchy/ODI/entity/Organization/35/path
```

Ejemplo de respuesta de API

La siguiente respuesta de ejemplo muestra las rutas desde el registro de entidad de negocio hacia la raíz de la jerarquía:

```
{
  "link": [],
  "item": [
    {
      "businessEntity": {
        "Organization": {
          "link": [
            {
              "href": "http://localhost:8080/cmx/request/hm_icons/hierarchymanager/Hospital/Hospital.png?ors=localhost-orcl-MDM_SAMPLE",
              "rel": "icon"
            },
            {
              "href": "http://localhost:8080/cmx/cs/localhost-orcl-MDM_SAMPLE/Organization/584",
              "rel": "self"
            }
          ]
        },
        "rowidObject": "584",
        "label": "Time Warner Inc",
        "partyType": "Organization",
        "displayName": "Time Warner Inc",
        "dunsNumber": "799527630"
      }
    },
    {
      "entityLabel": "Time Warner Inc",
      "entityType": "Organization",
      "depth": 2,
      "object": {
        "businessEntity": {
          "Organization": {
            "link": [
              {
                "href": "http://localhost:8080/cmx/request/hm_icons/hierarchymanager/Hospital/Hospital.png?ors=localhost-orcl-MDM_SAMPLE",
                "rel": "icon"
              },
              {
                "href": "http://localhost:8080/cmx/cs/localhost-orcl-MDM_SAMPLE/Organization/55",
                "rel": "self"
              }
            ]
          },
          "rowidObject": "55",
          "label": "Historic TW Inc",
          "partyType": "Organization",
          "displayName": "Historic TW Inc",
          "dunsNumber": "958466278"
        }
      },
      "relationshipLabel": "Historic TW Inc",
      "relationshipLabel": "is ODI parent of",
      "relationship": {
        "rowidObject": "834",
        "label": "is ODI parent of",
        "rowidRelType": "2",
        "rowidHierarchy": "1",
        "relName": "Parent",
        "relDesc": "Parent",
        "from": {
          "rowidObject": "584"
        },
        "to": {
          "rowidObject": "55"
        }
      }
    }
  ]
}
```

```

    "entityType": "Organization",
    "relationshipType": "IsOdiParentOf",
    "object": {
      "businessEntity": {
        "Organization": {
          "link": [
            {
              "href": "http://localhost:8080/cmx/request/hm_icons/hierarchymanager/
Hospital/Hospital.png?ors=localhost-orcl-MDM_SAMPLE",
              "rel": "icon"
            },
            {
              "href": "http://localhost:8080/cmx/cs/localhost-orcl-MDM_SAMPLE/
Organization/35",
              "rel": "self"
            }
          ],
          "rowidObject": "35",
          "effectivePeriod": {
            },
            "label": "People Magazine",
            "partyType": "Organization",
            "displayName": "People Magazine",
            "dunsNumber": "011584096"
          }
        }
      },
      "entityLabel": "People Magazine",
      "relationshipLabel": "is ODI parent of",
      "relationship": {
        "rowidObject": "889",
        "label": "is ODI parent of",
        "rowidRelType": "2",
        "rowidHierarchy": "1",
        "relName": "Parent",
        "relDesc": "Parent",
        "from": {
          "rowidObject": "55"
        },
        "to": {
          "rowidObject": "35"
        }
      }
    },
    "entityType": "Organization",
    "relationshipType": "IsOdiParentOf"
  }
}
]
}

```

Obtener elementos principales

La API de REST de obtención de elementos principales devuelve las relaciones de elementos principales directos para un registro de entidad de negocio.

La API usa el método GET.

URL de solicitud

La URL para obtención de elementos principales tiene el siguiente formato:

```

http://<host>:<port>/<context>/<database ID>/hierarchy/<hierarchy name>/entity/<entity
name>/<entity ID>/parent

```

Haga la siguiente solicitud HTTP GET a la URL para obtención de elementos principales:

```
GET http://<host>:<port>/<context>/<database ID>/hierarchy/<hierarchy name>/entity/  
<entity name>/<entity ID>/parent
```

Parámetros de consulta

Para filtrar las relaciones, puede anexar los parámetros de consulta a la URL de solicitud.

En la tabla siguiente, se describen los parámetros de la consulta:

Parámetro	Descripción
effectiveDate	La fecha para la cual desea recuperar las relaciones.
interactionId	ID de la interacción.
readLabelsOnly	Indica si desea ejecutar solo etiquetas en el resultado. Los valores son <code>true</code> o <code>false</code> .
readSystemFields	Indica si desea devolver los campos del sistema en el resultado. Los valores son <code>true</code> o <code>false</code> .
recordsToReturn	Especifica el número de filas que se van a devolver.
rejectInteractionId	ID de la interacción para rechazar cambios.
relationshipType	Lista de tipos de relación separados por comas. Por ejemplo, la lista <code>relationshipType=Employee,Employer</code> devuelve registros relacionados de los tipos de relación <code>Employee</code> y <code>Employer</code> .
returnTotal	Devuelve el número de registros en el conjunto de resultados. Se establece en <code>true</code> para obtener el número de registros de la lista de resultados.
rootId	ID de raíz de jerarquía.

Ejemplo de solicitud de API

La siguiente solicitud de ejemplo muestra las relaciones principales directas para un registro de entidad de negocio:

```
GET http://localhost:8080/cm/cs/localhost-orcl-DS_UI1/hierarchy/ODI/entity/  
Organization/55/parent
```

Ejemplo de respuesta de API

La siguiente respuesta de ejemplo muestra las relaciones principales directas para un registro de entidad de negocio:

```
{  
  "link": [],  
  "firstRecord": 1,  
  "pageSize": 10,  
  "searchToken": "SVR1.1G7VH",  
  "relatedEntity": [  
    {  
      "businessEntity": {  
        "Organization": {  
          "link": [  
            {  
              "href": "http://localhost:8080/cm/cs/localhost-orcl-MDM_SAMPLE/  
Organization/584",  
              "rel": "self"  
            }  
          ]  
        }  
      }  
    }  
  ]  
}
```

```

    },
    {
      "href":"http://localhost:8080/cmX/request/hm_icons/hierarchymanager/
Hospital/Hospital.png?ors=localhost-orcl-MDM_SAMPLE",
      "rel":"icon"
    }
  ],
  "rowidObject":"584",
  "label":"Time Warner Inc",
  "partyType":"Organization",
  "displayName":"Time Warner Inc",
  "dunsNumber":"799527630"
}
},
"entityLabel":"Time Warner Inc",
"relationshipLabel":"is ODI parent of",
"relationship":{
  "rowidObject":"834",
  "label":"is ODI parent of",
  "rowidRelType":"2",
  "rowidHierarchy":"1",
  "relName":"Parent",
  "relDesc":"Parent",
  "from":{
    "rowidObject":"584"
  },
  "to":{
    "rowidObject":"55"
  }
}
},
"entityType":"Organization",
"relationshipType":"IsOdiParentOf"
}
]
}

```

Obtener elemento secundario

La API de REST de obtención de elementos secundarios devuelve las relaciones de elementos secundarios de un registro de entidad de negocio.

La API usa el método GET.

URL de solicitud

La URL de obtención de elementos secundarios tiene el siguiente formato:

```
http://<host>:<port>/<context>/<database ID>/hierarchy/<hierarchy name>/entity/<entity
name>/<entity ID>/children
```

Haga la siguiente solicitud HTTP GET a la URL de obtención de elementos secundarios:

```
GET http://<host>:<port>/<context>/<database ID>/hierarchy/<hierarchy name>/entity/
<entity name>/<entity ID>/children
```

Parámetros de consulta

Para filtrar las relaciones, puede anexar los parámetros de consulta a la URL de solicitud.

En la tabla siguiente, se describen los parámetros de la consulta:

Parámetro	Descripción
effectiveDate	La fecha para la cual desea recuperar las relaciones.
interactionId	ID de la interacción.
readLabelsOnly	Indica si desea ejecutar solo etiquetas en el resultado. Los valores son <code>true</code> o <code>false</code> .
readSystemFields	Indica si desea devolver los campos del sistema en el resultado. Los valores son <code>true</code> o <code>false</code> .
recordsToReturn	Especifica el número de filas que se van a devolver.
rejectInteractionId	ID de la interacción para rechazar cambios.
relationshipType	Lista de tipos de relación separados por comas. Por ejemplo, la lista <code>relationshipType=Employee,Employer</code> devuelve registros relacionados de los tipos de relación <code>Employee</code> y <code>Employer</code> .
returnTotal	Devuelve el número de registros en el conjunto de resultados. Se establece en <code>true</code> para obtener el número de registros de la lista de resultados.
rootId	ID de raíz de jerarquía.

Ejemplo de solicitud de API

La siguiente solicitud de ejemplo enumera las relaciones secundarias para un registro de entidad de negocio:

```
GET http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/hierarchy/ODI/entity/  
Organization/55/children
```

Ejemplo de respuesta de API

La siguiente respuesta de ejemplo muestra las relaciones secundarias para un registro de entidad de negocio:

```
{  
  "link": [],  
  "firstRecord": 1,  
  "pageSize": 10,  
  "searchToken": "SVR1.1G7VF",  
  "relatedEntity": [  
    {  
      "businessEntity": {  
        "organization": {  
          "link": [  
            {  
              "href": "http://localhost:8080/cmx/request/hm_icons/hierarchymanager/  
Hospital/Hospital.png?ors=localhost-orcl-MDM_SAMPLE",  
              "rel": "icon"  
            },  
            {  
              "href": "http://localhost:8080/cmx/cs/localhost-orcl-MDM_SAMPLE/  
Organization/593",  
              "rel": "self"  
            }  
          ]  
        }  
      }  
    ],  
  ]  
}
```

```

        "rowidObject": "593",
        "label": "Turner Broadcasting System Inc",
        "partyType": "Organization",
        "displayName": "Turner Broadcasting System Inc",
        "dunsNumber": "003319068"
    },
    {
        "entityLabel": "Turner Broadcasting System Inc",
        "relationshipLabel": "is ODI parent of",
        "relationship": {
            "rowidObject": "887",
            "label": "is ODI parent of",
            "rowidRelType": "2",
            "rowidHierarchy": "1",
            "relName": "Parent",
            "relDesc": "Parent",
            "from": {
                "rowidObject": "55"
            },
            "to": {
                "rowidObject": "593"
            }
        },
        "entityType": "Organization",
        "relationshipType": "IsOdiParentOf"
    },
    {
        "businessEntity": {
            "Organization": {
                "link": [
                    {
                        "href": "http://localhost:8080/cmz/request/hm_icons/hierarchymanager/Hospital/Hospital.png?ors=localhost-orcl-MDM_SAMPLE",
                        "rel": "icon"
                    },
                    {
                        "href": "http://localhost:8080/cmz/cs/localhost-orcl-MDM_SAMPLE/Organization/414",
                        "rel": "self"
                    }
                ],
                "rowidObject": "414",
                "label": "Historic TW Inc",
                "partyType": "Organization",
                "displayName": "Historic TW Inc",
                "dunsNumber": "007305290"
            }
        },
        "entityLabel": "Historic TW Inc",
        "relationshipLabel": "is ODI parent of",
        "relationship": {
            "rowidObject": "888",
            "label": "is ODI parent of",
            "rowidRelType": "2",
            "rowidHierarchy": "1",
            "relName": "HQ",
            "relDesc": "HQ",
            "from": {
                "rowidObject": "55"
            },
            "to": {
                "rowidObject": "414"
            }
        },
        "entityType": "Organization",
        "relationshipType": "IsOdiParentOf"
    },
    {
        "businessEntity": {
            "Organization": {
                "link": [

```



```

        {
          "href":"http://localhost:8080/cmx/cs/localhost-orcl-MDM_SAMPLE/
Organization/35",
          "rel":"self"
        },
        {
          "href":"http://localhost:8080/cmx/request/hm_icons/hierarchymanager/
Hospital/Hospital.png?ors=localhost-orcl-MDM_SAMPLE",
          "rel":"icon"
        }
      ],
      "rowidObject":35,
      "label":"People Magazine",
      "partyType":"Organization",
      "displayName":"People Magazine",
      "dunsNumber":"011584096"
    }
  },
  "entityLabel":"People Magazine",
  "relationshipLabel":"is ODI parent of",
  "relationship":{
    "rowidObject":889,
    "label":"is ODI parent of",
    "rowidRelType":2,
    "rowidHierarchy":1,
    "relName":"Parent",
    "relDesc":"Parent",
    "from":{
      "rowidObject":55
    },
    "to":{
      "rowidObject":35
    }
  },
  "entityType":"Organization",
  "relationshipType":"IsOdiParentOf"
},
{
  "businessEntity":{
    "Organization":{
      "link":{
        {
          "href":"http://localhost:8080/cmx/cs/localhost-orcl-MDM_SAMPLE/
Organization/391",
          "rel":"self"
        },
        {
          "href":"http://localhost:8080/cmx/request/hm_icons/hierarchymanager/
Hospital/Hospital.png?ors=localhost-orcl-MDM_SAMPLE",
          "rel":"icon"
        }
      ],
      "rowidObject":391,
      "label":"Historic TW Inc",
      "partyType":"Organization",
      "displayName":"Historic TW Inc",
      "dunsNumber":"011816704"
    }
  },
  "entityLabel":"Historic TW Inc",
  "relationshipLabel":"is ODI parent of",
  "relationship":{
    "rowidObject":890,
    "label":"is ODI parent of",
    "rowidRelType":2,
    "rowidHierarchy":1,
    "relName":"HQ",
    "relDesc":"HQ",
    "from":{
      "rowidObject":55
    },
  },

```

```

        "to":{
            "rowidObject":"391"
        }
    },
    "entityType":"Organization",
    "relationshipType":"IsOdiParentOf"
},
{
    "businessEntity":{
        "Organization":{
            "link":[
                {
                    "href":"http://localhost:8080/cmx/cs/localhost-orcl-MDM_SAMPLE/
Organization/448",
                    "rel":"self"
                },
                {
                    "href":"http://localhost:8080/cmx/request/hm_icons/hierarchymanager/
Hospital/Hospital.png?ors=localhost-orcl-MDM_SAMPLE",
                    "rel":"icon"
                }
            ]
        },
        "rowidObject":"448",
        "label":"Historic TW Inc",
        "partyType":"Organization",
        "displayName":"Historic TW Inc",
        "dunsNumber":"069110711"
    }
},
    "entityLabel":"Historic TW Inc",
    "relationshipLabel":"is ODI parent of",
    "relationship":{
        "rowidObject":"891",
        "label":"is ODI parent of",
        "rowidRelType":2,
        "rowidHierarchy":1,
        "relName":"HQ",
        "relDesc":"HQ",
        "from":{
            "rowidObject":"55"
        },
        "to":{
            "rowidObject":"448"
        }
    },
    "entityType":"Organization",
    "relationshipType":"IsOdiParentOf"
},
{
    "businessEntity":{
        "Organization":{
            "link":[
                {
                    "href":"http://localhost:8080/cmx/request/hm_icons/hierarchymanager/
Hospital/Hospital.png?ors=localhost-orcl-MDM_SAMPLE",
                    "rel":"icon"
                },
                {
                    "href":"http://localhost:8080/cmx/cs/localhost-orcl-MDM_SAMPLE/
Organization/545",
                    "rel":"self"
                }
            ]
        },
        "rowidObject":"545",
        "label":"Historic TW Inc",
        "partyType":"Organization",
        "displayName":"Historic TW Inc",
        "dunsNumber":"072077139"
    }
},
    "entityLabel":"Historic TW Inc",

```

```

"relationshipLabel":"is ODI parent of",
"relationship":{
  "rowidObject":"892",
  "label":"is ODI parent of",
  "rowidRelType":"2",
  "rowidHierarchy":"1",
  "relName":"HQ",
  "relDesc":"HQ",
  "from":{
    "rowidObject":"55"
  },
  "to":{
    "rowidObject":"545"
  }
},
"entityType":"Organization",
"relationshipType":"IsOdiParentOf"
},
{
  "businessEntity":{
    "Organization":{
      "link":[
        {
          "href":"http://localhost:8080/cmx/cs/localhost-orcl-MDM_SAMPLE/
Organization/349",
          "rel":"self"
        },
        {
          "href":"http://localhost:8080/cmx/request/hm_icons/hierarchymanager/
Hospital/Hospital.png?ors=localhost-orcl-MDM_SAMPLE",
          "rel":"icon"
        }
      ],
      "rowidObject":"349",
      "label":"Historic TW Inc",
      "partyType":"Organization",
      "displayName":"Historic TW Inc",
      "dunsNumber":"074056123"
    }
  },
  "entityLabel":"Historic TW Inc",
  "relationshipLabel":"is ODI parent of",
  "relationship":{
    "rowidObject":"893",
    "label":"is ODI parent of",
    "rowidRelType":"2",
    "rowidHierarchy":"1",
    "relName":"HQ",
    "relDesc":"HQ",
    "from":{
      "rowidObject":"55"
    },
    "to":{
      "rowidObject":"349"
    }
  },
  "entityType":"Organization",
  "relationshipType":"IsOdiParentOf"
},
{
  "businessEntity":{
    "Organization":{
      "link":[
        {
          "href":"http://localhost:8080/cmx/request/hm_icons/hierarchymanager/
Hospital/Hospital.png?ors=localhost-orcl-MDM_SAMPLE",
          "rel":"icon"
        },
        {
          "href":"http://localhost:8080/cmx/cs/localhost-orcl-MDM_SAMPLE/
Organization/298",

```

```

        "rel":"self"
    }
    ],
    "rowidObject":"298",
    "label":"WF Cinema Holdings LP",
    "partyType":"Organization",
    "displayName":"WF Cinema Holdings LP",
    "dunsNumber":"075408711"
}
},
"entityLabel":"WF Cinema Holdings LP",
"relationshipLabel":"is ODI parent of",
"relationship":{
    "rowidObject":"894",
    "label":"is ODI parent of",
    "rowidRelType":"2",
    "rowidHierarchy":"1",
    "relName":"Parent",
    "relDesc":"Parent",
    "from":{
        "rowidObject":"55"
    },
    "to":{
        "rowidObject":"298"
    }
}
},
"entityType":"Organization",
"relationshipType":"IsOdiParentOf"
},
{
    "businessEntity":{
        "Organization":{
            "link":{
                {
                    "href":"http://localhost:8080/cmx/request/hm_icons/hierarchymanager/Hospital/Hospital.png?ors=localhost-orcl-MDM_SAMPLE",
                    "rel":"icon"
                },
                {
                    "href":"http://localhost:8080/cmx/cs/localhost-orcl-MDM_SAMPLE/Organization/596",
                    "rel":"self"
                }
            ],
            "rowidObject":"596",
            "label":"Historic TW Inc",
            "partyType":"Organization",
            "displayName":"Historic TW Inc",
            "dunsNumber":"076749246"
        }
    },
    "entityLabel":"Historic TW Inc",
    "relationshipLabel":"is ODI parent of",
    "relationship":{
        "rowidObject":"895",
        "label":"is ODI parent of",
        "rowidRelType":"2",
        "rowidHierarchy":"1",
        "relName":"HQ",
        "relDesc":"HQ",
        "from":{
            "rowidObject":"55"
        },
        "to":{
            "rowidObject":"596"
        }
    },
    "entityType":"Organization",
    "relationshipType":"IsOdiParentOf"
},
{

```

```

    "businessEntity":{
      "Organization":{
        "link":[
          {
            "href":"http://localhost:8080/cmx/request/hm_icons/hierarchymanager/
Hospital/Hospital.png?ors=localhost-orcl-MDM_SAMPLE",
            "rel":"icon"
          },
          {
            "href":"http://localhost:8080/cmx/cs/localhost-orcl-MDM_SAMPLE/
Organization/699",
            "rel":"self"
          }
        ],
        "rowidObject":"699",
        "label":"Historic TW Inc",
        "partyType":"Organization",
        "displayName":"Historic TW Inc",
        "dunsNumber":"083115568"
      }
    },
    "entityLabel":"Historic TW Inc",
    "relationshipLabel":"is ODI parent of",
    "relationship":{
      "rowidObject":"896",
      "label":"is ODI parent of",
      "rowidRelType":"2",
      "rowidHierarchy":"1",
      "relName":"HQ",
      "relDesc":"HQ",
      "from":{
        "rowidObject":"55"
      },
      "to":{
        "rowidObject":"699"
      }
    },
    "entityType":"Organization",
    "relationshipType":"IsOdiParentOf"
  }
]
}

```

Exportar jerarquía

La API de REST de exportación de jerarquía exporta una jerarquía en formato CSV.

La API usa el método GET.

URL de solicitud

La dirección URL de exportación de jerarquía tiene el siguiente formato:

```

http://<host>:<port>/<context>/<database ID>/hierarchy/<hierarchy name>/entity/<entity
name>/<entity ID>/exportHierarchy

```

Use la siguiente solicitud HTTP GET para la dirección URL de exportación de jerarquía:

```

GET http://<host>:<port>/<context>/<database ID>/hierarchy/<hierarchy name>/entity/
<entity name>/<entity ID>/exportHierarchy

```

Parámetros de consulta

Para filtrar las relaciones, puede anexar los parámetros de consulta a la URL de solicitud.

En la tabla siguiente, se describen los parámetros de la consulta:

Parámetro	Descripción
effectiveDate	La fecha para la cual desea recuperar las relaciones.
interactionId	ID de la interacción.
readLabelsOnly	Indica si desea ejecutar solo etiquetas en el resultado. Los valores son true o false.
readSystemFields	Indica si desea devolver los campos del sistema en el resultado. Los valores son true o false.
recordsToReturn	Especifica el número de filas que se van a devolver.
rejectInteractionId	ID de la interacción para rechazar cambios.
relationshipType	Lista de tipos de relación separados por comas. Por ejemplo, la lista relationshipType=Employee,Employer devuelve registros relacionados de los tipos de relación Employee y Employer.
returnTotal	Devuelve el número de registros en el conjunto de resultados. Se establece en true para obtener el número de registros de la lista de resultados.
rootId	ID de raíz de jerarquía.

Ejemplo de solicitud de API

La siguiente solicitud de ejemplo exporta la jerarquía para un registro de entidad de negocio:

```
GET http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/hierarchy/ODI/entity/
Organization/584/exportHierarchy
```

Ejemplo de respuesta de API

La siguiente respuesta de ejemplo muestra la jerarquía exportada en el formato CSV:

```
Level,Business Entity,ID,Label,Parent ID,Parent Label,Type of Relationship to
Parent,issuingCompany,expirationYear,accountType,accountNumber,securityCode,expirationMon
th,cardholderName,productType,productNumber,productName,inceptionDate,productDesc,groupTy
pe,groupName,partyType,lastName,displayName,middleName,birthdate,firstName,dunsNumber
Root,Organization,584,Time Warner Inc,,,,,,,,,,,,,Organization,,Time Warner Inc,,,,
799527630
1,Organization,55,Historic TW Inc,584,Time Warner Inc,is ODI parent
of,,,,,,,,,,,,,Organization,,Historic TW Inc,,,,958466278
1,Organization,206,Courtroom Television Network LLC,584,Time Warner Inc,is ODI parent
of,,,,,,,,,,,,,Organization,,Courtroom Television Network LLC,,,,043905707
1,Organization,674,Rebellion Pictures LLC,584,Time Warner Inc,is ODI parent
of,,,,,,,,,,,,,Organization,,Rebellion Pictures LLC,,,,557414278
1,Organization,881,"Time Warner Cable Programming, Inc",584,Time Warner Inc,is ODI
parent of,,,,,,,,,,,,,Organization,,Time Warner Cable Programming, Inc,,,,787474949
...
```

Exportar elementos principales y secundarios directos

La API de REST de exportación de elementos principales y secundarios directos exporta las relaciones principales y secundarias directas para un registro de entidad de negocio en formato CSV.

La API usa el método GET.

URL de solicitud

La dirección URL de exportación de elementos principales y secundarios directos tiene el siguiente formato:

```
http://<host>:<port>/<context>/<database ID>/hierarchy/<hierarchy name>/entity/<entity name>/<entity ID>/exportRelated
```

Haga la siguiente solicitud HTTP GET a la URL de exportación de elementos principales y secundarios directos:

```
GET http://<host>:<port>/<context>/<database ID>/hierarchy/<hierarchy name>/entity/<entity name>/<entity ID>/exportRelated
```

Parámetros de consulta

Para filtrar las relaciones, puede anexar los parámetros de consulta a la URL de solicitud.

En la tabla siguiente, se describen los parámetros de la consulta:

Parámetro	Descripción
effectiveDate	La fecha para la cual desea recuperar las relaciones.
interactionId	ID de la interacción.
readLabelsOnly	Indica si desea ejecutar solo etiquetas en el resultado. Los valores son <code>true</code> o <code>false</code> .
readSystemFields	Indica si desea devolver los campos del sistema en el resultado. Los valores son <code>true</code> o <code>false</code> .
recordsToReturn	Especifica el número de filas que se van a devolver.
rejectInteractionId	ID de la interacción para rechazar cambios.
relationshipType	Lista de tipos de relación separados por comas. Por ejemplo, la lista <code>relationshipType=Employee,Employer</code> devuelve registros relacionados de los tipos de relación <code>Employee</code> y <code>Employer</code> .
returnTotal	Devuelve el número de registros en el conjunto de resultados. Se establece en <code>true</code> para obtener el número de registros de la lista de resultados.
rootId	ID de raíz de jerarquía.

Ejemplo de solicitud de API

La siguiente solicitud de ejemplo muestra las relaciones principales y secundarias directas para un registro de entidad de negocio:

```
GET http://localhost:8080/cm/cs/localhost-orcl-DS_UI1/hierarchy/ODI/entity/Organization/55/exportRelated
```

Ejemplo de respuesta de API

La siguiente respuesta de ejemplo muestra las relaciones principales y secundarias directas exportadas para un registro de la entidad de negocio en formato CSV:

```
Level,Business Entity,ID,Label,Parent ID,Parent Label,Type of Relationship to
Parent,issuingCompany,expirationYear,accountType,accountNumber,securityCode,expirationMon
th,cardholderName,productType,productNumber,productName,inceptionDate,productDesc,groupTy
pe,groupName,partyType,lastName,displayName,middleName,birthdate,firstName,dunsNumber
Root,Organization,584,Time Warner Inc,,,,,,,,,,,,,Organization,,Time Warner Inc,,,,
799527630
1,Organization,55,Historic TW Inc,584,Time Warner Inc,is ODI parent
of,,,,,,,,,,,,,Organization,,Historic TW Inc,,,,958466278
2,Organization,593,Turner Broadcasting System Inc,55,Historic TW Inc,is ODI parent
of,,,,,,,,,,,,,Organization,,Turner Broadcasting System Inc,,,,003319068
2,Organization,414,Historic TW Inc,55,Historic TW Inc,is ODI parent
of,,,,,,,,,,,,,Organization,,Historic TW Inc,,,,007305290
2,Organization,35,People Magazine,55,Historic TW Inc,is ODI parent
of,,,,,,,,,,,,,Organization,,People Magazine,,,,011584096
2,Organization,391,Historic TW Inc,55,Historic TW Inc,is ODI parent
of,,,,,,,,,,,,,Organization,,Historic TW Inc,,,,011816704
...
```

Obtener cambios de jerarquía

La API de obtención de cambios de jerarquía recupera los cambios pendientes en una jerarquía.

La API usa el método GET.

URL de solicitud

La dirección URL de obtención de cambios de jerarquía tiene el siguiente formato:

```
http://<host>:<port>/<context>/<database ID>/hierarchy/{hierarchy name}/changes
```

Haga la siguiente solicitud HTTP GET a la dirección URL de obtención de cambios de jerarquía:

```
GET http://<host>:<port>/<context>/<database ID>/hierarchy/{hierarchy name}/changes
```

Parámetros de consulta

Para filtrar las relaciones, puede anexar los parámetros de consulta a la URL de solicitud.

En la tabla siguiente, se describen los parámetros de la consulta:

Parámetro	Descripción
effectiveDate	La fecha para la cual desea recuperar las relaciones.
interactionId	ID de la interacción.
readLabelsOnly	Indica si desea ejecutar solo etiquetas en el resultado. Los valores son true o false.
readSystemFields	Indica si desea devolver los campos del sistema en el resultado. Los valores son true o false.
recordsToReturn	Especifica el número de filas que se van a devolver.
rejectInteractionId	ID de la interacción para rechazar cambios.

Parámetro	Descripción
relationshipType	Lista de tipos de relación separados por comas. Por ejemplo, la lista relationshipType=Employee,Employer devuelve registros relacionados de los tipos de relación Employee y Employer.
returnTotal	Devuelve el número de registros en el conjunto de resultados. Se establece en true para obtener el número de registros de la lista de resultados.
rootId	ID de raíz de jerarquía.

Ejemplo de solicitud de API

La siguiente solicitud devuelve cambios pendientes a una jerarquía filtrada por ID de interacción, ID raíz e ID de interacción de rechazo:

```
GET http://localhost:8080/cmx/bulk/localhost-orcl-DS_UI1/hierarchy/Product/changes?
interactionId=480003&rootId=165&rejectInteractionId=480004
```

Ejemplo de respuesta de API

La siguiente respuesta de ejemplo muestra los cambios pendientes:

```
{
  "interactionId": "480003",
  "rejectInteractionId": "480004",
  "type": "hierarchy",
  "name": "Product",
  "objects": [
    {
      "before": [
      ],
      "after": [
        {
          "businessEntity": {
            "ProductGroup": {
              "link": [
                {
                  "href": "http://localhost:8080/cmx/request/hm_icons/hierarchymanager/
Group/Group.png?ors=localhost-orcl-MDM_SAMPLE",
                  "rel": "icon"
                }
              ]
            },
            "rowidObject": "165",
            "creator": "admin",
            "createDate": "2020-01-13T13:07:35.128-05:00",
            "updatedBy": "admin",
            "lastUpdateDate": "2020-01-13T13:07:35.128-05:00",
            "consolidationInd": 4,
            "lastRowidSystem": "PRODUCT",
            "hubStateInd": 1,
            "label": "Vehicles",
            "productType": "Product Group",
            "productName": "Vehicles"
          }
        },
        {
          "entityLabel": "Vehicles",
          "entityType": "ProductGroup",
          "depth": 1,
          "object": {
            "businessEntity": {
              "ProductGroup": {
                "link": [

```



```

        "rel":"icon"
    }
    ],
    "rowidObject":"166",
    "creator":"admin",
    "createDate":"2020-01-13T13:07:35.250-05:00",
    "updatedBy":"admin",
    "lastUpdateDate":"2020-01-13T13:07:35.250-05:00",
    "consolidationInd":4,
    "lastRowidSystem":"PRODUCT",
    "hubStateInd":1,
    "label":"Scooter",
    "productType":"Product Group",
    "productName":"Scooter"
}
},
"entityLabel":"Scooter",
"relationshipLabel":"Product Group is parent of Product Group",
"relationship":{
    "rowidObject":"124",
    "creator":"datasteward",
    "createDate":"2020-01-13T13:07:35.864-05:00",
    "updatedBy":"datasteward",
    "lastUpdateDate":"2020-01-13T13:07:35.865-05:00",
    "consolidationInd":4,
    "lastRowidSystem":"PRODUCT",
    "interactionId":"480003",
    "hubStateInd":0,
    "label":"Product Group is parent of Product Group",
    "rowidRelType":"7",
    "rowidHierarchy":"3",
    "from":{
        "rowidObject":"165"
    },
    "to":{
        "rowidObject":"166"
    }
},
"entityType":"ProductGroup",
"relationshipType":"ProductGroupIsParentOfProductGroup",
"object":{
    "businessEntity":{
        "ProductGroup":{
            "link":{
                {
                    "href":"http://localhost:8080/cm/reqest/hm_icons/
hierarchymanager/Group/Group.png?ors=localhost-orcl-MDM_SAMPLE",
                    "rel":"icon"
                }
            },
            "rowidObject":"167",
            "label":"E-Scooter",
            "productType":"Product Group",
            "productName":"E-Scooter"
        }
    },
    "entityLabel":"E-Scooter",
    "relationshipLabel":"Product Group is parent of Product Group",
    "relationship":{
        "rowidObject":"125",
        "creator":"datasteward",
        "createDate":"2020-01-13T13:07:36.033-05:00",
        "updatedBy":"datasteward",
        "lastUpdateDate":"2020-01-13T13:07:36.034-05:00",
        "consolidationInd":4,
        "lastRowidSystem":"PRODUCT",
        "interactionId":"480003",
        "hubStateInd":0,
        "label":"Product Group is parent of Product Group",
        "rowidRelType":"7",
        "rowidHierarchy":"3"
    }
}

```



```

"interactionId":"480003",
"hubStateInd":0,
"label":"Product Group is parent of Product Group",
"rowidRelType":"7",
"rowidHierarchy":"3",
"from":{
  "rowidObject":"165"
},
"to":{
  "rowidObject":"166"
}
},
"entityType":"ProductGroup",
"relationshipType":"ProductGroupIsParentOfProductGroup",
"object":{
  "businessEntity":{
    "ProductGroup":{
      "link":[
        {
          "href":"http://localhost:8080/cm/req/est/hm_icons/
hierarchymanager/Group/Group.png?ors=localhost-orcl-MDM_SAMPLE",
          "rel":"icon"
        }
      ],
      "rowidObject":"167",
      "creator":"admin",
      "createDate":"2020-01-13T13:07:35.364-05:00",
      "updatedBy":"admin",
      "lastUpdateDate":"2020-01-13T13:07:35.364-05:00",
      "consolidationInd":4,
      "lastRowidSystem":"PRODUCT",
      "hubStateInd":1,
      "label":"E-Scooter",
      "productType":"Product Group",
      "productName":"E-Scooter"
    }
  },
  "entityLabel":"E-Scooter",
  "relationshipLabel":"Product Group is parent of Product Group",
  "relationship":{
    "rowidObject":"125",
    "creator":"datasteward",
    "createDate":"2020-01-13T13:07:36.033-05:00",
    "updatedBy":"datasteward",
    "lastUpdateDate":"2020-01-13T13:07:36.034-05:00",
    "consolidationInd":4,
    "lastRowidSystem":"PRODUCT",
    "interactionId":"480003",
    "hubStateInd":0,
    "label":"Product Group is parent of Product Group",
    "rowidRelType":"7",
    "rowidHierarchy":"3",
    "from":{
      "rowidObject":"166"
    },
    "to":{
      "rowidObject":"167"
    }
  },
  "entityType":"ProductGroup",
  "relationshipType":"ProductGroupIsParentOfProductGroup",
  "object":{
    "businessEntity":{
      "Products":{
        "link":[
          {
            "href":"http://localhost:8080/cm/req/est/HM_Icons/
BulletSquareBlue/bullet_square_blue.png?ors=localhost-orcl-MDM_SAMPLE",
            "rel":"icon"
          }
        ],
      }
    }
  },
  ],

```


Parámetros

Especifique el tipo de cambio que quiera realizar y la relación que quiera modificar.

En la tabla siguiente se describen los parámetros del cuerpo de la solicitud:

Parámetro	Descripción
operation	Tipo de cambio que quiera realizar. Los valores son createRelationship, updateRelationship y deleteRelationship.
relación	Nombre de la relación.
rowidObject	ID de fila del objeto.
startDate	Opcional. Especifica la fecha en la cual el registro entra en vigencia. Proporcione estos parámetros para un objeto base que tenga habilitada la línea temporal.
endDate	Opcional. Especifica la fecha en la cual el registro deja de estar en vigencia. Proporcione estos parámetros para un objeto base que tenga habilitada la línea temporal.

Ejemplo de solicitud de API

La siguiente solicitud crea nuevas relaciones:

```
POST http://localhost:8080/cmxbulk/localhost-orcl-DS_UI1
```

```
{
  "systemName": "Product",
  "rootRowId": "165",
  "items": [
    {
      "operation": "createRelationship",
      "relationship": "ProductGroupIsParentOfProductGroup",
      "payload": {
        "from": {
          "rowidObject": "165"
        },
        "to": {
          "rowidObject": "166"
        }
      }
    },
    {
      "operation": "createRelationship",
      "relationship": "ProductGroupIsParentOfProductGroup",
      "payload": {
        "from": {
          "rowidObject": "166"
        },
        "to": {
          "rowidObject": "167"
        }
      }
    },
    {
      "operation": "createRelationship",
      "relationship": "ProductGroupIsParentOfProduct",
      "payload": {
        "from": {
          "rowidObject": "167"
        },
        "to": {
          "rowidObject": "168"
        }
      }
    }
  ]
}
```

```

    },
    "effectivePeriod":{
      "startDate":"2019-01-01T11:11:53.974-04:00",
      "endDate":"2039-12-31T11:11:53.974-04:00"
    }
  }
]
}

```

Ejemplo de respuesta de API

La siguiente respuesta de ejemplo muestra el ID de interacción asociado con los cambios, que puede usarse para promover o rechazar los cambios:

```

{
  "processId":"14068",
  "interactionId":"480003",
  "rejectInteractionId":"480004",
  "items":[
    {
      "payload":{
        "key":{
          "rowid":"124",
          "sourceKey":"SVR1.1G9AG"
        },
        "rowidObject":"124",
        "from":{
          "key":{
            "rowid":"165"
          },
          "rowidObject":"165"
        },
        "to":{
          "key":{
            "rowid":"166"
          },
          "rowidObject":"166"
        }
      }
    },
    {
      "payload":{
        "key":{
          "rowid":"125",
          "sourceKey":"SVR1.1G9AH"
        },
        "rowidObject":"125",
        "from":{
          "key":{
            "rowid":"166"
          },
          "rowidObject":"166"
        },
        "to":{
          "key":{
            "rowid":"167"
          },
          "rowidObject":"167"
        }
      }
    },
    {
      "payload":{
        "key":{
          "rowid":"126",
          "sourceKey":"SVR1.1G9AI"
        },
        "rowidObject":"126",
        "from":{

```



```

        "key":{
            "rowid":"167"
        },
        "rowidObject":"167"
    },
    "to":{
        "key":{
            "rowid":"168"
        },
        "rowidObject":"168"
    }
}
]
}

```

Promoción masiva

La API de REST de promoción masiva promueve cambios en la red o jerarquía.

La API usa el método POST.

URL de solicitud

La dirección URL de promoción masiva tiene el siguiente formato:

```
http://<host>:<port>/<context>/<database ID>/promote
```

Haga la siguiente solicitud HTTP GET a la URL de promoción masiva:

```
POST http://<host>:<port>/<context>/<database ID>/promote
```

Ejemplo de solicitud de API

La siguiente solicitud de ejemplo promueve cambios:

```

POST http://localhost:8080/cmx/bulk/localhost-orcl-DS_UI1/promote
{
  "interactionId":"67080007",
  "rejectInteractionId":"67080008",
  "objects":[
    {
      "type":"Hierarchy",
      "name":"HPerson"
    }
  ]
}

```

Ejemplo de respuesta de API

La siguiente respuesta de ejemplo muestra la respuesta después de promover satisfactoriamente los cambios:

```

{
  "interactionId": "68520005",
  "rejectInteractionId": "68520006",
  "type": "hierarchy",
  "name": "HPerson",
  "objects": [
    {
      "before": [
        {
          "businessEntity": {
            "Person": {
              "link": [
                {

```

```

                "href": "http://localhost:8080/cmx/request/hm_icons/
hierarchymanager/Person/Person_Small.png?ors=localhost-mdm103-DS_UI2",
                "rel": "icon"
            }
        ],
        "rowidObject": "2568254",
        "creator": "admin",
        "createDate": "2020-04-01T14:38:06.577-04:00",
        "updatedBy": "admin",
        "lastUpdateDate": "2020-04-01T14:38:06.577-04:00",
        "consolidationInd": 4,
        "lastRowidSystem": "SFA",
        "hubStateInd": 1,
        "effectivePeriod": {},
        "label": "LN, FN",
        "partyType": "Person",
        "lastName": "LN",
        "displayName": "FN LN",
        "generationSuffixCd": "X",
        "firstName": "FN"
    }
},
"entityLabel": "LN, FN",
"entityType": "Person",
"depth": 0
}
],
"after": [
    {
        "businessEntity": {
            "Person": {
                "link": [
                    {
                        "href": "http://localhost:8080/cmx/request/hm_icons/
hierarchymanager/Person/Person_Small.png?ors=localhost-mdm103-DS_UI2",
                        "rel": "icon"
                    }
                ],
                "rowidObject": "2568253",
                "creator": "admin",
                "createDate": "2020-04-01T14:38:03.956-04:00",
                "updatedBy": "admin",
                "lastUpdateDate": "2020-04-01T14:38:03.956-04:00",
                "consolidationInd": 4,
                "lastRowidSystem": "SFA",
                "hubStateInd": 1,
                "label": "LN, FN",
                "partyType": "Person",
                "lastName": "LN",
                "displayName": "FN LN",
                "generationSuffixCd": "X",
                "firstName": "FN"
            }
        },
        "entityLabel": "Person",
        "entityType": "Person",
        "depth": 1,
        "object": {
            "businessEntity": {
                "Automobile": {
                    "rowidObject": "1450063",
                    "creator": "admin",
                    "createDate": "2020-04-01T14:38:09.630-04:00",
                    "updatedBy": "admin",
                    "lastUpdateDate": "2020-04-01T14:38:09.630-04:00",
                    "consolidationInd": 4,
                    "lastRowidSystem": "SFA",
                    "hubStateInd": 1,
                    "effectivePeriod": {},
                    "label": "Subaru",
                    "model": "MODEL",
                }
            }
        }
    }
]
}

```

```

        "drivetrainCode": "4WD",
        "doorsCode": "2DR",
        "make": "Subaru"
    },
    },
    "entityLabel": "Subaru",
    "relationshipLabel": "PersonToAutomobile Relationship",
    "relationship": {
        "rowidObject": "1450083",
        "creator": "ds1",
        "createDate": "2020-04-01T14:40:14.921-04:00",
        "updatedBy": "admin",
        "lastUpdateDate": "2020-04-01T14:40:23.812-04:00",
        "consolidationInd": 4,
        "lastRowidSystem": "SYS0",
        "interactionId": "68520005",
        "hubStateInd": 1,
        "label": "PersonToAutomobile Relationship",
        "from": {
            "rowidObject": "2568253"
        },
        "to": {
            "rowidObject": "1450063"
        }
    },
    "entityType": "Automobile",
    "relationshipType": "PersonToAutomobile2"
},
{
    "businessEntity": {
        "Person": {
            "link": [
                {
                    "href": "http://localhost:8080/cm/ request/hm_icons/
hierarchymanager/Person/Person_Small.png?ors=localhost-mdm103-DS_UI2",
                    "rel": "icon"
                }
            ],
            "rowidObject": "2568253",
            "creator": "admin",
            "createDate": "2020-04-01T14:38:03.956-04:00",
            "updatedBy": "admin",
            "lastUpdateDate": "2020-04-01T14:38:03.956-04:00",
            "consolidationInd": 4,
            "lastRowidSystem": "SFA",
            "hubStateInd": 1,
            "label": "LN, FN",
            "partyType": "Person",
            "lastName": "LN",
            "displayName": "FN LN",
            "generationSuffixCd": "X",
            "firstName": "FN"
        }
    },
    "entityLabel": "Person",
    "entityType": "Person",
    "depth": 1,
    "object": {
        "businessEntity": {
            "Person": {
                "link": [
                    {
                        "href": "http://localhost:8080/cm/ request/
hm_icons/hierarchymanager/Person/Person_Small.png?ors=localhost-mdm103-DS_UI2",
                        "rel": "icon"
                    }
                ],
                "rowidObject": "2568254",
                "creator": "admin",
                "createDate": "2020-04-01T14:38:06.577-04:00",
            }
        }
    }
}

```

```

        "updatedBy": "admin",
        "lastUpdateDate": "2020-04-01T14:38:06.577-04:00",
        "consolidationInd": 4,
        "lastRowidSystem": "SFA",
        "hubStateInd": 1,
        "effectivePeriod": {},
        "label": "LN, FN",
        "partyType": "Person",
        "lastName": "LN",
        "displayName": "FN LN",
        "generationSuffixCd": "X",
        "firstName": "FN"
    },
    "entityLabel": "LN, FN",
    "relationshipLabel": "Person associate Person",
    "relationship": {
        "rowidObject": "1634729",
        "creator": "ds1",
        "createDate": "2020-04-01T14:40:14.870-04:00",
        "updatedBy": "admin",
        "lastUpdateDate": "2020-04-01T14:40:23.872-04:00",
        "consolidationInd": 4,
        "lastRowidSystem": "SYS0",
        "interactionId": "68520005",
        "hubStateInd": 1,
        "label": "Person associate Person",
        "rowidRelType": "103",
        "rowidHierarchy": "1",
        "note": "note DS",
        "from": {
            "rowidObject": "2568253"
        },
        "to": {
            "rowidObject": "2568254"
        }
    },
    "entityType": "Person",
    "relationshipType": "PersonAssociatePerson2"
}
],
"approved": [
    {
        "businessEntity": {
            "Person": {
                "link": [
                    {
                        "href": "http://localhost:8080/cmx/request/hm_icons/hierarchymanager/Person/Person_Small.png?ors=localhost-mdm103-DS_UI2",
                        "rel": "icon"
                    }
                ],
                "rowidObject": "2568253",
                "creator": "admin",
                "createDate": "2020-04-01T14:38:03.956-04:00",
                "updatedBy": "admin",
                "lastUpdateDate": "2020-04-01T14:38:03.956-04:00",
                "consolidationInd": 4,
                "lastRowidSystem": "SFA",
                "hubStateInd": 1,
                "label": "LN, FN",
                "partyType": "Person",
                "lastName": "LN",
                "displayName": "FN LN",
                "generationSuffixCd": "X",
                "firstName": "FN"
            }
        },
        "entityLabel": "Person",
        "entityType": "Person",
    }
]

```

```

"depth": 1,
"object": {
  "businessEntity": {
    "Automobile": {
      "rowidObject": "1450063",
      "creator": "admin",
      "createDate": "2020-04-01T14:38:09.630-04:00",
      "updatedBy": "admin",
      "lastUpdateDate": "2020-04-01T14:38:09.630-04:00",
      "consolidationInd": 4,
      "lastRowidSystem": "SFA",
      "hubStateInd": 1,
      "effectivePeriod": {},
      "label": "Subaru",
      "model": "MODEL",
      "drivetrainCode": "4WD",
      "doorsCode": "2DR",
      "make": "Subaru"
    }
  },
  "entityLabel": "Subaru",
  "relationshipLabel": "PersonToAutomobile Relationship",
  "relationship": {
    "rowidObject": "1450083",
    "creator": "dsl",
    "createDate": "2020-04-01T14:40:14.921-04:00",
    "updatedBy": "dsl",
    "lastUpdateDate": "2020-04-01T14:40:14.922-04:00",
    "consolidationInd": 4,
    "lastRowidSystem": "SYS0",
    "interactionId": "68520005",
    "hubStateInd": 0,
    "label": "PersonToAutomobile Relationship",
    "from": {
      "rowidObject": "2568253"
    },
    "to": {
      "rowidObject": "1450063"
    }
  },
  "entityType": "Automobile",
  "relationshipType": "PersonToAutomobile2"
},
{
  "businessEntity": {
    "Person": {
      "link": [
        {
          "href": "http://localhost:8080/cmz/request/hm_icons/hierarchymanager/Person/Person_Small.png?ors=localhost-mdm103-DS_UI2",
          "rel": "icon"
        }
      ],
      "rowidObject": "2568253",
      "creator": "admin",
      "createDate": "2020-04-01T14:38:03.956-04:00",
      "updatedBy": "admin",
      "lastUpdateDate": "2020-04-01T14:38:03.956-04:00",
      "consolidationInd": 4,
      "lastRowidSystem": "SFA",
      "hubStateInd": 1,
      "label": "LN, FN",
      "partyType": "Person",
      "lastName": "LN",
      "displayName": "FN LN",
      "generationSuffixCd": "X",
      "firstName": "FN"
    }
  },
  "entityLabel": "Person",

```


URL de solicitud

La dirección URL de rechazo masivo tiene el siguiente formato:

```
http://<host>:<port>/<context>/<database ID>/reject
```

Haga la siguiente solicitud HTTP GET a la dirección URL de rechazo masivo:

```
POST http://<host>:<port>/<context>/<database ID>/reject
```

Ejemplo de solicitud de API

La siguiente solicitud de ejemplo rechaza los cambios:

```
POST http://localhost:8080/cmx/bulk/localhost-orcl-DS_UI1/reject
{
  "interactionId": "67080007",
  "rejectInteractionId": "67080008",
  "objects": [
    {
      "type": "Hierarchy",
      "name": "HPerson"
    }
  ]
}
```

Ejemplo de respuesta de API

El siguiente ejemplo muestra la respuesta después de rechazar satisfactoriamente los cambios:

```
{
  "interactionId": "68520003",
  "rejectInteractionId": "68520004",
  "type": "hierarchy",
  "name": "HPerson",
  "objects": [
    {
      "before": [
        {
          "businessEntity": {
            "Person": {
              "link": [
                {
                  "href": "http://localhost:8080/cmx/request/hm_icons/hierarchymanager/Person/Person_Small.png?ors=localhost-mdm103-DS_UI2",
                  "rel": "icon"
                }
              ],
              "rowidObject": "2568254",
              "creator": "admin",
              "createDate": "2020-04-01T14:38:06.577-04:00",
              "updatedBy": "admin",
              "lastUpdateDate": "2020-04-01T14:38:06.577-04:00",
              "consolidationInd": 4,
              "lastRowidSystem": "SFA",
              "hubStateInd": 1,
              "effectivePeriod": {},
              "label": "LN, FN",
              "partyType": "Person",
              "lastName": "LN",
              "displayName": "FN LN",
              "generationSuffixCd": "X",
              "firstName": "FN"
            }
          },
          "entityLabel": "LN, FN",
          "entityType": "Person",
          "depth": 0
        }
      ]
    }
  ]
}
```

```

],
"after": [
  {
    "businessEntity": {
      "Person": {
        "link": [
          {
            "href": "http://localhost:8080/cmx/request/hm_icons/
hierarchymanager/Person/Person_Small.png?ors=localhost-mdm103-DS_UI2",
            "rel": "icon"
          }
        ],
        "rowidObject": "2568254",
        "creator": "admin",
        "createDate": "2020-04-01T14:38:06.577-04:00",
        "updatedBy": "admin",
        "lastUpdateDate": "2020-04-01T14:38:06.577-04:00",
        "consolidationInd": 4,
        "lastRowidSystem": "SFA",
        "hubStateInd": 1,
        "effectivePeriod": {},
        "label": "LN, FN",
        "partyType": "Person",
        "lastName": "LN",
        "displayName": "FN LN",
        "generationSuffixCd": "X",
        "firstName": "FN"
      }
    },
    "entityLabel": "LN, FN",
    "entityType": "Person",
    "depth": 0
  }
],
"approved": [
  {
    "businessEntity": {
      "Person": {
        "link": [
          {
            "href": "/request/hm_icons/hierarchymanager/Person/
Person_Small.png?ors={ors}",
            "rel": "icon"
          }
        ],
        "rowidObject": "2568253",
        "creator": "admin",
        "createDate": "2020-04-01T14:38:03.956-04:00",
        "updatedBy": "admin",
        "lastUpdateDate": "2020-04-01T14:38:03.956-04:00",
        "consolidationInd": 4,
        "lastRowidSystem": "SFA",
        "hubStateInd": 1,
        "label": "LN, FN",
        "partyType": "Person",
        "lastName": "LN",
        "displayName": "FN LN",
        "generationSuffixCd": "X",
        "firstName": "FN"
      }
    },
    "entityLabel": "Person",
    "entityType": "Person",
    "depth": 1,
    "object": {
      "businessEntity": {
        "Automobile": {
          "rowidObject": "1450063",
          "creator": "admin",
          "createDate": "2020-04-01T14:38:09.630-04:00",
          "updatedBy": "admin",

```



```

        "lastUpdateDate": "2020-04-01T14:38:09.630-04:00",
        "consolidationInd": 4,
        "lastRowidSystem": "SFA",
        "hubStateInd": 1,
        "effectivePeriod": {},
        "label": "Subaru",
        "model": "MODEL",
        "drivetrainCode": "4WD",
        "doorsCode": "2DR",
        "make": "Subaru"
    },
    },
    "entityLabel": "Subaru",
    "relationshipLabel": "PersonToAutomobile Relationship",
    "relationship": {
        "rowidObject": "1450082",
        "creator": "dsl",
        "createDate": "2020-04-01T14:38:13.916-04:00",
        "updatedBy": "dsl",
        "lastUpdateDate": "2020-04-01T14:38:13.916-04:00",
        "consolidationInd": 4,
        "lastRowidSystem": "SYS0",
        "interactionId": "68520003",
        "hubStateInd": 0,
        "label": "PersonToAutomobile Relationship",
        "from": {
            "rowidObject": "2568253"
        },
        "to": {
            "rowidObject": "1450063"
        }
    },
    "entityType": "Automobile",
    "relationshipType": "PersonToAutomobile2"
},
{
    "businessEntity": {
        "Person": {
            "link": [
                {
                    "href": "/request/hm_icons/hierarchymanager/Person/Person_Small.png?ors={ors}",
                    "rel": "icon"
                }
            ],
            "rowidObject": "2568253",
            "creator": "admin",
            "createDate": "2020-04-01T14:38:03.956-04:00",
            "updatedBy": "admin",
            "lastUpdateDate": "2020-04-01T14:38:03.956-04:00",
            "consolidationInd": 4,
            "lastRowidSystem": "SFA",
            "hubStateInd": 1,
            "label": "LN, FN",
            "partyType": "Person",
            "lastName": "LN",
            "displayName": "FN LN",
            "generationSuffixCd": "X",
            "firstName": "FN"
        }
    },
    "entityLabel": "Person",
    "entityType": "Person",
    "depth": 1,
    "object": {
        "businessEntity": {
            "Person": {
                "link": [
                    {
                        "href": "/request/hm_icons/hierarchymanager/

```


Haga la siguiente solicitud HTTP POST a la URL de inicio de coincidencias:

```
POST http://<host>:<port>/adhocmatch/<orsId>/job
```

Parámetros

Especifique los parámetros para comenzar un trabajo de coincidencia de entidad de negocio.

En la siguiente tabla se describen los parámetros:

Parámetro	Tipo	Descripción
orsId	Ruta de acceso	ID de la base de datos del almacén de referencias operativas desde la que se recuperan las entidades de negocio coincidentes.
fileId	Cuerpo	Opcional. El ID del archivo que quiere importar.
mappingID	Cuerpo	Opcional. El ID de la asignación del archivo que quiere importar.
matchRuleSet	Cuerpo	Opcional. Especifique el conjunto de reglas de coincidencia por aplicar al trabajo de coincidencia de entidad de negocio.

Ejemplo de solicitud de API

La siguiente solicitud de ejemplo devuelve las unidades de negocio importadas después de realizar una coincidencia:

```
POST /cmx/jobcontrol/localhost-orcl-MDM_SAMPLE/execute/match HTTP/1.1
Content-Type: application/json
Host: localhost:8080
Content-Length: 64

{
  "tableName" : "C_PRODUCT",
  "matchRuleSetName" : "IDL"
}
```

Ejemplo de respuesta de API

La API devuelve un código de respuesta 200 OK después de la coincidencia de las entidades de negocio y de la importación satisfactoria de un archivo. El cuerpo de la respuesta está vacío.

Leer registros coincidentes

La API de REST para la lectura de registros coincidentes devuelve registros que coinciden con un registro raíz específico. Puede revisar la lista de registros para determinar qué registros puede fusionar con el registro raíz original. Con la API para fusionar registros podrá fusionar los registros.

La API utiliza el método GET.

URL de solicitud

La URL para la lectura de registros coincidentes tiene el siguiente formato:

```
http://<host>:<port>/<context>/<database ID>/<business entity>/<rowId>?action=matched
```

Haga la siguiente solicitud HTTP GET a la URL para la lectura de registros coincidentes:

```
GET http://<host>:<port>/<context>/<database ID>/<business entity>/<rowId>?action=matched
```

Cuerpo de la respuesta

El cuerpo de la respuesta contiene el número de registros que coinciden con el registro especificado, los detalles de los registros coincidentes y un token de búsqueda. Utilice el token de búsqueda para obtener las páginas posteriores del resultado de coincidencias.

Ejemplo de solicitud de API

La siguiente solicitud de ejemplo busca la entidad de negocio para los registros que coincidan con un registro:

```
GET http://localhost:8080/cmxcsllocalhost-orcl-DS_UI1/Person/1038245?action=matched
```

Ejemplo de respuesta de API

La siguiente respuesta de ejemplo muestra los detalles del registro que coincide con el registro especificado:

```
{
  "firstRecord": 1,
  "recordCount": 1,
  "pageSize": 10,
  "searchToken": "SVR1.AU5HE",
  "matchedEntity": [
    {
      "businessEntity": {
        "Person": {
          "link": [
            {
              "href": "http://localhost:8080/cmxcsllocalhost-orcl-DS_UI1/
Person/1038246",
              "rel": "self"
            }
          ],
          "rowidObject": "1038246",
          "creator": "admin",
          "createDate": "2008-08-12T02:15:02-07:00",
          "updatedBy": "Admin",
          "lastUpdateDate": "2008-08-12T02:59:17-07:00",
          "consolidationInd": "1",
          "lastRowidSystem": "SFA",
          "dirtyIndicator": "0",
          "hubStateInd": "1",
          "partyType": "Person",
          "lastName": "BATES",
          "firstName": "DAISY",
          "displayName": "DAISY BATES"
        }
      },
      "matchRule": "PUT"
    }
  ]
}
```

Actualizar registros coincidentes

La API de REST para la actualización de registros coincidentes crea o actualiza un registro en la tabla de coincidencias. La tabla de coincidencias contiene los pares de registros coincidentes de una entidad de

negocio después de ejecutar un proceso de coincidencia en la entidad de negocio. Utilice la API para agregar los registros que son aptos para una fusión con el registro especificado.

La API utiliza el método PUT.

URL de solicitud

La URL para la actualización de los registros coincidentes tiene el siguiente formato:

```
http://<host>:<port>/<context>/<database ID>/<business entity>/<rowId>?action=matched
```

Haga la siguiente solicitud HTTP PUT a la URL para la actualización de registros coincidentes:

```
PUT http://<host>:<port>/<context>/<database ID>/<business entity>/<rowId>?action=matched
```

Agregue el encabezado Content-Type para especificar el tipo de medio de los datos que desea enviar con la solicitud:

```
PUT http://<host>:<port>/<context>/<database ID>/<business entity>/<rowId>?action=matched
Content-Type: application/<json/xml>
```

Cuerpo de la solicitud

Envíe la lista de registros que coinciden con el registro especificado en el cuerpo de la solicitud. Utilice el ID de fila o el sistema de origen y la clave de fuente para especificar los registros.

Ejemplo de solicitud de API

El siguiente ejemplo agrega un registro en la tabla de coincidencias:

```
PUT http://localhost:8080/cmx/cs/localhost-ORCL-DS_UI1/Person/1038245?action=matched
{
  keys: [
    {
      rowid: "1038246"
    }
  ]
}
```

Ejemplo de respuesta de API

La API devuelve una respuesta de 200 OK al crear correctamente un registro en la tabla de coincidencias. El cuerpo de la respuesta está vacío.

Eliminar registros coincidentes

La API de REST para la eliminación de registros coincidentes elimina los registros coincidentes que estén asociados a un registro raíz en la tabla de coincidencia.

La API utiliza el método DELETE.

URL de solicitud

La URL para la eliminación de registros coincidentes tiene el siguiente formato:

```
http://<host>:<port>/<context>/<database ID>/<business entity>/<rowId>?action=matched
```

Haga la siguiente solicitud HTTP DELETE a la URL para la eliminación de registros coincidentes:

```
DELETE http://<host>:<port>/<context>/<database ID>/<business entity>/<rowId>?
action=matched
```

Agregue el encabezado Content-Type para especificar el tipo de medio de los datos que desea enviar con la solicitud:

```
DELETE http://<host>:<port>/<context>/<database ID>/<business entity>/<rowId>?
action=matched
Content-Type: application/<json/xml>
```

Cuerpo de la solicitud

Envíe la lista de registros que desea eliminar de la tabla de coincidencia en el cuerpo de la solicitud.

Ejemplo de solicitud de API

La siguiente solicitud de ejemplo elimina un registro que coincida con el registro raíz especificado de la tabla de coincidencia:

```
DELETE http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/Person/1038245?action=matched
{
  keys: [
    {
      rowid: "1038246"
    }
  ]
}
```

Ejemplo de respuesta de API

La API devuelve una respuesta de 200 OK al eliminar correctamente un registro de la tabla de coincidencias. El cuerpo de la respuesta está vacío.

Obtener datos de coincidencia en CSV

La API de REST de obtención de datos de coincidencia en CSV recupera entidades de negocio con o sin coincidencia en formato CSV después de que el sistema empareja registros del sistema de origen y los campos de destino.

La API usa el método GET.

URL de solicitud

La URL de obtención de datos de coincidencia tiene el siguiente formato:

```
http://<host>:<port>/adhocmatch/<orsId>/result/{jobGroupControlId}{?
withMatches,withoutMatches}
```

Haga la siguiente solicitud HTTP GET a la dirección URL de obtención de datos de coincidencia:

```
GET http://<host>:<port>/adhocmatch/<orsId>/result/{jobGroupControlId}{?
withMatches,withoutMatches}
```

Parámetros

Especifique los parámetros para devolver las entidades de negocio coincidentes o no coincidentes en formato CSV.

En la siguiente tabla se describen los parámetros:

Parámetro	Tipo	Descripción
jobGroupld	Ruta de acceso	ID de grupo del trabajo en lotes.
orsld	Ruta de acceso	ID de la base de datos del almacén de referencias operativas desde la que se recuperan registros de la entidad de negocio.
withMatches	Consulta	Opcional. Utilice el valor siguiente: True: para devolver registros que coincidan con datos existentes.
withoutMatches	Consulta	Opcional. Utilice el valor siguiente: False: para devolver registros que no coincidan con datos existentes.

Ejemplo de solicitud de API

La siguiente solicitud de ejemplo devuelve entidades de negocio coincidentes o no coincidentes en formato CSV:

```
GET /cmx/adhocmatch/{ors}/result/{jobId}?withMatches=true&withoutMatches=true
Accept: application/csv
```

```
{
  item: [
    {
      businessEntity: {
        Person: {
          firstName: "John"
        }
      },
      matchedEntity: {
        businessEntity: {Person: {rowidObject: 123}},
        matchRule: "MRS|1",
        matchScore: 0.9
      }
    },
    {
      businessEntity: {
        Person: {
          firstName: "Alex"
        }
      }
    }
  ]
}
```

Ejemplo de respuesta de API

La API devuelve un código de respuesta 200 OK y el archivo CSV con los resultados de la coincidencia.

Obtener datos de coincidencia en JSON

La API de REST de obtención de datos de coincidencia en JSON recupera entidades de negocio con o sin coincidencia en formato CSV después de que el sistema empareja registros del sistema de origen y los campos de destino.

La API usa el método GET.

URL de solicitud

La dirección URL de coincidencia de resultados tiene el siguiente formato:

```
http://<host>:<port>/adhocmatch/<orsId>/result/{jobGroupControlId}{?  
withMatches,withoutMatches}
```

Haga la siguiente solicitud HTTP GET a la dirección URL de coincidencia de resultados:

```
GET http://<host>:<port>/adhocmatch/<orsId>/result/{jobGroupControlId}{?  
withMatches,withoutMatches}
```

Parámetros

Especifique los parámetros para devolver las entidades de negocio coincidentes o no coincidentes en formato CSV.

En la siguiente tabla se describen los parámetros:

Parámetro	Tipo	Descripción
jobGroupId	Ruta de acceso	ID de grupo del trabajo en lotes.
orsId	Ruta de acceso	ID de la base de datos del almacén de referencias operativas desde la que se recuperan registros de la entidad de negocio.
withMatches	Consulta	Opcional. Utilice el valor siguiente: True: para devolver registros que coincidan con datos existentes.
withoutMatches	Consulta	Opcional. Utilice el valor siguiente: False: para devolver registros que no coincidan con datos existentes.

Ejemplo de solicitud de API

La siguiente solicitud de ejemplo devuelve la lista de entidades de negocio coincidentes o no coincidentes después de que el sistema realiza una coincidencia:

```
GET /cmx/adhocmatch/{ors}/result/{jobId}?withMatches=true&withoutMatches=true  
Accept: application/json
```

```
{  
  item: [  
    {  
      businessEntity: {  
        Person: {  
          firstName: "John"  
        }  
      },  
      matchedEntity: {
```



```

    {
      businessEntity: {Person: {rowidObject: 123}},
    }
  },
  {
    businessEntity: {
      Person: {
        firstName: "Alex"
      }
    }
  }
]
}

```

Ejemplo de respuesta de API

La API devuelve un código de respuesta 200 OK, y el archivo JSON con los resultados de la coincidencia.

Importar archivo coincidente

La API de REST de importación de archivos coincidentes hace coincidir entidades de negocio nuevas con entidades de negocio existentes antes de importar el archivo. Después de que se complete la operación de importación, las entidades de negocio coincidentes se actualizan con los datos nuevos.

Puede importar entidades de negocio duplicadas, entidades de negocio únicas o entidades de negocio únicas y duplicadas.

La API usa el método POST.

URL de solicitud

La dirección URL de importación de archivos con coincidencia tiene el siguiente formato:

```
http://<host>:<port>/<context>/<orsId>/<aftermatch>
```

Haga la siguiente solicitud HTTP POST a la dirección URL de importación de archivos con coincidencia:

```
POST http://<host>:<port>/<context>/<orsId>/<aftermatch>
```

Parámetros

Especifique los parámetros para hacer coincidir entidades de negocio nuevas con negocios existentes antes de importar el archivo.

En la siguiente tabla se describen los parámetros:

Parámetro	Tipo	Descripción
orsId	Ruta de acceso	ID de la base de datos del almacén de referencia operativa para importar el archivo después de la coincidencia con registros existentes.
filter	Consulta	Opcional. Utilice los siguientes valores de filtro: withMatches: introduzca <code>true</code> si los registros coinciden con registros existentes. withoutMatches: introduzca <code>false</code> si los registros no coinciden con registros existentes.

Parámetro	Tipo	Descripción
jobGroupControlId	Consulta	Opcional. ID de grupo de trabajos de coincidencia.
systemName	Consulta	Opcional. Nombre del sistema de origen.

Ejemplo de solicitud de API

La siguiente solicitud de ejemplo hace coincidir entidades de negocio nuevas con existentes antes de importar el archivo:

```
POST /cmx/beimport/{ors}/aftermatch
{
  jobGroupControlId: "...",
  systemName: "SFA",
  filter: {
    withMatches: true,
    withoutMatches: false
  }
}
```

Ejemplo de respuesta de API

La API devuelve un código de respuesta 200 OK después de la coincidencia exitosa de nuevas entidades de negocio con las existentes y de la importación del archivo. El cuerpo de la respuesta está vacío.

Obtener metadatos de sistema de origen

La API de REST de obtención de metadatos de sistema de origen devuelve los metadatos de los sistemas de origen.

La API usa el método GET.

URL de solicitud

La dirección URL de obtención de metadatos del sistema de origen tiene el siguiente formato:

```
http://<host>:<port>/<context>/<orsId>/sourceSystem
```

Haga la siguiente solicitud HTTP GET a la dirección URL de obtención de metadatos del sistema de origen:

```
GET http://<host>:<port>/<context>/<orsId>/sourceSystem
```

Parámetros

Especifique el sistema de origen que quiere devolver en la respuesta.

En la tabla siguiente se describen los parámetros del cuerpo de la solicitud:

Parámetro	Tipo	Descripción
orsId	Ruta de acceso	ID de almacén de datos de referencia operativa de la base de datos desde la cual recuperar el sistema de origen.
entityId	Ruta de acceso	ID de la entidad de negocio.

Parámetro	Tipo	Descripción
entityName	Ruta de acceso	Nombre de la entidad de negocio.
action	Consulta	La acción que quiere que realice la API. El siguiente es el valor predeterminado: getSourceSystems
aggregate	Consulta	Opcional. Datos agregados entre varios XREF. Use los siguientes valores: - RECORD - NODE - ENTITY
children	Consulta	Opcional. Lista de nombres de nodo secundario. Si se especifica, la respuesta contiene los nombres de los nodos secundarios.
compact	Consulta	Opcional. Especifique para devolver un resultado compacto.
contentMetadata	Consulta	Opcional. Metadatos de la lista de eventos del historial. Use los siguientes valores: XREF PENDING_XREF DELETED_XREF HISTORY MATCH BVT TRUST
contributorsOnly	Consulta	Opcional. Cuenta todos los XREF o aquellos que contribuyen con la mejor versión de confianza (BVT).
depth	Consulta	Opcional. El número de niveles secundarios que se devolverán.
describe	Consulta	Opcional. Agrega la descripción del sistema de origen.
fields	Consulta	Opcional. Lista de campos de entidad de negocio separados por comas. Si se especifica, la respuesta solo contiene los campos indicados.
filter	Consulta	Opcional. Expresión de filtro.
firstRecord	Consulta	Opcional. Especifica la primera fila del resultado.
order	Consulta	Opcional. Lista de nombres de campo separados por comas con el prefijo opcional + o -. El prefijo + indica la ordenación ascendente de los resultados y el prefijo - indica la ordenación descendente de los resultados. El valor predeterminado es +. Si especifica más de un parámetro, el conjunto de resultados se ordena por el primer parámetro de la lista, seguido del segundo.

Parámetro	Tipo	Descripción
recordStates	Consulta	Opcional. Estados de registro devueltos en la lista de eventos del historial. Proporcione una lista de valores separados por comas. Puede usar los siguientes valores: - ACTIVE - PENDING - DELETED
recordstoReturn	Consulta	Opcional. Especifica el número de filas que se van a devolver.
recordsTotal	Consulta	Opcional. Si se establece en <code>true</code> , devuelve el número de registros en el resultado. El valor predeterminado es <code>false</code> .

Ejemplo de solicitud de API

La siguiente solicitud de ejemplo devuelve los metadatos del sistema de origen disponibles:

```
GET /cmx/metadata/localhost-orcl-MDM_SAMPLE/sourceSystem HTTP/1.1
Host: localhost:8080
```

Ejemplo de respuesta de API

La siguiente respuesta de ejemplo devuelve los metadatos del sistema de origen disponibles:

```
HTTP/1.1 200 OK
Server: JBoss-EAP/7
X-Powered-By: Undertow/1
Content-Type: application/json;charset=UTF-8
Content-Length: 328
```

```
{
  "totalCount" : 7,
  "firstRecord" : 1,
  "pageSize" : 7,
  "sourceSystems" : [ {
    "name" : "Product"
  }, {
    "name" : "Informatica Data Director"
  }, {
    "name" : "Legacy"
  }, {
    "name" : "SFA"
  }, {
    "name" : "Lookups"
  }, {
    "name" : "OrgDataInc"
  }, {
    "name" : "Admin"
  } ]
}
```

Obtener eventos de historial de registros

La API de REST de obtención de eventos de historial de registros devuelve una lista de eventos de historial o grupos de eventos de historial asociados a un registro. Envíe el ID del registro en el cuerpo de la solicitud.

La API usa el método GET para devolver los siguientes datos para cada grupo de eventos de historial:

- Fechas de inicio y finalización del grupo

- Número de eventos del grupo

La API devuelve los siguientes datos para cada evento de historial:

- ID del evento de historial
- Fecha del cambio
- Usuario que ha realizado el cambio
- Lista de tablas de historial afectadas por el cambio
- Lista de nodos de registro afectados por el cambio

URL de solicitud

La dirección URL de obtención de eventos de historial de registros tiene el siguiente formato:

```
http://<host>:<port>/<context>/<database ID>/<business entity>/<rowId>?
action=listHistoryEvents
```

Haga la siguiente solicitud HTTP GET a la dirección URL de obtención de eventos de historial de registros:

```
GET http://<host>:<port>/<context>/<database ID>/<business entity>/<rowId>?
action=listHistoryEvents
```

Parámetros de consulta

El ID del registro es un parámetro obligatorio. La API usa el ID de registro para buscar todos los eventos de historial relacionados.

En la tabla siguiente se muestran los parámetros de la consulta:

Parámetro	Descripción
startDate y endDate	Opcional. Intervalo de fechas para el que desea recuperar los datos. Si especifica un intervalo de fechas, la respuesta contiene solo eventos de este intervalo.
granularity	Opcional. Nivel de detalle para agrupar eventos de historial. Si se especifica, la respuesta agrupa los eventos de historial. En caso contrario, la respuesta no agrupa los eventos de historial. Use uno de los siguientes valores: <ul style="list-style-type: none"> - YEAR - QUARTER - MONTH - WEEK - DAY - HOUR - MINUTE - AUTO
recordStates	Opcional. Estados de registro devueltos en la lista de eventos del historial. Proporcione una lista de valores separados por comas. Puede usar los siguientes valores: <ul style="list-style-type: none"> - ACTIVE - PENDING - DELETED

Parámetro	Descripción
contentMetadata	Opcional. Metadatos para la lista de eventos del historial. Proporcione una lista de valores separados por comas. Puede usar los siguientes valores: <ul style="list-style-type: none"> - XREF - PENDING_XREF - DELETED_XREF - HISTORY - MATCH - BVT - TRUST
children	Opcional. Lista separada por comas de los nombres de nodos secundarios. Si se especifica, la respuesta contiene los nombres de los nodos secundarios.
order	Opcional. Lista de nombres de campo separados por comas con el prefijo opcional + o -. El prefijo + indica la ordenación ascendente de los resultados y el prefijo - indica la ordenación descendente de los resultados. El valor predeterminado es +. Si especifica más de un parámetro, el conjunto de resultados se ordena por el primer parámetro de la lista, seguido del segundo.
fields	Opcional. Lista de campos de entidad de negocio separados por comas. Si se especifica, la respuesta solo contiene los campos indicados.
filter	Opcional. Expresión de filtro.
depth	Opcional. El número de niveles secundarios que se devolverán.
recordsToReturn	Opcional. Especifica el número de filas que se van a devolver.
searchToken	Opcional. Especifica el token de búsqueda que se devuelve con la solicitud anterior.
returnTotal	Opcional. Si se establece en <code>true</code> , devuelve el número de registros en el resultado. El valor predeterminado es <code>false</code> .
firstRecord	Opcional. Especifica la primera fila del resultado.
changeType	Opcional. Especifica los tipos de cambio devueltos en el resultado. Proporcione una lista de valores separados por comas. Puede usar los siguientes valores: <ul style="list-style-type: none"> - BO - XREF - BVT - MERGE - MERGE_AS_SOURCE - MERGE_AS_TARGET - UNMERGE_AS_SOURCE - UNMERGE_AS_TARGET

TEMAS RELACIONADOS

- [“Formatos de fecha y hora UTC” en la página 36](#)

Ejemplo de solicitud de API

La siguiente solicitud de ejemplo devuelve todas las fusiones, agrupadas por año, para un registro desde el 1 de enero de 2000:

```
GET http://localhost:8080/cm/cs/localhost-orcl-DS_UI1/Person/123?
action=listHistoryEvents&startDate=2010-01-01&granularity=YEAR&depth=2&changeType=MERGE
```

Ejemplo de respuesta de API

La siguiente respuesta de ejemplo indica las fusiones para el registro especificado desde el 1 de enero de 2000:

```
{
  firstRecord: 1,
  recordCount: 2
  item: [
    {
      link: [ // you can use links directly to get event list
        {rel: "events", href: "/Person/123?
action=listHistoryEvents&startDate=2000-01-01&endDate=2001-01-01&depth=2&changeType=MERGE
"}
      ],
      startDate: "2000-01-01",
      endDate: "2001-01-01",
      eventCount: 123
    },
    // no events in 2001, 2002, ... 2009
    {
      link: [
        {rel: "events", href: "/Person/123?
action=listHistoryEvents&startDate=2010-01-01&endDate=2011-01-01&depth=2&changeType=MERGE
"}
      ],
      startDate: "2010-01-01",
      endDate: "2011-01-01",
      eventCount: 23
    }
    // no events in 2011, ..., 2016
  ]
}
```

Obtener detalles de evento

La API de REST de obtención de detalles de evento devuelve los detalles de un evento de historial específico asociado a un registro. La API devuelve detalles como el tipo de cambio realizado y los valores antes y después del cambio. Envíe el ID de registro y el ID de evento de historial en el cuerpo de la solicitud.

La API usa el método GET.

URL de solicitud

La dirección URL de obtención de detalles de evento tiene el siguiente formato:

```
http://<host>:<port>/<context>/<database ID>/<business entity>/<rowId>?
action=getHistoryEventDetails
```

Use la siguiente solicitud HTTP GET para la dirección URL de obtención de detalles de evento:

```
GET http://<host>:<port>/<context>/<database ID>/<business entity>/<rowId>?
action=getHistoryEventDetails
```

Parámetros de consulta

Antes de empezar, use la API de obtención de eventos de historial de registros para indicar los eventos del historial que están asociados a un registro. En los resultados que se devuelven, use el ID de evento de historial y el ID de registro como parámetros de consulta.

En la tabla siguiente se muestran los parámetros de la consulta:

Parámetro	Descripción
eventID	Obligatorio. La API de obtención de eventos de historial de registros devuelve los ID de evento de los eventos del historial.
recordStates	Opcional. Estados de registro devueltos en la lista de eventos del historial. Proporcione una lista de valores separados por comas. Puede usar los siguientes valores: <ul style="list-style-type: none">- ACTIVE- PENDING- DELETED
contentMetadata	Opcional. Metadatos para la lista de eventos del historial. Proporcione una lista de valores separados por comas. Puede usar los siguientes valores: <ul style="list-style-type: none">- XREF- PENDING_XREF- DELETED_XREF- HISTORY- MATCH- BVT- TRUST
children	Opcional. Lista separada por comas de los nombres de nodos secundarios. Si se especifica, la respuesta contiene los nombres de los nodos secundarios.
order	Opcional. Lista de nombres de campo separados por comas con el prefijo opcional + o -. El prefijo + indica la ordenación ascendente de los resultados y el prefijo - indica la ordenación descendente de los resultados. El valor predeterminado es +. Si especifica más de un parámetro, el conjunto de resultados se ordena por el primer parámetro de la lista, seguido del segundo.
fields	Opcional. Lista de campos de entidad de negocio separados por comas. Si se especifica, la respuesta solo contiene los campos indicados.
filter	Opcional. Expresión de filtro.
depth	Opcional. El número de niveles secundarios que se devolverán.
recordsToReturn	Opcional. Especifica el número de filas que se van a devolver.
searchToken	Opcional. Especifica el token de búsqueda que se devuelve con la solicitud anterior.
returnTotal	Opcional. Si se establece en <code>true</code> , devuelve el número de registros en el resultado. El valor predeterminado es <code>false</code> .
firstRecord	Opcional. Especifica la primera fila del resultado.

Ejemplo de solicitud de API

En la siguiente solicitud de ejemplo se devuelve información para un evento de historial:

```
GET http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/Person/123?
action=getHistoryEventDetails&eventId=2016-07-14T02%3A01%3A24.529%2B0000
```

Ejemplo de respuesta de API

En la siguiente respuesta de ejemplo se muestran los detalles del evento especificado:

```
{
  "eventId": "2016-07-14T02:01:24.529+0000",
  "eventDate": "2016-07-14T02:01:24.529Z",
  "user": "admin",
  "changeType": [
    "BVT",
    "BO",
    "UNMERGE_AS_TARGET"
  ],
  "businessEntity": {
    "Person": {
      "rowidObject": "438243",
      "creator": "datasteward1",
      "createDate": "2016-07-08T20:42:47.402Z",
      "updatedBy": "admin",
      "lastUpdateDate": "2016-07-14T01:42:50.841Z",
      "consolidationInd": 1,
      "lastRowidSystem": "SYS0",
      "hubStateInd": 1,
      "label": "BE,AC",
      "partyType": "Person",
      "lastName": "BE",
      "displayName": "AC BE",
      "firstName": "AC"
    }
  }
}
```

Lista Informes

La API de REST de enumeración de informes devuelve una lista de informes registrados y su configuración.

La API usa el método GET.

URL de solicitud

La dirección URL de REST de enumeración de informes tiene el siguiente formato:

```
http://<host>:<port>/<context>/<database ID>/list
```

Haga la siguiente solicitud HTTP GET a la URL:

```
GET http://<host>:<port>/<context>/<database ID>/list
```

Parámetros de consulta

Puede usar el parámetro `show` para especificar si quiere enumerar todos los informes o solo los informes raíz.

Para devolver todos los informes, incluso raíz y de detalle, use el siguiente parámetro de consulta:

```
list?show=all
```

Para devolver solo los informes raíz, use el siguiente parámetro de consulta:

```
list?show=root
```

Nota: Si no especifica el parámetro `show`, la API devuelve informes raíz.

Cuerpo de la respuesta

El cuerpo de la respuesta contiene la lista de informes, incluso los detalles de cada uno de ellos.

En la siguiente tabla se describen los parámetros de configuración del informe en el cuerpo de la respuesta:

Parámetro	Descripción
ROWID_RPT_CONFIG	ID de un informe.
DIMENSION_NAME_1	El nombre de una dimensión de datos en el informe. Por ejemplo, una dimensión puede ser nombre de usuario, tipo de tarea de revisión o tipo de entidad de negocio.
DIMENSION_NAME_2	Opcional. El nombre de una dimensión de datos en el informe. Por ejemplo, una dimensión puede ser nombre de usuario, tipo de tarea de revisión o tipo de entidad de negocio.
TIMEPERIOD_NAME	Opcional. Nombre del período de tiempo al que se aplican los datos. Por ejemplo, puede especificar <code>Mes</code> .
RPT_NAME	Nombre del informe.
METRIC_NAME	Etiqueta del tipo de datos recabados por el informe. De forma predeterminada, el valor especificado para este parámetro aparece como el nombre del eje y en el cuadro. Por ejemplo, puede tener un informe que recaba información acerca de tareas cerradas por el usuario. Puede usar el nombre del indicador <code>Número de tareas cerradas</code> .
RPT_DESC	Descripción del informe.
RPT_TYPE	Indica si el informe es un informe de detalle. Si el valor es <code>null</code> , el informe es un informe raíz. Si el valor no es <code>null</code> , el informe es un informe de detalle.

Ejemplo de solicitud de API

La siguiente solicitud de ejemplo devuelve todos los informes registrados, incluso los informes raíz e informes de detalle:

```
GET http://localhost:8080/cmz/report/localhost-orcl-DS_UI1/list?show=all
```

Ejemplo de respuesta de API

La siguiente respuesta de ejemplo muestra una lista de los informes registrados:

```
{
  "ROWID_RPT_CONFIG": "7 ",
  "DIMENSION_NAME_1": "Country",
  "DIMENSION_NAME_2": "Users",
  "TIMEPERIOD_NAME": "Month",
  "RPT_NAME": "Top 10 Countries",
  "METRIC_NAME": "Number Of Customers",
  "RPT_DESC": "Top 10 Countries",
  "RPT_TYPE": "null"
},
{
  "metadata": {
    "ROWID_RPT_CONFIG": "12 ",
    "DIMENSION_NAME_1": "City code",
    "DIMENSION_NAME_2": "null",
    "TIMEPERIOD_NAME": "null",
  }
}
```

```

    "RPT_NAME": "Customer by City",
    "METRIC_NAME": "Customer by City",
    "RPT_DESC": "Customer by City",
    "RPT_TYPE": "null"
}

```

Obtener configuración y datos del informe

La API de REST de obtención de configuración y datos del informe devuelve la configuración del informe y sus datos.

La API usa el método GET.

URL de solicitud

La dirección URL de REST de obtención de configuración del informe y datos tiene el siguiente formato:

```
http://<host>:<port>/<context>/<database ID>/data/<report ID>
```

Haga la siguiente solicitud HTTP GET a la URL:

```
GET http://<host>:<port>/<context>/<database ID>/data/<reportId>
```

Parámetros de consulta

El ID de informe es un parámetro de URL obligatorio. Use el ID de informe para especificar el informe para el que quiera la configuración y los datos.

Cuerpo de la respuesta

La respuesta incluye los datos y la configuración del informe.

En la tabla siguiente se describen los parámetros en el cuerpo de la respuesta:

Parámetro	Descripción
metadatos	Incluye la configuración del informe.
ROWID_RPT_CONFIG	ID de un informe.
DIMENSION_NAME_1	El nombre de una dimensión de datos en el informe. Por ejemplo, una dimensión puede ser nombre de usuario, tipo de tarea de revisión o tipo de entidad de negocio.
DIMENSION_NAME_2	Opcional. El nombre de una dimensión de datos en el informe. Por ejemplo, una dimensión puede ser nombre de usuario, tipo de tarea de revisión o tipo de entidad de negocio.
TIMEPERIOD_NAME	Opcional. Nombre del período de tiempo al que se aplican los datos. Por ejemplo, puede especificar <i>Mes</i> .
RPT_NAME	Nombre del informe.
METRIC_NAME	Etiqueta del tipo de datos recabados por el informe. De forma predeterminada, el valor especificado para este parámetro aparece como el nombre del eje y en el cuadro. Por ejemplo, puede tener un informe que recaba información acerca de tareas cerradas por el usuario. Puede usar el nombre del indicador <i>Número de tareas cerradas</i> .
RPT_DESC	Descripción del informe.

Parámetro	Descripción
RPT_TYPE	Indica si el informe es un informe de detalle. Si el valor es <code>null</code> , el informe es un informe raíz. Si el valor no es <code>null</code> , el informe es un informe de detalle.
fieldsMetadata	Enumera los campos en una matriz de datos del informe. Los valores incluyen <code>DIMENSION_VALUE_1</code> , <code>DIMENSION_VALUE_2</code> , <code>TIMEPERIOD_VALUE</code> , <code>METRIC_VALUE</code> y <code>DRILLDOWN_RPT_ID</code> .
datos	Enumera los valores de cada campo en una matriz de datos del informe.

Ejemplo de solicitud de API

La siguiente solicitud de ejemplo devuelve la configuración del informe y los datos para un informe:

```
GET http://localhost:8080/cmz/report/localhost-orcl-DS_UI1/data/MDM.RPT.1
```

Ejemplo de respuesta de API

La siguiente respuesta de ejemplo muestra la configuración del informe y los datos para un informe:

```
{
  "metadata": {
    "ROWID_RPT_CONFIG": "11 ",
    "DIMENSION_NAME_1": "Country code",
    "DIMENSION_NAME_2": "null",
    "TIMEPERIOD_NAME": "null",
    "RPT_NAME": "Customer by Region",
    "METRIC_NAME": "Customer by Region",
    "RPT_DESC": "Customer by Region",
    "RPT_TYPE": "null",
    "fieldsMetadata": ["DIMENSION_VALUE_1", "DIMENSION_VALUE_2", "TIMEPERIOD_VALUE",
"METRIC_VALUE", "DRILLDOWN_RPT_ID"]
  },
  "data": [
    [
      "CA ",
      "null",
      "null",
      "9",
      "123"
    ],
    [
      "IN ",
      "null",
      "null",
      "19",
      "456"
    ],
    [
      "US ",
      "United States",
      "null",
      "9",
      "678"
    ],
    [
      "IN ",
      "India ",
      "null",
      "19",
      null
    ]
  ]
}
```

Obtener configuración del informe e informes de detalle

La API de REST de obtención de configuración del informe e informes de detalle devuelve la configuración del informe e informes de detalle asociados.

La API usa el método GET.

URL de solicitud

La dirección URL de REST de obtención de configuración del informe e informes de detalle tiene le siguiente formato:

```
http://<host>:<port>/<context>/<database ID>/meta/<report ID>
```

Haga la siguiente solicitud HTTP GET a la URL:

```
GET http://<host>:<port>/<context>/<database ID>/meta/<report ID>
```

Parámetros de consulta

El ID de informe es un parámetro de URL obligatorio. Use el ID de informe para especificar el informe para el que quiera los informes de detalle asociados y la configuración.

Cuerpo de la respuesta

El cuerpo de la respuesta incluye los siguientes parámetros:

Parámetro	Descripción
ROWID_RPT_CONFIG	ID de un informe.
DIMENSION_NAME_1	El nombre de una dimensión de datos en el informe. Por ejemplo, una dimensión puede ser nombre de usuario, tipo de tarea de revisión o tipo de entidad de negocio.
DIMENSION_NAME_2	Opcional. El nombre de una dimensión de datos en el informe. Por ejemplo, una dimensión puede ser nombre de usuario, tipo de tarea de revisión o tipo de entidad de negocio.
TIMEPERIOD_NAME	Opcional. Nombre del período de tiempo al que se aplican los datos. Por ejemplo, puede especificar <i>Mes</i> .
RPT_NAME	Nombre del informe.
METRIC_NAME	Etiqueta del tipo de datos recabados por el informe. De forma predeterminada, el valor especificado para este parámetro aparece como el nombre del eje y en el cuadro. Por ejemplo, puede tener un informe que recaba información acerca de tareas cerradas por el usuario. Puede usar el nombre del indicador <i>Número de tareas cerradas</i> .
RPT_DESC	Descripción del informe.
RPT_TYPE	Indica si el informe es un informe de detalle. Si el valor es <i>null</i> , el informe es un informe raíz. Si el valor no es <i>null</i> , el informe es un informe de detalle.
DRILLDOWN	Enumera los informes de detalle. El orden de los informes enumerados refleja el nivel del informe en relación con el informe raíz.
RPT_NAME	Nombre del informe de detalle.

Parámetro	Descripción
DETAILED_RPT_IDS	Enumera los ID de los detalles del informe. Incluye los ID de detalle del informe de los informes de detalle para el nivel de detalle.
CONFIG_RPT_IDS	Enumera los ID del informe de detalles para el nivel de detalle.

Ejemplo de solicitud de API

La siguiente solicitud de ejemplo devuelve la configuración del informe e informes de detalle asociados:

```
GET http://localhost:8080/cmz/report/localhost-orcl-DS_UI1/meta/SVR1.2X9N0
```

Ejemplo de respuesta de API

La siguiente respuesta de ejemplo muestra la configuración del informe e informes de detalle asociados:

```
{
  "ROWID_RPT_CONFIG": "7 ",
  "DIMENSION_NAME_1": "Country",
  "DIMENSION_NAME_2": "Users",
  "TIMEPERIOD_NAME": "Month",
  "RPT_NAME": "Top 10 Countries",
  "METRIC_NAME": "Number Of Customers",
  "RPT_DESC": "Top 10 Countries",
  "RPT_TYPE": "null",
  "DRILLDOWN":
  [
    {
      "RPT_NAME": "Customer by region - First level drilldown",
      "DETAILED_RPT_IDS":["1", "2", "3"],
      "CONFIG_RPT_IDS":["1"]
    },
    {
      "RPT_NAME": "Customer by city - Second level drilldown",
      "DETAILED_RPT_IDS":["5", "6", "7", "8", "9", "10"],
      "CONFIG_RPT_IDS":["2"]
    }
  ]
}
```

Registrar informe

La API de REST para registro de informes registra informes personalizados. La API devuelve el ID del informe, que se puede usar en otras API.

La API usa el método POST.

Para obtener información acerca de cómo registrar los informes de fábrica, consulte [“Administrar informes de fábrica” en la página 355](#).

URL de solicitud

La dirección URL de REST de registro de informe tiene el siguiente formato:

```
http://<host>:<port>/<context>/<database ID>/list
```

Haga la siguiente solicitud HTTP POST a la dirección URL:

```
POST http://<host>:<port>/<context>/<database ID>/list
```

Cuerpo de la solicitud

Especifique la configuración del informe.

En la tabla siguiente, se describen los parámetros de los detalles del informe en el cuerpo de la solicitud:

Parámetro	Descripción
ROWID_RPT_CONFIG	ID de un informe.
DIMENSION_NAME_1	El nombre de una dimensión de datos en el informe. Por ejemplo, una dimensión puede ser nombre de usuario, tipo de tarea de revisión o tipo de entidad de negocio.
DIMENSION_NAME_2	Opcional. El nombre de una dimensión de datos en el informe. Por ejemplo, una dimensión puede ser nombre de usuario, tipo de tarea de revisión o tipo de entidad de negocio.
TIMEPERIOD_NAME	Opcional. Nombre del período de tiempo al que se aplican los datos. Por ejemplo, puede especificar <i>Mes</i> .
RPT_NAME	Nombre del informe.
METRIC_NAME	Etiqueta del tipo de datos recabados por el informe. De forma predeterminada, el valor especificado para este parámetro aparece como el nombre del eje y en el cuadro. Por ejemplo, puede tener un informe que recaba información acerca de tareas cerradas por el usuario. Puede usar el nombre del indicador <i>Número de tareas cerradas</i> .
RPT_DESC	Descripción del informe.
RPT_TYPE	Indica si el informe es un informe de detalle. Si el valor es <code>null</code> , el informe es un informe raíz. Si el valor no es <code>null</code> , el informe es un informe de detalle.

Ejemplo de solicitud de API

La siguiente solicitud de ejemplo registra un informe:

```
POST http://localhost:8080/cm/report/localhost-orcl-DS_UI1/list
{
  "DIMENSION_NAME_1": "Country",
  "DIMENSION_NAME_2": "Users",
  "TIMEPERIOD_NAME": "null",
  "RPT_NAME": "Top 10 Countries",
  "METRIC_NAME": "Number Of Customers",
  "RPT_DESC": "Top 10 Countries"
}
```

Ejemplo de respuesta de API

El siguiente ejemplo muestra la respuesta al registrar satisfactoriamente un informe:

```
{
  "ROWID_RPT_CONFIG": "SVR1.2X9N0",
  "DIMENSION_NAME_1": "Subject area",
  "DIMENSION_NAME_2": "Source System",
  "TIMEPERIOD_NAME": "Month",
  "RPT_NAME": "randomname750",
  "RPT_DESC": "Source system Metrics",
  "METRIC_NAME": "Number Of Records",
  "RPT_TYPE": null
}
```

La API devuelve el ID del informe en el parámetro ROWID_RPT_CONFIG.

Actualizar la configuración del informe

La API de REST de actualización de la configuración de informes actualiza la configuración de un informe.

La API usa el método POST.

URL de solicitud

La dirección URL de REST de actualización de la configuración del informe tiene el siguiente formato:

```
http://<host>:<port>/<context>/<database ID>/list/<report ID>
```

Haga la siguiente solicitud HTTP POST a la dirección URL:

```
POST http://<host>:<port>/<context>/<database ID>/list/<report ID>
```

Parámetros de consulta

El ID de informe es un parámetro de URL obligatorio. Use el ID de informe para especificar el informe para el que quiera actualizar la configuración.

Cuerpo de la solicitud

Especifique la configuración del informe que quiera actualizar.

En la tabla siguiente se describen los parámetros del cuerpo de la solicitud:

Parámetro	Descripción
DIMENSION_NAME_1	El nombre de una dimensión de datos en el informe. Por ejemplo, una dimensión puede ser nombre de usuario, tipo de tarea de revisión o tipo de entidad de negocio.
DIMENSION_NAME_2	Opcional. El nombre de una dimensión de datos en el informe. Por ejemplo, una dimensión puede ser nombre de usuario, tipo de tarea de revisión o tipo de entidad de negocio.
TIMEPERIOD_NAME	Opcional. Nombre del período de tiempo al que se aplican los datos. Por ejemplo, puede especificar <i>Mes</i> .
RPT_NAME	Nombre del informe.
METRIC_NAME	Etiqueta del tipo de datos recabados por el informe. De forma predeterminada, el valor especificado para este parámetro aparece como el nombre del eje y en el cuadro. Por ejemplo, puede tener un informe que recaba información acerca de tareas cerradas por el usuario. Puede usar el nombre del indicador <i>Número de tareas cerradas</i> .
RPT_DESC	Descripción del informe.

Ejemplo de solicitud de API

La siguiente solicitud de ejemplo actualiza la configuración de un informe:

```
POST http://localhost:8080/cmx/report/localhost-orcl-DS_UI1/list/SVR1.2X9N0
{
  "DIMENSION_NAME_1": "Country",
  "DIMENSION_NAME_2": "Users",
  "TIMEPERIOD_NAME": "null",
  "RPT_NAME": "Top 10 Countries",
  "METRIC_NAME": "Number Of Customers",
  "RPT_DESC": "Top 10 Countries"
}
```


Ejemplo de respuesta de API

La API no devuelve una respuesta cuando la configuración del informe se actualiza correctamente.

Agregar o actualizar datos de informe

La API de REST para agregar o actualizar entradas de datos en un informe agrega o actualiza entradas de datos en un informe.

Nota: Si el informe contiene datos de informe, la API reemplaza los datos del informe.

La API usa el método POST.

URL de solicitud

La dirección URL de REST para agregar o actualizar datos de informe tiene el siguiente formato:

```
http://<host>:<port>/<context>/<database ID>/data/<report ID>
```

Haga la siguiente solicitud HTTP POST a la dirección URL:

```
POST http://<host>:<port>/<context>/<database ID>/data/<report ID>
```

Parámetros de consulta

El ID de informe es un parámetro de URL obligatorio. Use el ID de informe para especificar el informe para el que quiera agregar o actualizar las entradas.

Cuerpo de la solicitud

Especifique el informe para el que quiera agregar o actualizar entradas de datos.

En la tabla siguiente, se describen los parámetros de los datos del informe en el cuerpo de la solicitud:

Parámetro	Descripción
DIMENSION_NAME_1	El nombre de una dimensión de datos en el informe. Por ejemplo, una dimensión puede ser nombre de usuario, tipo de tarea de revisión o tipo de entidad de negocio.
DIMENSION_NAME_2	Opcional. El nombre de una dimensión de datos en el informe. Por ejemplo, una dimensión puede ser nombre de usuario, tipo de tarea de revisión o tipo de entidad de negocio.
TIMEPERIOD_NAME	Opcional. Nombre del período de tiempo al que se aplican los datos. Por ejemplo, puede especificar <i>Mes</i> .
METRIC_VALUE	Número que representa las dimensiones de datos. Por ejemplo, puede tener un informe que recaba información acerca de tareas cerradas por el usuario. Una dimensión del informe es el usuario John Smith, y otra dimensión es el tipo de tarea Revisión. El valor del indicador puede ser 5. Esto significa que John Smith cerró cinco tareas del tipo de tarea Revisión.
DRILLDOWN_RPT_ID	ID del informe de detalle. Si el valor es <i>null</i> , no se configura ningún informe de detalle.

Ejemplo de solicitud de API

La siguiente solicitud de ejemplo agrega o actualiza entradas de datos en un informe:

```
POST http://localhost:8080/cmx/report/localhost-orcl-DS_UI1/data/SVR1.2X9N0
[
  {
```

```

    "DIMENSION_VALUE_1": "11:Org,ex.Phone",
    "DIMENSION_VALUE_2": "Cross Source Matches",
    "TIMEPERIOD_VALUE": "null",
    "METRIC_VALUE": "5",
    "DRILLDOWN_RPT_ID": null
  },
  {
    "DIMENSION_VALUE_1": "9:Org Name,City,Zip",
    "DIMENSION_VALUE_2": "Cross Source Matches",
    "TIMEPERIOD_VALUE": "null",
    "METRIC_VALUE": "40",
    "DRILLDOWN_RPT_ID": null
  },
  {
    "DIMENSION_VALUE_1": "12:Org,ex.Email",
    "DIMENSION_VALUE_2": "CRM Matches",
    "TIMEPERIOD_VALUE": "null",
    "METRIC_VALUE": "7",
    "DRILLDOWN_RPT_ID": null
  }
]

```

Ejemplo de respuesta de API

La API no devuelve una respuesta cuando las entradas de datos del informe se agregan o actualizan satisfactoriamente.

Eliminar informe

La API de REST de eliminación de informes elimina un informe.

La API usa el método DELETE.

URL de solicitud

La dirección URL de REST de eliminación de informe tiene el siguiente formato:

```
http://<host>:<port>/<context>/<database ID>/data/<report ID>
```

Haga la siguiente solicitud HTTP DELETE a la dirección URL:

```
DELETE http://<host>:<port>/<context>/<database ID>/data/<report ID>
```

Parámetros de consulta

El ID de informe es un parámetro de URL obligatorio. Use el ID de informe para especificar el informe que quiera eliminar.

Ejemplo de solicitud de API

La siguiente solicitud de ejemplo elimina un informe:

```
DELETE http://localhost:8080/cm/report/localhost-orcl-DS_UI1/data/SVR1.2X9N0
```

Ejemplo de respuesta de API

La API no devuelve una respuesta cuando el informe se elimina correctamente.

Ejecutar el trabajo de actualización de informe

La API de REST de ejecución del trabajo de actualización de informe inicia el trabajo en lotes asociado con el informe de fábrica.

La API usa el método GET.

URL de solicitud

La dirección URL de REST de ejecución del trabajo de actualización de informe tiene el siguiente formato:

```
http://<host>:<port>/<context>/<database ID>/data/collect/<report ID>
```

Haga la siguiente solicitud HTTP GET a la URL:

```
GET http://<host>:<port>/<context>/<database ID>/data/collect/<report ID>
```

Parámetros de consulta

El ID de informe es un parámetro de URL obligatorio. Use el ID de informe para especificar el informe para el que quiera ejecutar un trabajo de actualización.

Cuerpo de la respuesta

La respuesta puede contener uno de los siguientes valores en el parámetro `status`:

Valor	Descripción
PROCESSING	Se desencadenó el trabajo. Cuando se completa el trabajo, se devuelve su ID.
COMPLETED_SUCCESSFULLY	Los datos de informe se calcularon inmediatamente sin ejecutar un trabajo.
DONE	El trabajo no se inició, y se realizó el cálculo en un modo sincronizado.
FAILED	El trabajo no se inició debido a un error. Los atributos <code>errorMessage</code> y <code>errorCode</code> explican el error.

Ejemplo de solicitud de API

La siguiente solicitud de ejemplo devuelve un trabajo de actualización para un informe:

```
GET http://localhost:8080/cmx/report/localhost-orcl-DS_UI1/data/collect/SVR1.2X9N0
```

Ejemplo de respuesta de API

La siguiente respuesta de ejemplo muestra la respuesta sobre la ejecución correcta del trabajo de actualización para un informe:

```
{
  "status": "PROCESSING"
}
```

Obtener estado de trabajo de actualización de informe

La API de REST de obtención de estado de trabajo de actualización de informe devuelve el estado de un trabajo de actualización de informe.

La API usa el método GET.

URL de solicitud

La dirección URL de obtención de estado de trabajo de actualización de informe tiene el siguiente formato:

```
http://<host>:<port>/<context>/<database ID>/data/collect/<report ID>/status
```

Haga la siguiente solicitud HTTP GET a la URL:

```
GET http://<host>:<port>/<context>/<database ID>/data/collect/<report ID>/status
```

Parámetros de consulta

El ID del informe es un parámetro de URL obligatorio. Use el ID de informe para especificar el informe para el que quiera obtener el estado del trabajo de actualización.

Cuerpo de la respuesta

La respuesta contiene los siguientes parámetros:

Parámetro	Descripción
estado	Estado del trabajo de actualización del informe. Los valores son COMPLETED_SUCCESSFULLY, PROCESSING, FAILED o ERROR.
lastUpdateDate	Fecha de la última actualización para el informe. Definida para los trabajos de actualización de informe con el estado COMPLETED_SUCCESSFULLY.
errorCode	Opcional. Código de error para un error con el trabajo de actualización de informe. Definido para los trabajos de actualización de informe con el estado ERROR.
errorMessage	Opcional. Mensaje de error para un error con el trabajo de actualización de informe. Definido para los trabajos de actualización de informe con el estado ERROR.

Ejemplo de solicitud de API

La siguiente solicitud de ejemplo devuelve el estado de un trabajo de actualización de informe:

```
GET http://<host>:<port>/<context>/<database ID>/data/collect/SVR1.2X9N1/status
```

Ejemplo de respuesta de API

El siguiente ejemplo muestra la respuesta al comprobar el estado de un trabajo de actualización de informe:

```
{
  "status": "COMPLETED_SUCCESSFULLY",
  "jobId": "SVR1.2X9N1",
  "lastUpdateDate": "2019-10-31T14:37:11.846-04:00",
  "startRunDate": "2019-10-31T14:37:09.120-04:00"
}
```

Enumerar grupos de trabajos

La API de REST de enumeración de grupos de trabajos devuelve la lista de los grupos de trabajos existentes y sus estados. Puede filtrar los trabajos por tipo de grupo de trabajos. Por ejemplo, puede ver solamente todos los trabajos de importación.

La API usa el método GET.

URL de solicitud

La URL de obtención de lista de trabajos tiene el siguiente formato:

```
http://<host>:<port>/<jobcontrol>/<orsId>/<group?jobGroupType>
```

Haga la siguiente solicitud HTTP GET a la URL de obtención de lista de trabajos:

```
GET http://<host>:<port>/<jobcontrol>/<orsId>/<group?jobGroupType>
```

Parámetros

Especifique los parámetros para devolver la lista de grupos de trabajos existentes y el estado.

En la siguiente tabla se describen los parámetros:

Parámetro	Ruta de acceso	Descripción
jobGroupControllId	Ruta de acceso	ID del grupo de trabajos.
orsId	Ruta de acceso	ID de la base de datos del almacén de referencia operativa para devolver la lista de grupos de trabajos existentes y el estado.
detailed	Consulta	Opcional. Utilice uno de los siguientes valores: True: para ver una lista detallada de grupos de trabajos existentes y el estado. False: para ver una lista breve de grupos de trabajos existentes y el estado.
firstRecord	Consulta	Opcional. El primer registro en la lista de grupos de trabajos existentes.
jobGroupType	Consulta	Opcional. Lista de trabajos por tipo de grupos de trabajos. Utilice uno de los siguientes valores: Import Match Import_After_Match Find_Replace
groupBy	Consulta	Opcional. Agrupe los trabajos en función de la entidad de negocio.
recordsToReturn	Consulta	Opcional. El número de registros que desea devolver.
returnTotal	Consulta	Opcional. Establecida en true. El número total de registros por devolver.

Ejemplo de solicitud de API

La siguiente solicitud de ejemplo devuelve la lista de grupos de trabajos existentes y el estado:

```
GET /cmx/jobcontrol/localhost-orcl-MDM_SAMPLE/group/SVR1.1G7UY?groupBy=owningTable
HTTP/1.1
Host: localhost:8080
```

Ejemplo de respuesta de API

La siguiente respuesta de ejemplo devuelve la lista de grupos de trabajos existentes y el estado:

```
HTTP/1.1 200 OK
Server: JBoss-EAP/7
Content-Disposition: inline;filename=f.txt
X-Powered-By: Undertow/1
Content-Type: application/json;charset=UTF-8
Content-Length: 1551

{
  "jobGroupControlId" : "SVR1.1G7UY",
  "jobGroup" : {
    "type" : "IMPORT",
    "name" : "BE_Import_RH_LE_PersonView[1]_16f9ffd55b5"
  },
  "startDate" : "2020-01-13T13:06:26.649-05:00",
  "endDate" : "2020-01-13T13:06:40.842-05:00",
  "status" : "COMPLETED WITH ERRORS",
  "statusMessage" : "Completed with errors/warnings",
  "errorMessages" : [ "Completed with errors/warnings", "Completed with errors/warnings/warnings", "Completed with errors/warnings", "Completed with errors/warnings", "Completed with errors/warnings" ],
  "jobItems" : [ {
    "startDate" : "2020-01-13T13:06:36.150-05:00",
    "endDate" : "2020-01-13T13:06:40.808-05:00",
    "status" : "COMPLETED WITH ERRORS",
    "owningTable" : "PersonView",
    "errorMessages" : [ "Completed with errors/warnings", "Completed with errors/warnings/warnings", "Completed with errors/warnings", "Completed with errors/warnings" ],
    "progress" : 100,
    "metrics" : {
      "200" : 6,
      "300" : 5,
      "201" : 5,
      "301" : 4,
      "202" : 1,
      "302" : 1
    }
  } ],
  "parameters" : {
    "mappingId" : "SVR1.1G7UX",
    "fileName" : "my.pdf",
    "systemName" : "Admin",
    "fileId" : "DB_SVR1.1G7UW"
  },
  "metricLabels" : {
    "200" : "Total lines in file",
    "201" : "Converted lines",
    "300" : "Total records",
    "202" : "Rejected lines",
    "301" : "Loaded records",
    "302" : "Rejected records"
  },
  "progress" : 100
}
```

Enumerar un grupo de trabajos

La API de REST de enumeración de un grupo de trabajos devuelve datos para cada trabajo en un grupo por lotes.

La API usa el método GET.

URL de solicitud

La dirección URL de REST de enumeración de un grupo de trabajos tiene el siguiente formato:

```
http://<host>:<port>/<jobcontrol>/<orsId>/<groupBy=>
```

Haga la siguiente solicitud HTTP GET a la URL de enumeración de un grupo de trabajos:

```
GET http://<host>:<port>/<jobcontrol>/<orsId>/<groupBy=>
```

Parámetros

Especifique los parámetros para devolver datos para cada trabajo en un grupo por lotes.

En la tabla siguiente se describen los parámetros del cuerpo de la solicitud:

Parámetro	Descripción
orsID	ID de almacén de datos de referencia operativa de la base de datos desde la cual recuperar la asignación.
groupBy	Opcional. Agrupa los trabajos por el tipo de trabajo solicitado.
jobGroupControlId	Especifique el ID de control del grupo de trabajos por lotes.

Ejemplo de solicitud de API

La siguiente solicitud de ejemplo devuelve datos para cada trabajo en un grupo por lotes:

```
GET /cmx/jobcontrol/localhost-orcl-MDM_SAMPLE/group?jobGroupType=IMPORT HTTP/1.1  
Host: localhost:8080
```

Ejemplo de respuesta de API

La siguiente respuesta de ejemplo muestra datos para cada trabajo en un grupo por lotes:

```
HTTP/1.1 200 OK  
Server: JBoss-EAP/7  
X-Powered-By: Undertow/1  
Content-Type: application/json; charset=UTF-8  
Content-Length: 564  
  
{  
  "records" : [ {  
    "jobGroupControlId" : "SVR1.1G7UY",  
    "startDate" : "2020-01-13T13:06:26.649-05:00",  
    "status" : "PENDING",  
    "parameters" : { },  
    "metricLabels" : { }  
  }, {  
    "jobGroupControlId" : "SVR1.1G61Y",  
    "startDate" : "2020-01-13T12:39:42.152-05:00",  
    "endDate" : "2020-01-13T12:40:03.221-05:00",  
    "status" : "COMPLETED_WITH_ERRORS",  
    "statusMessage" : "Completed with errors/warnings",  
    "parameters" : { },  
    "metricLabels" : { }  
  } ],  
  "totalCount" : 2,  
  "firstRecord" : 1,  
  "pageSize" : 10  
}
```

Obtener metadatos de DaaS

La API de REST de obtención de metadatos de DaaS devuelve información sobre un proveedor de DaaS, como el nombre, el tipo, la entidad de negocio con la que trabaja y la lista de campos obligatorios.

La API usa el método GET.

URL de solicitud

La dirección URL de obtención de metadatos de DaaS tiene el siguiente formato:

```
http://<host>:<port>/<context>/<database ID>/meta/daas/<providerName>
```

Use la siguiente solicitud HTTP GET para la dirección URL de obtención de metadatos de DaaS:

```
GET http://<host>:<port>/<context>/<database ID>/meta/daas/<providerName>
```

Parámetro de consulta

El parámetro `providerName` es obligatorio. El parámetro es el nombre del proveedor de DaaS configurado.

Ejemplo de solicitud de API

En la siguiente solicitud de ejemplo se devuelve la información de los metadatos del proveedor de DaaS `dcp`:

```
GET http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/meta/daas/dcp
```

En la siguiente solicitud de ejemplo se devuelve la información de los metadatos del proveedor de DaaS `ondemand`:

```
GET http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/meta/daas/ondemand
```

Ejemplo de respuesta de API

En el siguiente ejemplo se muestra la información de los metadatos del proveedor de DaaS `dcp` en formato JSON:

```
{
  "providerName": "dcp",
  "providerType": "READ",
  "businessEntity": "Organization",
  "systemName": "SFA",
  "requiredFields": [
    "dunsNumber"
  ]
}
```

En el siguiente ejemplo se muestra la información de los metadatos del proveedor de DaaS `ondemand` en formato JSON:

```
{
  "providerName": "ondemand",
  "providerType": "SEARCH",
  "businessEntity": "Organization",
  "systemName": "SFA",
  "requiredFields": [
    "displayName"
  ]
}
```


Búsqueda de DaaS

La API de REST de búsqueda de DaaS usa determinados campos de entrada de una entidad de negocio para llamar a un servicio de DaaS externo y transforma la respuesta en una lista de registros.

La API usa el método POST.

URL de solicitud

La dirección URL de búsqueda de DaaS tiene el siguiente formato:

```
http://<host>:<port>/<context>/<database ID>/daas/search/<daas provider>
```

Use la siguiente solicitud HTTP POST para la dirección URL de búsqueda de DaaS:

```
POST http://<host>:<port>/<context>/<database ID>/daas/search/<daas provider>
```

Nota: En el cuerpo de la solicitud, especifique los detalles de la entidad de negocio con el campo obligatorio.

Cuerpo de la solicitud

El cuerpo de la solicitud debe contener un elemento XML o un elemento JSON del tipo `Organization` u `OrganizationView` en el espacio de nombres `urn:co-ors.informatica.mdm`.

Ejemplo de solicitud de API

El siguiente ejemplo es una solicitud para el proveedor de DaaS `ondemand` para buscar la entidad de negocio `organization` con el nombre para mostrar `INFA`:

```
POST http://localhost:8080/cmx/cs/localhost-orcl-MDM_SAMPLE/daas/search/ondemand
{
  "Organization": {
    "displayname": "INFA"
  }
}
```

Ejemplo de respuesta de API

En la siguiente respuesta de ejemplo se muestran los resultados de una búsqueda que el proveedor de DaaS devuelve para la entidad de negocio de organización con el nombre para mostrar `INFA`:

```
{
  "link": [],
  "item": [
    {
      "businessEntity": {
        "Organization": {
          "displayName": "INFA LAB INC",
          "dunsNumber": "001352574",
          "TelephoneNumbers": {
            "link": [],
            "item": [
              {
                "phoneNum": "9736252265"
              }
            ]
          }
        }
      },
      "Addresses": {
        "link": [],
        "item": [
          {
            "Address": {
              "cityName": "ROCKAWAY",
              "addressLine1": "11 WALL ST",
              "postalCd": "07866",
            }
          }
        ]
      }
    }
  ]
}
```

```

        "stateCd": {
            "stateAbbreviation": "NJ"
        }
    }
}
],
}
},
},
"label": "001352574-INFA LAB INC",
"systemName": "SFA"
},
{
    "businessEntity": {
        "Organization": {
            "displayName": "INFA INC",
            "dunsNumber": "007431013",
            "TelephoneNumbers": {
                "link": [],
                "item": [
                    {
                        "phoneNum": "5629019971"
                    }
                ]
            },
            "Addresses": {
                "link": [],
                "item": [
                    {
                        "Address": {
                            "cityName": "LONG BEACH",
                            "addressLine1": "3569 GARDENIA AVE",
                            "postalCd": "90807",
                            "stateCd": {
                                "stateAbbreviation": "CA"
                            }
                        }
                    }
                ]
            }
        }
    },
    "label": "007431013-INFA INC",
    "systemName": "SFA"
},
{
    "businessEntity": {
        "Organization": {
            "displayName": "INFA INC",
            "dunsNumber": "020591086",
            "TelephoneNumbers": {
                "link": [],
                "item": [
                    {
                        "phoneNum": "7186248777"
                    }
                ]
            },
            "Addresses": {
                "link": [],
                "item": [
                    {
                        "Address": {
                            "cityName": "BROOKLYN",
                            "addressLine1": "16 COURT ST STE 2004",
                            "postalCd": "11241",
                            "stateCd": {
                                "stateAbbreviation": "NY"
                            }
                        }
                    }
                ]
            }
        }
    }
}

```

```

    ]
  }
},
"label": "020591086-INFA INC",
"systemName": "SFA"
},
{
  "businessEntity": {
    "Organization": {
      "displayName": "INFA INC",
      "dunsNumber": "604057286",
      "TelephoneNumbers": {
        "link": [],
        "item": [
          {
            "phoneNum": "8473580802"
          }
        ]
      }
    },
    "Addresses": {
      "link": [],
      "item": [
        {
          "Address": {
            "cityName": "PALATINE",
            "addressLine1": "THE HARRIS BANK BLDG STE 614,800E NW HWY",
            "postalCd": "60074",
            "stateCd": {
              "stateAbbreviation": "IL"
            }
          }
        }
      ]
    }
  }
},
"label": "604057286-INFA INC",
"systemName": "SFA"
},
{
  "businessEntity": {
    "Organization": {
      "displayName": "INFA INC",
      "dunsNumber": "032785606",
      "TelephoneNumbers": {
        "link": [],
        "item": [
          {
            "phoneNum": "5629019971"
          }
        ]
      }
    },
    "Addresses": {
      "link": [],
      "item": [
        {
          "Address": {
            "cityName": "SIMI VALLEY",
            "addressLine1": "3962 HEMWAY CT",
            "postalCd": "93063",
            "stateCd": {
              "stateAbbreviation": "CA"
            }
          }
        }
      ]
    }
  }
},
"label": "032785606-INFA INC",

```

```

    "systemName": "SFA"
  },
  {
    "businessEntity": {
      "Organization": {
        "displayName": "INFA",
        "dunsNumber": "045228877",
        "TelephoneNumbers": {
          "link": [],
          "item": [
            {
              "phoneNum": "3304410033"
            }
          ]
        }
      },
      "Addresses": {
        "link": [],
        "item": [
          {
            "Address": {
              "cityName": "NORTON",
              "addressLine1": "4725 ROCK CUT RD",
              "postalCd": "44203",
              "stateCd": {
                "stateAbbreviation": "OH"
              }
            }
          }
        ]
      }
    }
  },
  "label": "045228877-INFA",
  "systemName": "SFA"
},
{
  "businessEntity": {
    "Organization": {
      "displayName": "INFA INC",
      "dunsNumber": "609028209",
      "TelephoneNumbers": {
        "link": [],
        "item": [
          {
            "phoneNum": "9084394655"
          }
        ]
      },
      "Addresses": {
        "link": [],
        "item": [
          {
            "Address": {
              "cityName": "CALIFON",
              "addressLine1": "21 FAIRMOUNT RD W",
              "postalCd": "07830",
              "stateCd": {
                "stateAbbreviation": "NJ"
              }
            }
          }
        ]
      }
    }
  },
  "label": "609028209-INFA INC",
  "systemName": "SFA"
},
{
  "businessEntity": {
    "Organization": {

```

```

        "displayName": "INFA INC",
        "dunsNumber": "195271796",
        "TelephoneNumbers": {
            "link": [],
            "item": [
                {
                    "phoneNum": "7137824181"
                }
            ]
        },
        "Addresses": {
            "link": [],
            "item": [
                {
                    "Address": {
                        "cityName": "HOUSTON",
                        "addressLine1": "1800 AUGUSTA DR STE 200",
                        "postalCd": "77057",
                        "stateCd": {
                            "stateAbbreviation": "TX"
                        }
                    }
                }
            ]
        }
    },
    "label": "195271796-INFA INC",
    "systemName": "SFA"
},
{
    "businessEntity": {
        "Organization": {
            "displayName": "INFA INC",
            "dunsNumber": "098605830",
            "Addresses": {
                "link": [],
                "item": [
                    {
                        "Address": {
                            "cityName": "BELLEFLOWER",
                            "postalCd": "90707",
                            "stateCd": {
                                "stateAbbreviation": "CA"
                            }
                        }
                    }
                ]
            }
        }
    },
    "label": "098605830-INFA INC",
    "systemName": "SFA"
}
]
}

```

Lectura de DaaS

La API de REST de lectura de DaaS usa determinados campos de una entidad de negocio para solicitar un servicio de DaaS externo y transforma la respuesta en un registro completo.

La API usa el método POST. Especifique el campo obligatorio en la solicitud para el proveedor de DaaS.

URL de solicitud

La dirección URL de lectura de DaaS tiene el siguiente formato:

```
http://<host>:<port>/<context>/<database ID>/daas/read/<daas provider>
```

Use la siguiente solicitud POST para la dirección URL:

```
POST http://<host>:<port>/<context>/<database ID>/daas/read/<daas provider>
```

Nota: En el cuerpo de la solicitud, especifique los detalles del registro con el campo obligatorio.

Parámetro de consulta

En la siguiente tabla se indica el parámetro de consulta que puede usar:

Parámetro	Descripción
logChanges	Opcional. Si se establece en true, el registro resultante incluye el resumen de los cambios del objeto de datos de servicio (SDO), el cual se pasa al servicio de entidad de negocio de escritura. El valor predeterminado es false.

Cuerpo de la solicitud

El cuerpo de la solicitud debe contener un elemento XML o un elemento JSON del tipo `OrganizationView` en el espacio de nombres `urn:coors.informatica.mdm`.

Ejemplo de solicitud de API

El siguiente ejemplo es una solicitud para el proveedor de DaaS `ondemand` para buscar la entidad de negocio `organization` con el nombre para mostrar `INFA`:

```
POST http://localhost:8080/cmxcsllocalhost-orcl-MDM_SAMPLE/daas/search/ondemand
{
  "Organization": {
    "displayname": "INFA"
  }
}
```

Ejemplo de respuesta de API

En la siguiente respuesta de ejemplo se muestran los detalles que el proveedor de DaaS ha devuelto para la organización con el número D-U-N-S 001352574:

```
{
  "Organization": {
    "displayName": "Infa Lab Inc",
    "dunsNumber": "001352574",
    "TelephoneNumbers": {
      "link": [],
      "item": [
        {
          "phoneNum": "09736250550"
        }
      ]
    },
    "Addresses": {
      "link": [],
      "item": [
        {
          "Address": {
            "cityName": "Rockaway",
            "addressLine1": "11 WALL ST"
          }
        }
      ]
    }
  }
}
```

```
}
  }
}
]
```

WriteMerge

La API de REST WriteMerge acepta una lista de registros recuperados de un proveedor de DaaS, los conserva en XREF independientes con el sistema de origen correspondiente y los fusiona en un solo registro. Todas las XREF pertenecen al mismo registro.

La API usa el método POST.

URL de solicitud

La dirección URL de WriteMerge tiene el siguiente formato:

```
http://<host>:<port>/<context>/<database ID>/daas/write-merge/<business entity name>
```

Use la siguiente solicitud HTTP POST para la dirección URL de WriteMerge:

```
POST http://<host>:<port>/<context>/<database ID>/daas/write-merge/<business entity name>
```

Cuerpo de la solicitud

El cuerpo de la solicitud debe contener un elemento XML o JSON del tipo `DaaSEntity.Pager` en el espacio de nombres `urn:cobase.informatica.mdm`.

Encabezado de respuesta

Si la respuesta es satisfactoria, la API devuelve `interactionId` y `processId` en el encabezado de la respuesta.

Ejemplo de solicitud de API

En la siguiente solicitud de ejemplo se usa el resultado de una búsqueda de DaaS como entrada para crear un registro del tipo de entidad de negocio de organización:

```
POST http://localhost:8080/cmx/cs/localhost-orcl-MDM_SAMPLE/daas/write-merge/Organization
{
  "item": [
    {
      "businessEntity": {
        "organization": {
          "displayName": "INFA LAB INC",
          "dunsNumber": "001352574",
          "telephoneNumbers": {
            "item": [
              {
                "phoneNum": "9736252265"
              }
            ]
          }
        }
      }
    },
    {
      "systemName": "DNB"
    }
  ],
  "businessEntity": {
    "organization": {
      "displayName": "INFA INC",
      "dunsNumber": "007431013",
    }
  }
}
```

```

        "TelephoneNumbers": {
            "item": [
                {
                    "phoneNum": "5629019971"
                }
            ]
        }
    },
    "systemName": "Admin"
}
]
}

```

Ejemplo de respuesta de API

La siguiente respuesta de ejemplo muestra el encabezado y el cuerpo de la respuesta después de fusionar correctamente la lista de registros en un solo registro:

```

{
  "Organization": {
    "key": {
      "rowid": "121921",
      "sourceKey": "140000000000"
    },
    "rowidObject": "121921",
    "TelephoneNumbers": {
      "link": [],
      "item": [
        {
          "key": {
            "rowid": "21721",
            "sourceKey": "140000001000"
          },
          "rowidObject": "21721"
        }
      ]
    }
  }
}

```

Importación de DaaS

La API de REST de importación de DaaS acepta datos en formato XML y los convierte en un registro.

La API usa el método POST.

URL de solicitud

La dirección URL de importación de datos de DaaS tiene el siguiente formato:

```

http://<host>:<port>/<context>/<database ID>/daas/import/
FamilyTreeMemberOrganizationToOrgView

```

Use la siguiente solicitud HTTP POST para la dirección URL de importación de datos de DaaS:

```

POST http://<host>:<port>/<context>/<database ID>/daas/import/
FamilyTreeMemberOrganizationToOrgView

```


Parámetros de consulta

El nombre del sistema de origen es un parámetro de obligatorio.

En la siguiente tabla se indican los parámetros que puede usar en la solicitud:

Parámetro	Descripción
systemName	Obligatorio. Nombre del sistema de origen que realiza el cambio en los datos.
interactionId	Opcional. ID de interacción que se asigna a todos los cambios. Normalmente, el concentrador genera el ID al crear un cambio pendiente como resultado de una llamada.
effectivePeriod	Opcional. Contiene la fecha de inicio y la fecha de finalización. Especifica el periodo de tiempo en el que el registro es efectivo. Especifique estos parámetros para un registro habilitado para la línea temporal.
validateOnly	Opcional. Si se establece en TRUE, solo se aplican las reglas de validación al registro modificado y los cambios no son persistentes.
recordState	Opcional. Estado del concentrador para los registros creados o actualizados. Los estados de registro admitidos son ACTIVE y PENDING.
processId	Opcional. ID del proceso de flujo de trabajo que contiene la tarea. Si un flujo de trabajo se inicia como un resultado de la llamada de servicio, el parámetro contiene el identificador del proceso de flujo de trabajo iniciado.

TEMAS RELACIONADOS

- [“Formatos de fecha y hora UTC” en la página 36](#)

Cuerpo de la solicitud

El cuerpo de la solicitud debe contener un elemento XML o JSON del tipo

`DaaSChangeFamilyTreeMemberOrganizationToOrgView` en el espacio de nombres `urn:co-ors.informatica.mdm`.

Encabezado de respuesta

Cuando la respuesta sea satisfactoria, la API devolverá el ID de interacción (`interactionId`) y el ID de proceso (`processId`) en el encabezado de la respuesta y los detalles del registro en el cuerpo de la respuesta.

Si el proceso genera un ID de interacción y lo utiliza para crear el registro, la API devolverá el ID de interacción. Si el proceso inicia un flujo de trabajo en lugar de guardar directamente el registro en la base de datos, la API devolverá el ID de proceso, que es el ID del proceso de flujo de trabajo.

En el ejemplo siguiente se muestra un encabezado de respuesta con un ID de interacción y un ID de proceso:

```
BES-interactionId: 72200000242000  
BES-processId: 15948
```

Ejemplo de solicitud de API

La solicitud siempre tiene el formato XML.

En la siguiente solicitud de ejemplo se muestran los datos XML del tipo ChildAssociation del espacio de nombres de vinculación:

```

POST http://localhost:8080/cmxcsllocalhost-orcl-MDM_SAMPLE/daas/import/linkage2org?
systemName=Admin
<FamilyTreeMemberOrganization xmlns="http://services.dnb.com/LinkageServiceV2.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="ChildAssociation">
  <AssociationTypeText>ParentSubsidiary</AssociationTypeText>
  <OrganizationName>
    <OrganizationPrimaryName>
      <OrganizationName>INFORMATICA JAPAN K.K.</OrganizationName>
    </OrganizationPrimaryName>
  </OrganizationName>
  <SubjectHeader>
    <DUNSNumber>697557825</DUNSNumber>
  </SubjectHeader>
  <Location>
    <PrimaryAddress>
      <StreetAddressLine>
        <LineText>2-5-1, ATAGO</LineText>
      </StreetAddressLine>
      <StreetAddressLine>
        <LineText>ATAGO GREEN HILLS MORI TOWER 26F.</LineText>
      </StreetAddressLine>
      <PrimaryTownName>MINATO-KU</PrimaryTownName>
      <CountryISOAlpha2Code>JP</CountryISOAlpha2Code>
      <TerritoryAbbreviatedName>TKY</TerritoryAbbreviatedName>
      <PostalCode>105-0002</PostalCode>
      <TerritoryOfficialName>TOKYO</TerritoryOfficialName>
    </PrimaryAddress>
  </Location>
  <OrganizationDetail>
    <FamilyTreeMemberRole>
      <FamilyTreeMemberRoleText>Subsidiary</FamilyTreeMemberRoleText>
    </FamilyTreeMemberRole>
    <FamilyTreeMemberRole>
      <FamilyTreeMemberRoleText>Headquarters</FamilyTreeMemberRoleText>
    </FamilyTreeMemberRole>
    <StandaloneOrganizationIndicator>>false</StandaloneOrganizationIndicator>
  </OrganizationDetail>
  <Linkage>
    <LinkageSummary>
      <ChildrenSummary>
        <ChildrenQuantity>1</ChildrenQuantity>
        <DirectChildrenIndicator>>false</DirectChildrenIndicator>
      </ChildrenSummary>
      <ChildrenSummary>
        <ChildrenTypeText>Affiliate</ChildrenTypeText>
        <ChildrenQuantity>29</ChildrenQuantity>
        <DirectChildrenIndicator>>false</DirectChildrenIndicator>
      </ChildrenSummary>
      <ChildrenSummary>
        <ChildrenTypeText>Branch</ChildrenTypeText>
        <ChildrenQuantity>1</ChildrenQuantity>
        <DirectChildrenIndicator>>true</DirectChildrenIndicator>
      </ChildrenSummary>
      <SiblingCount>29</SiblingCount>
    </LinkageSummary>
    <GlobalUltimateOrganization>
      <DUNSNumber>825320344</DUNSNumber>
    </GlobalUltimateOrganization>
    <DomesticUltimateOrganization>
      <DUNSNumber>697557825</DUNSNumber>
    </DomesticUltimateOrganization>
    <ParentOrganization>
      <DUNSNumber>825320344</DUNSNumber>
    </ParentOrganization>
    <FamilyTreeMemberOrganization>
      <AssociationTypeText>HeadquartersBranch</AssociationTypeText>
      <OrganizationName>
        <OrganizationPrimaryName>

```

```

        <OrganizationName>INFORMATICA JAPAN K.K.</OrganizationName>
    </OrganizationPrimaryName>
</OrganizationName>
<SubjectHeader>
    <DUNSNumber>692640710</DUNSNumber>
    <SubjectHandling>
        <SubjectHandlingText DNBCodeValue="11028">De-listed</
SubjectHandlingText>
    </SubjectHandling>
</SubjectHeader>
<Location>
    <PrimaryAddress>
        <StreetAddressLine>
            <LineText>2-2-2, UMEDA, KITA-KU</LineText>
        </StreetAddressLine>
        <PrimaryTownName>OSAKA</PrimaryTownName>
        <CountryISOAlpha2Code>JP</CountryISOAlpha2Code>
        <TerritoryAbbreviatedName>OSK</TerritoryAbbreviatedName>
        <PostalCode>530-0001</PostalCode>
        <TerritoryOfficialName>OSAKA</TerritoryOfficialName>
    </PrimaryAddress>
</Location>
<OrganizationDetail>
    <FamilyTreeMemberRole>
        <FamilyTreeMemberRoleText>Branch</FamilyTreeMemberRoleText>
    </FamilyTreeMemberRole>
    <StandaloneOrganizationIndicator>>false</StandaloneOrganizationIndicator>
</OrganizationDetail>
<Linkage>
    <GlobalUltimateOrganization>
        <DUNSNumber>825320344</DUNSNumber>
    </GlobalUltimateOrganization>
    <DomesticUltimateOrganization>
        <DUNSNumber>697557825</DUNSNumber>
    </DomesticUltimateOrganization>
    <HeadquartersOrganization>
        <DUNSNumber>697557825</DUNSNumber>
    </HeadquartersOrganization>
    <FamilyTreeHierarchyLevel>2</FamilyTreeHierarchyLevel>
</Linkage>
</FamilyTreeMemberOrganization>
<FamilyTreeHierarchyLevel>2</FamilyTreeHierarchyLevel>
</Linkage>
</FamilyTreeMemberOrganization>

```

Ejemplo de respuesta de API

En la siguiente respuesta de ejemplo se muestran el encabezado y el cuerpo de la respuesta después de importar correctamente una entidad de negocio de organización:

```

BES-interactionId: 72202100242034
BES-processId: 156048
{
  "Organization": {
    "key": {
      "rowid": "101921",
      "sourceKey": "697557825"
    },
    "rowidObject": "101921"
  }
}

```

Actualización de DaaS

La API de REST de actualización de DaaS acepta los datos en formato XML antes y después de un cambio. La API aplica los cambios al registro.

La API usa el método POST.

URL de solicitud

La dirección URL de actualización de DaaS tiene el siguiente formato:

```
http://<host>:<port>/<context>/<database ID>/daas/update/  
FamilyTreeMemberOrganizationToOrgView
```

Use la siguiente solicitud HTTP POST para la dirección URL de actualización de DaaS:

```
POST http://<host>:<port>/<context>/<database ID>/daas/update/  
FamilyTreeMemberOrganizationToOrgView
```

Parámetros de consulta

El nombre del sistema de origen es un parámetro de obligatorio.

En la siguiente tabla se indican los parámetros que puede usar en la solicitud:

Parámetro	Descripción
systemName	Obligatorio. Nombre del sistema de origen que realiza el cambio en los datos.
interactionId	Opcional. ID de interacción que se asigna a todos los cambios. Normalmente, el concentrador genera el ID al crear un cambio pendiente como resultado de una llamada.
effectivePeriod	Opcional. Contiene la fecha de inicio y la fecha de finalización. Especifica el periodo de tiempo en el que el registro es efectivo. Especifique estos parámetros para un registro habilitado para la línea temporal.
validateOnly	Opcional. Si se establece en TRUE, solo se aplican las reglas de validación al registro modificado y los cambios no son persistentes.
recordState	Opcional. Estado del concentrador para los registros creados o actualizados. Los estados de registro admitidos son ACTIVE y PENDING.
processId	Opcional. ID del proceso de flujo de trabajo que contiene la tarea. Si el flujo de trabajo se inicia como un resultado de la llamada de servicio, el parámetro contiene el identificador del proceso de flujo de trabajo iniciado.

TEMAS RELACIONADOS

- [“Formatos de fecha y hora UTC” en la página 36](#)

Cuerpo de la solicitud

El cuerpo de la solicitud debe contener un elemento XML o JSON del tipo

DaaSChangeFamilyTreeMemberOrganizationToOrgView en el espacio de nombres urn:co-
ors.informatica.mdm.

Encabezado de respuesta

Cuando la respuesta sea satisfactoria, la API devolverá el ID de interacción (interactionId) y el ID de proceso (processId) en el encabezado de la respuesta y los detalles del registro en el cuerpo de la respuesta.

Si el proceso genera un ID de interacción y lo utiliza para crear el registro, la API devolverá el ID de interacción. Si el proceso inicia un flujo de trabajo en lugar de guardar directamente el registro en la base de datos, la API devolverá el ID de proceso, que es el ID del proceso de flujo de trabajo.

En el ejemplo siguiente se muestra un encabezado de respuesta con un ID de interacción y un ID de proceso:

```
BES-interactionId: 72200000242000
BES-processId: 15948
```

Ejemplo de solicitud de API

La API acepta dos respuestas en formato XML. Una respuesta es anterior a un cambio y la otra respuesta es posterior al cambio. En la siguiente solicitud se añade un número de teléfono nuevo a la organización. Los datos XML *before* no incluyen el número de teléfono y los datos XML *after* sí incluyen el número de teléfono.

La siguiente solicitud contiene el número de teléfono recién añadido:

```
POST http://localhost:8080/cmx/cs/localhost-orcl-MDM_SAMPLE/daas/update/linkage2org?
systemName=Admin
<urn:DaaSChangelinkage2org xmlns:urn="urn:cs-ors.informatica.mdm" xmlns:xsi="http://
www.w3.org/2001/XMLSchema-instance" xsi:type="urn:DaaSChangelinkage2org">
  <urn:before xmlns="http://services.dnb.com/LinkageServiceV2.0">
    <SubjectHeader>
      <DUNSNumber>697557825</DUNSNumber>
    </SubjectHeader>
  </urn:before>

  <urn:after xmlns="http://services.dnb.com/LinkageServiceV2.0">
    <SubjectHeader>
      <DUNSNumber>697557825</DUNSNumber>
    </SubjectHeader>
    <Telecommunication>
      <TelephoneNumber>
        <TelecommunicationNumber>09736250550</TelecommunicationNumber>
        <InternationalDialingCode>1</InternationalDialingCode>
        <UnreachableIndicator>true</UnreachableIndicator>
      </TelephoneNumber>
    </Telecommunication>
  </urn:after>
</urn:DaaSChangelinkage2org>
```

Ejemplo de respuesta de API

En el siguiente ejemplo se muestra el valor de rowId del número de teléfono recién creado de la organización:

```
{
  "Organization": {
    "key": {
      "rowid": "101921",
      "sourceKey": "697557825"
    },
    "rowidObject": "101921",
    "TelephoneNumbers": {
      "link": [],
      "item": [
        {
          "key": {
            "rowid": "1722",
            "sourceKey": "09736250550"
          },
          "rowidObject": "1722"
        }
      ]
    }
  }
}
```

CAPÍTULO 4

API REST para Data Director

Puede utilizar las API REST para cambiar la marca de Data Director y configurar la exención de responsabilidad legal de la página de inicio de sesión.

API REST para cambiar la marca de Data Director

Puede utilizar las API REST para cambiar la marca de Data Director y personalizar el aspecto de la aplicación. Puede cambiar el tema de color predeterminado, el fondo de inicio de sesión y el logotipo de la aplicación Data Director.

La siguiente tabla describe las API REST para cambiar el nombre de Data Director:

API REST	Descripción
Upload Logo File	Carga el archivo de imagen del logotipo. Admite el método POST.
Eliminar archivo de logotipo	Elimina el archivo de imagen del logotipo. Admite el método DELETE.
Cargar archivo de fondo de inicio de sesión	Carga el archivo de imagen de fondo de la página de inicio de sesión. Admite el método POST.
Delete Login BG File	Elimina el archivo de imagen de fondo de la página de inicio de sesión. Admite el método DELETE.
Obtener variables	Lee la configuración de la interfaz de usuario, como los colores de los botones, menús y texto en la aplicación Data Director. Admite el método GET.
Actualizar variables	Actualiza la configuración de la interfaz de usuario, como los colores de los botones, menús y texto en la aplicación Data Director. Admite el método POST.
Eliminar esquema de color	Restaura la configuración predeterminada de la interfaz de usuario. Admite el método DELETE.

Cargar archivo de fondo de inicio de sesión

La API REST Upload Login BG File carga una imagen de fondo para la página de inicio de sesión de la aplicación Data Director. Para especificar el nombre del archivo, agréguelo al encabezado X-File-Name de la

solicitud POST. Envíe los datos de la imagen en el cuerpo de la solicitud REST. El valor del encabezado Content-Type debe ser `application/octet-stream`.

La API usa el método POST.

El tamaño del archivo de imagen no debe exceder el valor de la propiedad `cmx.file.max_file_size_mb` especificada en el archivo `cmxserver.properties`. La API admite los tipos de archivos de imagen que admite el navegador. Para obtener información sobre la versión compatible del navegador, consulte la Tabla de disponibilidad de productos (PAM) en <https://network.informatica.com/community/informatica-network/product-availability-matrices>.

URL de solicitud

La URL de Upload Login BG File tiene el siguiente formato:

```
http://<host>:<port>/<context>/<path>
```

Use la siguiente solicitud HTTP POST para la URL de Update Login BG:

```
POST http://<host>:<port>/cmx/ui/theme/login_bg_image
```

Ejemplo de solicitud de API

La siguiente solicitud de ejemplo carga el archivo de imagen de fondo de la página de inicio de sesión:

```
POST http://localhost:8080/cmx/ui/theme/login_bg_image
```

Ejemplo de respuesta de API

La API devuelve un código de respuesta 200 OK después de cargar con éxito un nuevo archivo de imagen de fondo para la página de inicio de sesión de Data Director. El cuerpo de la respuesta está vacío.

Delete Login BG File

La API REST Delete Login BG File elimina la imagen de fondo personalizada de la página de inicio de sesión de la aplicación Data Director.

La API usa el método DELETE.

Después de eliminar la imagen de fondo personalizada, la página de inicio de sesión muestra la imagen de fondo predeterminada.

URL de solicitud

La dirección URL de Delete Login BG File tiene el siguiente formato:

```
http://<host>:<port>/<context>/<path>
```

Haga la siguiente solicitud HTTP DELETE a la dirección URL de Delete Login BG File:

```
DELETE http://<host>:<port>/cmx/ui/theme/login_bg_image
```

Ejemplo de solicitud de API

La siguiente solicitud de ejemplo elimina la imagen de fondo de la página de inicio de sesión de la aplicación Data Director:

```
DELETE http://localhost:8080/cmx/ui/theme/login_bg_image
```

Ejemplo de respuesta de API

La API devuelve un código de respuesta 200 OK después de eliminar con éxito la imagen de fondo de la página de inicio de sesión de la aplicación Data Director. El cuerpo de la respuesta está vacío.

Upload Logo File

La API REST Upload Logo File carga una nueva imagen de logotipo para la aplicación Data Director. Para especificar el nombre del archivo, agréguelo al encabezado X-File-Name de la solicitud POST. Envíe los datos de la imagen en el cuerpo de la solicitud REST. El valor del encabezado Content-Type debe ser `application/octet-stream`.

La API usa el método POST.

El tamaño del archivo de la imagen no puede superar los 130 píxeles de ancho y los 33 píxeles de alto. La API admite los tipos de archivos de imagen que admite el navegador. Para obtener información sobre la versión compatible del navegador, consulte la Tabla de disponibilidad de productos (PAM) en <https://network.informatica.com/community/informatica-network/product-availability-matrices>.

URL de solicitud

La dirección URL de Upload Logo File tiene el siguiente formato:

```
http://<host>:<port>/<context>/<path>
```

Use la siguiente solicitud HTTP POST para la dirección URL de Upload Logo File:

```
POST http://localhost:8080/cmx/ui/theme/logo
```

Ejemplo de solicitud de API

La siguiente solicitud de ejemplo carga la imagen del logotipo de la aplicación:

```
POST http://localhost:8080/cmx/ui/theme/logo HTTP/1.1
Host: localhost:8080
X-File-Name: logo_dockers.jpg
Content-Type: application/octet-stream
```

Ejemplo de respuesta de API

La API devuelve un código de respuesta de 200 OK después de cargar correctamente la imagen del logotipo. El cuerpo de la respuesta está vacío.

Eliminar archivo de logotipo

La API REST Delete Logo File elimina el archivo de imagen de logotipo personalizado de la aplicación.

La API usa el método DELETE.

Nota: Después de eliminar la imagen del logotipo personalizado, la página de inicio de sesión muestra la imagen del logotipo predeterminada.

URL de solicitud

La dirección URL de Delete Logo File tiene el siguiente formato:

```
http://<host>:<port>/<context>/<path>
```

Haga la siguiente solicitud HTTP DELETE a la dirección URL de Delete Logo File:

```
DELETE http://<host>:<port>/cmx/ui/theme/logo
```

URL de solicitud de API de ejemplo

La siguiente solicitud de ejemplo elimina el archivo del logotipo de Data Director:

```
DELETE http://localhost:8080/cmx/ui/theme/logo
```

Ejemplo de respuesta de API

La API devuelve un código de respuesta 200 OK después de eliminar con éxito el archivo de imagen del logotipo. El cuerpo de la respuesta está vacío.

Obtener variables

La API REST Obtener variables devuelve las propiedades de esquema de color actuales de Data Director.

La API usa el método GET.

URL de solicitud

La URL de Obtener variables tiene el siguiente formato:

```
http://<host>:<port>/<context>/<path>
```

Haga la siguiente solicitud HTTP GET a la URL de Obtener variables:

```
GET http://<host>:<port>/cmx/ui/theme/variables
```

Ejemplo de solicitud de API

La siguiente solicitud devuelve las propiedades del tema de color actual.

```
GET http://localhost:8080/cmx/ui/theme/variables
```

Ejemplo de respuesta de API

La API devuelve un código de respuesta 200 OK después de recuperar con éxito las propiedades del esquema de color.

```
"variables": {
  "menu_text_color": "black",
  "dropdown_menu_item_checked_hover_bg_color": "Gold",
  "button_default_selected_border_color": "blue",
  "dropdown_menu_item_checked_hover_text_color": "Coral",
  "button_success_text_color": "white",
  "login_form_text_color": "green",
  "menu_hover_text_color": "#07fc03",
  "dropdown_menu_item_checked_text_color": "LightSalmon",
  "button_danger_text_color": "black",
  "menu_hover_bg_color": "#00349D",
  "header_search_text_color": "#36270a",
  "button_default_border_color": "red",
  "dropdown_menu_item_text_color": "#daf97b",
  "button_danger_bg_color": "pink",
  "button_default_bg_color": "yellow",
  "menu_sel_text_color": "#26cfe0",
  "button_success_border_color": "olive",
  "dropdown_menu_text_color": "#945d29",
  "dropdown_menu_bg_color": "#a4c7f5",
  "primary_button_text_color": "white",
  "header_search_bg_color": "#a4c7f5",
  "dropdown_menu_item_checked_bg_color": "LightGoldenrodYellow",
  "button_default_selected_bg_color": "orange",
  "primary_button_bg_color": "green",
  "entity_view_label_font_wight": "bold",
  "primary_button_selected_bg_color": "pink",
  "primary_button_selected_border_color": "red",
  "header_text_color": "#36270a",
  "button_danger_selected_text_color": "green",
  "primary_button_border_color": "white",
  "menu_sel_bg_color": "#636260",
  "button_success_selected_border_color": "red",
  "login_form_background_color": "#fcd483",
  "dropdown_menu_item_hover_text_color": "OrangeRed",
  "header_background_color": "#c4922d",
  "button_success_selected_bg_color": "PaleVioletRed",
  "button_danger_border_color": "purple",
  "button_default_text_color": "red",
  "button_success_bg_color": "orange",
  "button_danger_selected_bg_color": "yellow",
  "button_danger_selected_border_color": "red",
  "primary_button_selected_text_color": "red",
```

```

        "workspace_bg": "#cadee0",
        "button_default_selected_text_color": "blue",
        "dropdown_menu_item_hover_bg_color": "RoyalBlue",
        "menu_background_color": "#cdc1bb",
        "button_success_selected_text_color": "black"
    }
}

```

Para obtener más información sobre las propiedades del esquema de color de Data Director, consulte ["Esquema de color predeterminado" en la página 267](#).

Actualizar variables

La API REST Update Variables actualiza las propiedades de color actuales de Data Director. Utilice el formato JSON para enviar datos en el cuerpo de la solicitud.

La API usa el método POST.

Nota: Es posible que la solicitud de la API no actualice las propiedades de color de forma coherente en toda la aplicación. Por ejemplo, después de enviar la solicitud para actualizar las propiedades de color de los botones y menús, el botón Guardar puede diferir en distintas pantallas y algunos menús pueden cambiar solo el color de fondo.

URL de solicitud

La dirección URL de Update Variables tiene el siguiente formato:

```
http://<host>:<port>/<context>/<path>
```

Use la siguiente solicitud HTTP POST para la dirección URL de Update Variables:

```
POST http://<host>:<port>/cmx/ui/theme/variables
```

Ejemplo de solicitudes de API

La siguiente solicitud actualiza todas las propiedades de color de la aplicación Data Director:

```

POST http://localhost:8080/cmx/ui/theme/variables

{
  "variables": {
    "menu_text_color": "#FFF",
    "button_default_selected_border_color": "orange",
    "button_success_text_color": "white",
    "login_form_text_color": "red",
    "button_danger_text_color": "black",
    "menu_hover_bg_color": "#00349D",
    "button_default_border_color": "yellow",
    "dropdown_menu_item_text_color": "white",
    "button_danger_bg_color": "pink",
    "button_default_bg_color": "yellow",
    "button_success_border_color": "olive",
    "dropdown_menu_text_color": "white",
    "dropdown_menu_bg_color": "olive",
    "primary_button_text_color": "white",
    "button_default_selected_bg_color": "orange",
    "primary_button_bg_color": "green",
    "entity_view_label_font_wight": "normal",
    "primary_button_selected_bg_color": "silver",
    "primary_button_selected_border_color": "black",
    "header_text_color": "#444444",
    "button_danger_selected_text_color": "black",
    "primary_button_border_color": "green",
    "menu_sel_bg_color": "#001E60",
    "button_success_selected_border_color": "black",
    "login_form_background_color": "#E5EDF8",
    "header_background_color": "#ffd900",
    "button_success_selected_bg_color": "silver",
  }
}

```

```

"button_danger_border_color": "pink",
"button_default_text_color": "black",
"button_success_bg_color": "olive",
"button_danger_selected_bg_color": "orange",
"button_danger_selected_border_color": "red",
"primary_button_selected_text_color": "black",
"workspace_bg": "#E5EDF8",
"button_default_selected_text_color": "black",
"dropdown_menu_item_hover_bg_color": "navy",
"menu_background_color": "#004fb6",
"button_success_selected_text_color": "black"
}
}

```

La siguiente solicitud actualiza las propiedades de color de los botones principales en la aplicación Data Director.

```

{
  "variables":{
    "primary_button_text_color": "white",
    "primary_button_bg_color": "green",
    "primary_button_border_color": "white",
    "primary_button_selected_bg_color": "pink",
    "primary_button_selected_border_color": "red",
    "primary_button_selected_text_color": "red"
  }
}

```

Ejemplo de respuesta de API

La API devuelve un código de respuesta 200 OK después de actualizar con éxito las propiedades de color especificadas. El cuerpo de la respuesta está vacío.

Esquema de color predeterminado

Según las necesidades de su organización, puede cambiar las propiedades de color predeterminadas de la aplicación Data Director. Algunas propiedades afectan a un componente específico de la interfaz de usuario de Data Director. Otras propiedades, como las propiedades del botón, afectan a varios componentes de la interfaz de usuario.

Página de inicio de sesión

La siguiente tabla enumera las propiedades que puede cambiar para la página de inicio de sesión:

Propiedad	Descripción	Predeterminado
login_form_text_color	Color del texto de las etiquetas de campo en la página de inicio de sesión.	#333333
login_form_background_color	Color de fondo de la página de inicio de sesión.	#fff

Encabezado de página

La siguiente tabla enumera las propiedades que puede cambiar para el encabezado de página de Data Director:

Propiedad	Descripción	Predeterminado
header_text_color	Color del texto del encabezado de la página.	#fff
header_search_text_color	Color de los nombres de las entidades de negocio que aparecen en el encabezado de la página a la izquierda del campo de búsqueda.	#000
header_background_color	Color de fondo del encabezado de la página.	#000

Barra de navegación

La siguiente tabla enumera las propiedades que puede cambiar para la barra de navegación izquierda:

Propiedad	Descripción	Predeterminado
menu_text_color	Color del texto de un elemento de la barra de navegación que el usuario no seleccionó o sobre el que no pasó el cursor.	#fff
menu_sel_text_color	Color del texto de un elemento de la barra de navegación seleccionado.	#fff
menu_hover_text_color	Color del texto de un elemento de la barra de navegación sobre el que un usuario pasa el cursor.	#fff
menu_background_color	Color del texto de fondo de un elemento de la barra de navegación que el usuario no seleccionó o sobre el que no pasó el cursor.	#333
menu_hover_bg_color	Color de fondo del elemento de la barra de navegación sobre el que el usuario pasa el cursor.	#000
menu_sel_bg_color	Color de fondo de un elemento de la barra de navegación seleccionado.	#000

Espacios de trabajo

La siguiente tabla enumera las propiedades que puede cambiar para un espacio de trabajo o la vista de registro de la entidad de negocio:

Propiedad	Descripción	Predeterminado
workspace_bg	Color de fondo del espacio de trabajo que aparece a la derecha de la barra de navegación.	#fff
entity_view_label_font_weight	Peso de fuente de las etiquetas de campo en la vista de solo lectura.	Normal

Listas desplegables

La siguiente tabla enumera las propiedades que puede cambiar para las listas desplegables en la aplicación:

Propiedad	Descripción	Predeterminado
dropdown_menu_text_color	Color del texto de los elementos de la lista.	#000
dropdown_menu_bg_color	Color de fondo de una lista desplegable.	#fff
dropdown_menu_item_hover_text_color	Color del texto de un elemento de la lista que el usuario selecciona o sobre el que pasa el cursor.	#2f3342
dropdown_menu_item_hover_bg_color	Color de fondo de un elemento de la lista que el usuario selecciona o sobre el que pasa el cursor.	#fabfc

Botones de aplicación

Los botones de la aplicación pertenecen a las siguientes categorías:

- 1. Botones que pueden provocar la pérdida de datos al cancelar una operación. Por ejemplo, el botón **Cancelar edición** en las vistas de registros de entidad de negocio.
- 2. Botones que envían o guardan datos. Por ejemplo, el botón **Editar** o **Guardar** en las vistas de registros de entidad de negocio.
- 3. Botones que realizan la acción principal en la interfaz de usuario. Por ejemplo, el botón **Volver al formulario**.
- 4. Botones predeterminados. Botones que se encuentran en la mayoría de las páginas y cuadros de diálogo. Por ejemplo, el botón **Aceptar**.

La siguiente tabla enumera los estilos y propiedades de los botones que puede cambiar:

Estilo de botón	Propiedad	Descripción	Predeterminado
1	button_danger_text_color	Color del texto en un botón.	#3df
1	button_danger_bg_color	Color de fondo de un botón.	#ddd
1	button_danger_border_color	Color de borde de un botón.	#2373e4
1	button_danger_selected_border_color	Color del borde de un botón que el usuario selecciona o sobre el que pasa el cursor.	#2373e4
1	button_danger_selected_text_color	Color del texto de un botón que el usuario selecciona o sobre el que pasa el cursor.	#3df
1	button_danger_selected_bg_color	Color de fondo de un botón que el usuario selecciona o sobre el que pasa el cursor.	#ddd
2	button_success_text_color	Color del texto en un botón.	#3df
2	button_success_bg_color	Color de fondo de un botón.	#ddd
2	button_success_border_color	Color de borde de un botón.	#2373e4
2	button_success_selected_text_color	Color del texto de un botón que el usuario selecciona o sobre el que pasa el cursor.	#3df
2	button_success_selected_bg_color	Color de fondo de un botón que el usuario selecciona o sobre el que pasa el cursor.	#ddd
2	button_success_selected_border_color	Color del borde de un botón que el usuario selecciona o sobre el que pasa el cursor.	#2373e4
3	primary_button_text_color	Color del texto en un botón.	#fff
3	primary_button_bg_color	Color de fondo de un botón.	#4b91ea
3	primary_button_border_color	Color de borde de un botón.	#3483e7
3	primary_button_selected_text_color	Color del texto de un botón que el usuario selecciona o sobre el que pasa el cursor.	#fff
3	primary_button_selected_bg_color	Color de fondo de un botón que el usuario selecciona o sobre el que pasa el cursor.	#4b91ea
3	primary_button_selected_border_color	Color del borde de un botón cuando el usuario no lo selecciona ni pasa el cursor sobre él.	#3483e7
4	button_default_text_color	Color del texto en un botón.	#333
4	button_default_bg_color	Color de fondo de un botón.	#fff
4	button_default_border_color	Color de borde de un botón.	#eaeaea

Estilo de botón	Propiedad	Descripción	Predeterminado
4	button_default_selected_text_color	Color del texto de un botón que el usuario selecciona o sobre el que pasa el cursor.	#333
4	button_default_selected_bg_color	Color de fondo de un botón que el usuario selecciona o sobre el que pasa el cursor.	#fff
4	button_default_selected_border_color	Color del borde de un botón que el usuario selecciona o sobre el que pasa el cursor.	#eaeaea

Eliminar esquema de color

La API REST Delete Color Schema elimina las propiedades de color personalizadas y restaura las propiedades de color predeterminadas en la aplicación Data Director.

La API usa el método DELETE.

URL de solicitud

La URL de Delete Color Schema tiene el siguiente formato:

```
http://<host>:<port>/<context>/
```

Haga la siguiente solicitud DELETE HTTP a la URL de Delete Color Schema:

```
DELETE http://<host>:<port>/cmx/ui/theme
```

Ejemplo de solicitud de API

La siguiente solicitud de muestra elimina las propiedades de color personalizadas y restaura las propiedades predeterminadas:

```
DELETE http://localhost:8080/cmx/ui/theme
```

El cuerpo de la solicitud está vacío.

Ejemplo de respuesta de API

La API devuelve un código de respuesta 200 OK después de eliminar con éxito las propiedades de color actuales de la aplicación Data Director y restaurar las propiedades predeterminadas. El cuerpo de la respuesta está vacío.

API REST para configurar la exención de responsabilidad legal

Puede utilizar las API REST para configurar la exención de responsabilidad legal de la página de inicio de sesión para Data Director, la herramienta de aprovisionamiento y la consola del concentrador. Por ejemplo,

puede describir los usos permitidos de las aplicaciones y las políticas de seguridad impuestas por su organización. El usuario debe reconocer la exención de responsabilidad legal antes de iniciar sesión.

La siguiente tabla enumera las API REST para configurar la exención de responsabilidad legal de la página de inicio de sesión:

API de REST	Descripción
Read Legal Message Configuration	Lee la información del archivo de exención de responsabilidad legal. Por ejemplo, recupera la exención de responsabilidad legal actual. Admite el método GET.
Establecer la configuración de mensajes legales	Crea o actualiza un archivo de descargo de responsabilidad legal. Por ejemplo, actualiza la exención de responsabilidad legal actual. Admite el método POST.
Eliminar la configuración de mensajes legales	Elimina un archivo de exención de responsabilidad legal. Por ejemplo, elimina la exención de responsabilidad legal actual. Admite el método DELETE.

Parámetros de descargo de responsabilidad legal

Puede especificar el texto de la exención de responsabilidad legal, el título y si la exención de responsabilidad aparece cada vez que el usuario intenta iniciar sesión. También puede especificar las aplicaciones que muestran la exención de responsabilidad legal en la página de inicio de sesión.

En la siguiente tabla, se enumeran los parámetros de exención de responsabilidad legal que puede especificar:

Parámetro	Descripción
messageContent	El texto de la exención de responsabilidad legal.
messageTitle	El título de la exención de responsabilidad legal.
acceptLabel	El texto del botón en el que el usuario hace clic para aceptar la exención de responsabilidad legal.
rejectLabel	El texto del botón en el que el usuario hace clic para rechazar la exención de responsabilidad legal. El usuario no puede iniciar sesión sin aceptar la exención de responsabilidad. Si el usuario rechaza la exención de responsabilidad legal, vuelve a aparecer la pantalla de inicio de sesión.
showOnce	Especifica que la exención de responsabilidad legal aparece cuando el usuario inicia sesión por primera vez. El valor predeterminado es <code>true</code> . Establezca el valor en <code>false</code> si desea que aparezca la exención de responsabilidad legal cada vez que el usuario intente iniciar sesión.
enabledApplications	Las aplicaciones que muestran la exención de responsabilidad legal cuando el usuario intenta iniciar sesión. Puede usar los siguientes valores: <ul style="list-style-type: none">- <code>e360</code>. Especifica la aplicación Data Director.- <code>provisioning</code>. Especifica la herramienta de aprovisionamiento.- <code>hubconsole</code>. Especifica la consola del concentrador.

Read Legal Message Configuration

La API REST Read Legal Message Configuration devuelve la exención de responsabilidad legal de la página de inicio de sesión actual.

La API usa el método GET.

URL de solicitud

La URL de Read Legal Message Configuration tiene el siguiente formato:

```
http://<host>:<port>/<context>/<path>
```

Realice la siguiente solicitud HTTP GET a la URL Read Legal Message Configuration:

```
GET http://localhost:8080/cmx/ui/legal/config
```

Ejemplo de solicitud de API

La siguiente solicitud devuelve las propiedades del tema de color actual.

```
GET http://localhost:8080/cmx/ui/legal/config
Content-Type: application/json
```

Ejemplo de respuesta de API

La API devuelve un código de respuesta 200 OK después de recuperar correctamente la exención de responsabilidad legal de la página de inicio de sesión actual.

Por ejemplo, puede recibir la siguiente respuesta:

```
{
  "messageContent": "The use of this systems may be monitored and recorded. If you
access this system, you expressly consent to such monitoring and the use of the
security policies enforced by the company.",
  "messageTitle": "Legal Disclaimer",
  "acceptLabel": "Accept",
  "rejectLabel": "Reject",
  "showOnce": true,
  "enabledApplications": [
    "e360",
    "provisioning",
    "hubconsole"
  ]
}
```

Establecer la configuración de mensajes legales

La API REST Set Legal Message Configuration crea o actualiza la exención de responsabilidad legal de la página de inicio de sesión. Utilice el formato JSON para enviar datos en el cuerpo de la solicitud.

La API usa el método POST.

URL de solicitud

La URL Set Legal Message Configuration tiene el siguiente formato:

```
http://<host>:<port>/<context>/<path>
```

Realice la siguiente solicitud HTTP POST para establecer la URL de configuración de mensajes legales:

```
POST http://localhost:8080/cmx/ui/legal/config
```

Ejemplo de solicitud de API

El siguiente cuerpo de solicitud contiene un ejemplo de exención de responsabilidad legal:

```
{
  "messageContent": "The use of this systems may be monitored and recorded. If you
access this system, you expressly consent to such monitoring and the use of the
security policies enforced by the company.",
  "messageTitle": "Legal Disclaimer",
  "acceptLabel": "Accept",
  "rejectLabel": "Reject",
  "showOnce": true,
```

```
    "enabledApplications": [
      "e360",
      "provisioning",
      "hubconsole"
    ]
  }
}
```

Ejemplo de respuestas de API

La API devuelve un código de respuesta 200 OK después de crear o actualizar correctamente la exención de responsabilidad legal de la página de inicio de sesión. El cuerpo de la respuesta está vacío.

Eliminar la configuración de mensajes legales

La API REST Delete Legal Message Configuration elimina los valores de configuración de la exención de responsabilidad legal de la página de inicio de sesión.

La API usa el método DELETE.

URL de solicitud

La URL de Delete Legal Message Configuration tiene el siguiente formato:

```
http://<host>:<port>/<context>/<path>
```

Realice la siguiente solicitud HTTP DELETE a la URL de Delete Legal Message Configuration:

```
DELETE http://<host>:<port>/cmx/ui/legal/config
```

Ejemplo de solicitud de API

La siguiente solicitud de ejemplo elimina la configuración de exención de responsabilidad legal para la página de inicio de sesión:

```
DELETE http://localhost:8080/cmx/ui/legal/config
```

Ejemplo de respuesta de API

El usuario puede iniciar sesión utilizando el procedimiento de inicio de sesión estándar. La API devuelve un código de respuesta 200 OK después de eliminar con éxito la configuración de exención de responsabilidad legal para la página de inicio de sesión. El cuerpo de la respuesta está vacío.

CAPÍTULO 5

Llamadas del servicio de entidad de negocio de SOAP

Este capítulo incluye los siguientes temas:

- [Descripción de las llamadas SOAP para servicios de entidad de negocio, 275](#)
- [Método de autenticación, 276](#)
- [Cookies de autenticación para el inicio de sesión desde aplicaciones de terceros, 276](#)
- [Archivo de lenguaje de descripción de servicios web, 278](#)
- [URL de SOAP, 278](#)
- [Respuestas y solicitudes SOAP, 279](#)
- [Visualización de parámetros de entrada y salida, 280](#)
- [Referencia de API de SOAP, 281](#)
- [Respuestas y solicitudes SOAP de ejemplo, 282](#)

Descripción de las llamadas SOAP para servicios de entidad de negocio

Las llamadas de extremo de Protocolo simple de acceso a objetos (SOAP) hacen que todos los servicios de entidad de negocio estén disponibles como servicios web. Puede realizar llamadas SOAP para crear, eliminar, actualizar y buscar registros en una entidad de negocio. Puede realizar operaciones como fusionar, anular la fusión y hacer coincidir registros. También puede hacer llamadas SOAP para crear, actualizar, buscar y realizar tareas.

Los extremos de SOAP para servicios de entidad de negocio utilizan el UsernameToken de Seguridad de servicios web (WS-Security) para autenticar a los usuarios.

Nota: Antes de que utilice las API de SOAP para llamar a los servicios de entidad de negocio, valide el Almacén de referencias operativas.

Método de autenticación

Todas las llamadas SOAP a los servicios de entidad de negocio requieren que el usuario se autentique. Proporcione el nombre de usuario y la contraseña en el encabezado de mensaje SOAP de la solicitud de servicio web.

El elemento de encabezado SOAP Security contiene los datos relacionados con la seguridad. El elemento Security contiene el elemento UsernameToken que incluye los siguientes elementos secundarios:

nombre de usuario

El nombre de usuario asociado al token.

contraseña

La contraseña para el nombre de usuario asociado al token.

Envíe el nombre de usuario y la contraseña del elemento UsernameToken.

En el ejemplo siguiente se muestra el elemento de encabezado Security del mensaje SOAP:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:urn="urn:cs-ors.informatica.mdm">
  <soapenv:Header>
    <Security xmlns="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd">
      <UsernameToken>
        <Username>admin</Username>
        <Password>admin</Password>
      </UsernameToken>
    </Security>
  </soapenv:Header>
  <soapenv:Body>
    .....
  </soapenv:Body>
</soapenv:Envelope>
```

Cookies de autenticación para el inicio de sesión desde aplicaciones de terceros

Utilice cookies de autenticación para autenticar a los usuarios de MDM Hub y llamar a servicios de entidad de negocio desde aplicaciones de terceros, como Postman. Con el uso de cookies de autenticación, no es necesario codificar el nombre de usuario y la contraseña.

El proceso para utilizar cookies de autenticación incluye los siguientes pasos:

1. Utilice las credenciales de un usuario autenticado para obtener una cookie de autenticación.
2. Guarde la cookie de autenticación.
3. Utilice la cookie de autenticación para llamar a las API SOAP.

Todas las sesiones de MDM utilizan un token CSRF de Informatica (ICT). Aparece un ICT en el encabezado HTTP.

Cuando un usuario solicita un ID de sesión, la respuesta devuelve un ICT como parte del encabezado HTTP. Para las aplicaciones externas, hay una URL para la autenticación que pueden usar las aplicaciones externas. Esta URL devuelve una cookie de autenticación y un ICT que se encuentra en el encabezado. La cookie de autenticación es un ID de sesión de MDM.

Toda llamada de servicios de entidad de negocio posterior debe incluir la cookie de autenticación y el ICT. Las solicitudes posteriores pueden utilizar el ID de sesión de MDM y el ICT hasta que expire la sesión.

Obtener la cookie de autenticación y el token CSRF de Informatica

El siguiente ejemplo muestra cómo obtener la cookie de autenticación y el token CSRF de Informatica (ICT):

```
POST: https://<host>:<port>/cmx/auth/<database ID>

Authorization: No Auth

Body:
{
  "username" : "admin",
  "password" : {
    "encrypted" : false,
    "password" : "admin"
  },
  "userInfo" : {
    "sessionTimeout":300,
    "role":"Manager"
  }
}
```

El siguiente ejemplo muestra cómo aparece el ICT en el encabezado:

```
ICT: 0ffa3b95c67e111b9c14c140a70273a15db266993d0c763d555135de4c4d6110
```

El siguiente ejemplo muestra cómo aparece la cookie de autenticación en la respuesta:

```
Name: mdmsessionid
Value: vppOTOhJLKMjYJf7Diil
Domain: 10.xx.xx.xx
Path: /cmx
Expires: Session
HttpOnly: true
Secure: false
```

Ejemplo de SOAP con cookie de autenticación y token CSRF de Informatica

El siguiente ejemplo muestra cómo utilizar una cookie de autenticación y un token CSRF de Informatica (ICT) en el encabezado de su solicitud de API:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:urn="urn:cs-ors.informatica.mdm">
  <soapenv:Header/>
  <soapenv:Body>
    <urn:listAssignableUsers>
      <!--Optional:-->
      <urn:parameters>
        <!--Optional:-->
        <urn:taskType>Update</urn:taskType>
        <!--Optional:-->
        <urn:businessEntity>Person</urn:businessEntity>
      </urn:parameters>
    </urn:listAssignableUsers>
  </soapenv:Body>
</soapenv:Envelope>

Headers:
Cookie: mdmsessionid: vppOTOhJLKMjYJf7Diil
ICT: 0ffa3b95c67e111b9c14c140a70273a15db266993d0c763d555135de4c4d6110
```

Archivo de lenguaje de descripción de servicios web

Los archivos de lenguaje de descripción de servicios web (WSDL) contienen las descripciones XML de los servicios web, los formatos de las respuestas y las solicitudes SOAP y todos los parámetros. MDM Hub genera un archivo WSDL para cada Almacén de referencias operativas.

Los archivos WSDL para cada Almacén de referencias operativas se encuentran en la siguiente ubicación:

```
http://<host>:<puerto>/cmx/csfiles
```

La siguiente imagen muestra la ubicación en la que puede descargar el archivo WSDL para los almacenes de referencias operativas:

localhost-rome-TCR_HUB

[cs-rest.xsd \(urn:cs-rest.informatica.mdm\)](#)

[cs-base.xsd \(urn:cs-base.informatica.mdm\)](#)

[co-base.xsd \(urn:co-base.informatica.mdm\)](#)

[co-types.xsd \(urn:co-types.informatica.mdm\)](#)

[D&BDetailedCompanyProfile_0.xsd \(http://www.strikeiron.com\)](#)

[D&BDetailedCompanyProfile_1.xsd \(http://services.dnb.com/FirmographicsProductServiceV2.0\)](#)

[D&BDetailedCompanyProfile_2.xsd \(http://ws.strikeiron.com\)](#)

[D&BOnDemandEntitySearch_1.xsd \(http://services.dnb.com/CompanyServiceV2.0\)](#)

[D&BOnDemandEntitySearch_0.xsd \(http://www.strikeiron.com\)](#)

[D&BOnDemandEntitySearch_2.xsd \(http://ws.strikeiron.com\)](#)

[co-ors.xsd \(urn:co-ors.informatica.mdm\)](#)

[cs-ors.xsd \(urn:cs-ors.informatica.mdm\)](#)

[co-meta.xsd \(urn:co-meta.informatica.mdm\)](#)

[task-base.xsd \(urn:task-base.informatica.mdm\)](#)

[localhost-rome-TCR_HUB.wsdl](#)

localhost-rome-CMX_ORIS

[cs-rest.xsd \(urn:cs-rest.informatica.mdm\)](#)

[cs-base.xsd \(urn:cs-base.informatica.mdm\)](#)

[co-base.xsd \(urn:co-base.informatica.mdm\)](#)

[co-types.xsd \(urn:co-types.informatica.mdm\)](#)

[co-ors.xsd \(urn:co-ors.informatica.mdm\)](#)

[cs-ors.xsd \(urn:cs-ors.informatica.mdm\)](#)

[co-meta.xsd \(urn:co-meta.informatica.mdm\)](#)

[task-base.xsd \(urn:task-base.informatica.mdm\)](#)

[localhost-rome-CMX_ORIS.wsdl](#)

Haga clic en el vínculo para descargar el archivo WSDL para los almacenes de referencias operativas disponibles.

URL de SOAP

Una URL de SOAP es la dirección que utiliza para conectarse al servidor de SOAP.

Una URL de SOAP tiene la siguiente sintaxis:

```
http://<host>:<puerto>/<contexto>/<ID de base de datos>
```

La URL tiene los siguientes campos:

host

El host que ejecuta la base de datos.

puerto

Número de puerto que utiliza el servicio de escucha de la base de datos.

contexto

El contexto siempre es `cmx/services/BEServices`.

ID de base de datos

El ID de ORS como está registrado en la herramienta de bases de datos de la Consola del concentrador.

El siguiente ejemplo muestra una URL de SOAP:

```
http://localhost:8080/cmx/services/BEServices/localhost-orcl-DS_UI1
```

Respuestas y solicitudes SOAP

Utilice el formato de mensaje XML SOAP para enviar solicitudes a través de un cliente SOAP al servicio de entidad de negocio y para recibir respuestas del servicio de entidad de negocio al cliente. El formato de respuesta y solicitud SOAP es el mismo.

El mensaje SOAP contiene los siguientes elementos:

Sobre

Define el inicio y el final del mensaje.

Encabezado

Opcional. Contiene atributos adicionales, como los detalles de autenticación para procesar el mensaje. Si el elemento Header está presente, debe ser el primer elemento secundario del elemento Envelope.

Cuerpo

Contiene los datos XML que procesa el cliente o el servicio web.

Un mensaje SOAP tiene el siguiente formato:

```
<?xml version="1.0"?>
<env:Envelope xmlns:env="http://schemas.xmlsoap.org/soap/envelope/" >

  <env:Header>
  </env:Header>

  <env:Body>
  </env:Body>

</env:Envelope>
```

Una solicitud SOAP tiene el siguiente formato:

```
POST /<host>:<port>/<context>/<database ID> HTTP/1.0
Content-Type: text/xml; charset=utf-8

<?xml version="1.0"?>
<env:Envelope xmlns:env="http://schemas.xmlsoap.org/soap/envelope/" >

  <env:Header>
  </env:Header>

  <env:Body>
  </env:Body>

</env:Envelope>
```

Una respuesta SOAP tiene el siguiente formato:

```
HTTP/1.0 200 OK
Content-Type: text/xml; charset=utf-8

<?xml version="1.0"?>
<env:Envelope xmlns:env="http://schemas.xmlsoap.org/soap/envelope/" >

  <env:Header>
  </env:Header>

  <env:Body>
  </env:Body>

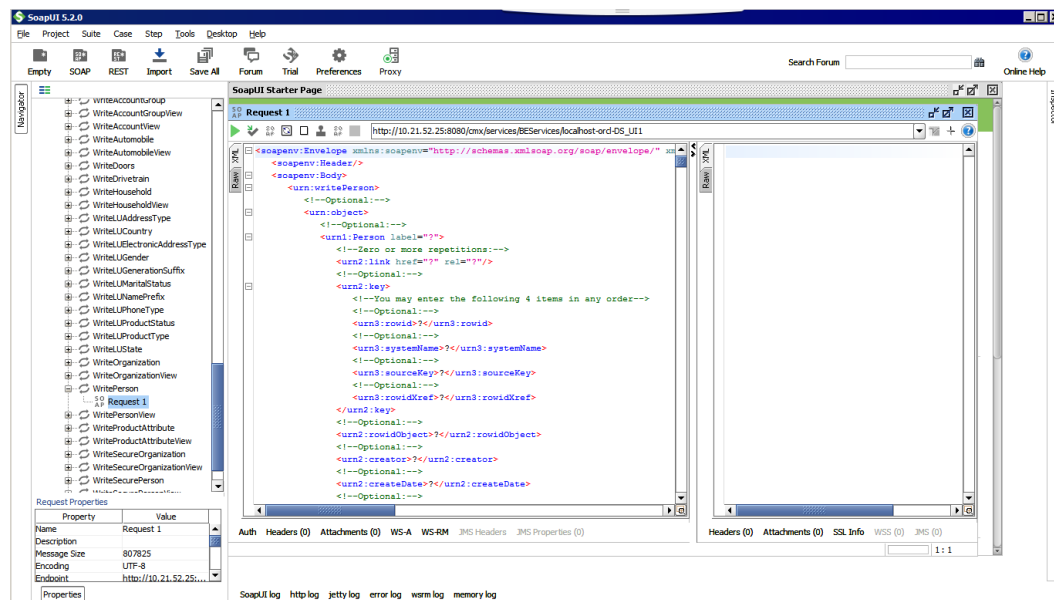
</env:Envelope>
```

Visualización de parámetros de entrada y salida

Puede utilizar una herramienta de pruebas funcional, como SoapUI, para ver los parámetros de entrada y salida de la API de SOAP.

Cree un proyecto de SOAP e importe el archivo WSDL en el proyecto. Las operaciones que puede realizar mediante los servicios de entidad de negocio aparecen como nodos en SoapUI. Cada operación tiene un formato de mensaje de respuesta y de solicitud. SoapUI crea un ejemplo de solicitud para cada operación al importar el archivo WSDL.

Abra el proyecto y haga doble clic en una solicitud para abrir el editor de solicitudes. La siguiente imagen muestra los parámetros de entrada en SoapUI para la API de SOAP WritePerson:



Referencia de API de SOAP

La referencia de la API de SOAP para servicios de entidad de negocio enumera las API de SOAP y proporciona una descripción de cada una de ellas. Consulte también el archivo WSDL para ver las descripciones de los servicios de entidad de negocio.

Utilice las siguientes API de SOAP para realizar operaciones en las entidades de negocio:

Obtener metadatos

Devuelve la estructura de datos de una entidad de negocio.

Enumerar registros

Devuelve la lista de valores de búsqueda o valores de clave externa.

Leer registros

Devuelve los detalles de un registro raíz de la entidad de negocio.

Crear registros

Crea un registro en la entidad de negocio especificada.

Actualizar registros

Actualiza el registro raíz especificado y sus registros secundarios.

Eliminar registros

Elimina un registro raíz en una entidad de negocio.

Buscar registros

Busca un valor de cadena en una entidad de negocio raíz que admite búsquedas y devuelve los registros raíz que coinciden con los criterios de búsqueda.

Vista previa de promoción

Devuelve una vista previa de un registro resultante si promueve los cambios pendientes en función del ID de interacción de la solicitud de cambios.

Promover

Promueve todos los cambios pendientes que se realicen en un registro en función del ID de interacción de la solicitud de cambio.

Eliminar promoción

Elimina todos los cambios pendientes que realice en un registro en función del ID de interacción de la solicitud de cambio.

Vista previa de fusión

Devuelve una vista previa de un registro raíz consolidado si fusiona dos o más registros raíz.

Fusionar registros

Combina dos o más registros raíz para crear un único registro consolidado.

Anular fusión de registros

Anula la fusión de un registro raíz en registros raíz individuales que existían antes de que se fusionaran los registros.

Obtener registros relacionados

Devuelve una lista de los registros relacionados en función de las relaciones que configure en el administrador de jerarquía.

Leer registros coincidentes

Devuelve los registros que coinciden con un registro raíz especificado.

Actualizar registros coincidentes

Crea o actualiza un registro en la tabla de coincidencia.

Eliminar registros coincidentes

Elimina los registros coincidentes de la tabla de coincidencias.

Obtener metadatos de BPM

Devuelve los tipos de tarea y dos indicadores que especifican si el adaptador del flujo de trabajo puede realizar el servicio para obtener el linaje de tarea y los servicios de administración.

Lista de tareas

Devuelve una lista de tareas del flujo de trabajo.

Leer tarea

Devuelve los detalles de una tarea.

Crear tarea

Crea una tarea e inicia un flujo de trabajo.

Actualizar tarea

Actualiza una tarea individual.

Ejecutar acción de tarea

Realiza una acción de tarea y restablece la tarea en el flujo de trabajo para que se siga procesando.

Lista de usuarios asignables

Devuelve una lista de usuarios a los que se les pueden asignar las tareas que pertenecen a un tipo de tarea.

Tarea completada

Cierra un flujo de trabajo de tareas después de completar todas las tareas del flujo de trabajo.

Respuestas y solicitudes SOAP de ejemplo

Para comprender las API de SOAP, revise las siguientes solicitudes y respuestas SOAP de ejemplo.

Lista de usuarios asignables Muestra

La siguiente solicitud SOAP de ejemplo recupera una lista de los usuarios asignables:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:urn="urn:cs-ors.informatica.mdm">
  <soapenv:Header>
    <Security xmlns="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-
secext-1.0.xsd">
      <UsernameToken>
        <Username>admin</Username>
        <Password>admin</Password>
      </UsernameToken>
    </Security>
  </soapenv:Header>
  <soapenv:Body>
```

```

    <urn:listAssignableUsers>
      <!--Optional:-->
      <urn:parameters>
        <!--Optional:-->
        <urn:taskType>Update</urn:taskType>
        <!--Optional:-->
        <urn:businessEntity>Person</urn:businessEntity>
      </urn:parameters>
    </urn:listAssignableUsers>
  </soapenv:Body>
</soapenv:Envelope>

```

La siguiente respuesta SOAP de ejemplo recupera una lista de los usuarios asignables:

```

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance">
  <soapenv:Body>
    <ns6:listAssignableUsersReturn xmlns:ns1="urn:cs-base.informatica.mdm"
xmlns:ns2="urn:co-base.informatica.mdm" xmlns:ns3="urn:co-ors.informatica.mdm"
xmlns:ns4="urn:co-meta.informatica.mdm" xmlns:ns5="urn:task-base.informatica.mdm"
xmlns:ns6="urn:cs-ors.informatica.mdm">
      <ns6:object>
        <ns1:users>
          <user>
            <userName>admin</userName>
          </user>
        </users>
        <ns1:roles/>
      </ns6:object>
    </ns6:listAssignableUsersReturn>
  </soapenv:Body>
</soapenv:Envelope>

```

Entidad de negocio de lectura Muestra

La siguiente solicitud SOAP de ejemplo recupera información para una entidad de negocio de persona:

```

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:urn="urn:cs-ors.informatica.mdm" xmlns:urn1="urn:cs-base.informatica.mdm">
  <soapenv:Header/>
  <soapenv:Body>
    <urn:readEntity>
      <urn:parameters>
        <urn:coFilter>
          <urn1:object name="Person" depth="3" suppress="false">
            <urn1:key>
              <urn1:rowid>931</urn1:rowid>
            </urn1:key>
            <urn1:fields>firstName,lastName,middleName,gender</urn1:fields>
            <urn1:recordState>ACTIVE</urn1:recordState>
            <urn1:recordState>PENDING</urn1:recordState>
            <urn1:contentMetadata>XREF</urn1:contentMetadata>
          </urn1:object>
        </urn:coFilter>

        <urn:outputFilter>
          <urn1:object name="PersonView" depth="2" suppress="false">
            </urn1:object>
          </urn:outputFilter>
        </urn:parameters>
      </urn:readEntity>
    </soapenv:Body>
  </soapenv:Envelope>

```

Entidad de negocio de creación Muestra

La siguiente solicitud SOAP de ejemplo crea un registro de entidad de negocio de persona:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:urn="urn:cs-ors.informatica.mdm" xmlns:urn1="urn:co-base.informatica.mdm"
xmlns:urn2="urn:cs-base.informatica.mdm" xmlns:urn3="urn:co-ors.informatica.mdm"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Header/>
  <soapenv:Body>
    <urn:writeEntity>
      <urn:object xmlns:urn3="urn:co-ors.informatica.mdm" xsi:type="urn3:Person">
        <urn3:Person label="Person">
          <urn3:partyType>Person</urn3:partyType>
          <urn3:lastName>test1</urn3:lastName>
          <urn3:middleName>test1</urn3:middleName>
          <urn3:firstName>test1</urn3:firstName>
        </urn3:Person>
        <urn3:changeSummary create="#/urn:writeEntity/urn:object/urn1:Person"
logging="false" xmlns:sdo="commonj.sdo">
          <urn:object sdo:ref="#/urn:writeEntity/urn:object"
sdo:unset="Person"/>
          </urn3:changeSummary>
        </urn:object>
        <urn:parameters>
          <urn:systemName>Admin</urn:systemName>
        </urn:parameters>
      </urn:writeEntity>
    </soapenv:Body>
  </soapenv:Envelope>
```

CAPÍTULO 6

Servicios de cálculos de BVT y registros de referencias cruzadas

Este capítulo incluye los siguientes temas:

- [Descripción general de los servicios de cálculos de BVT y registros de referencias cruzadas, 285](#)
- [Obtención de datos de referencias cruzadas e investigación de cálculos de BVT, 285](#)
- [Filtrado y paginación de respuestas, 290](#)
- [Establecer la mejor versión de confianza, 290](#)

Descripción general de los servicios de cálculos de BVT y registros de referencias cruzadas

Puede usar los servicios para registros de referencias cruzadas y cálculos de la mejor versión de confianza (BVT) para obtener información sobre cómo los datos de origen conforman el registro principal.

Puede usar estos servicios para realizar las siguientes tareas:

- Recopilar información sobre los datos de origen
- Determinar cómo se ha establecido la mejor versión de confianza
- Reemplazar los cálculos de BVT para garantizar que los registros principales contienen la mejor versión de confianza

Obtención de datos de referencias cruzadas e investigación de cálculos de BVT

Los registros principales de MDM Hub mantienen la mejor versión de confianza (BVT). MDM Hub consolida los datos más fiables de varios sistemas de origen en cada registro principal para conseguir la mejor versión de confianza. MDM Hub almacena los datos de origen en registros de referencias cruzadas. Puede usar servicios de entidad de negocio para leer los datos de los registros de referencias cruzadas y determinar el método de cálculo de BVT.

Obtener registros de referencias cruzadas

Puede usar un servicio de entidad de negocio para obtener los registros de referencias cruzadas para un registro principal concreto.

La dirección URL de la API de REST para la obtención de registros de referencias cruzadas tiene el siguiente formato:

```
http://<host>:<port>/<context>/<database ID>/<business entity>/<row ID>?
contentMetadata=XREF
```

En la siguiente solicitud de ejemplo se recuperan los registros de referencias cruzadas para un registro de entidad de negocio de persona con el ID de fila 123:

```
GET http://localhost:8080/cmx/cs/localhost-orcl-ORS/Person/123?contentMetadata=XREF
```

Respuesta generada al obtener registros de referencias cruzadas

En el siguiente ejemplo se muestran los registros de referencias cruzadas que devuelve el registro de persona con el ID de fila 123:

```
GET /Person/123?contentMetadata=XREF
```

```
{
  "firstName": "Joe",
  "lastName": "Smith",
  "XREF": {
    "item": [
      {
        "rowidXref": 111,
        "firstName": "Joe",
        "lastName": "Smith",
      },
      {
        "rowidXref": 222,
        "firstName": "John",
        "lastName": "Smith"
      }
    ]
  }
}
```

Determinar colaboradores del registro principal

Puede usar servicios de entidad de negocio para ver qué campo de registro de referencias cruzadas colabora en el registro principal. El registro que colabora en cada campo se identifica por el ID de fila del registro de referencias cruzadas.

La dirección URL de la API de REST para determinar los colaboradores del registro principal tiene el siguiente formato:

```
http://<host>:<port>/<context>/<database ID>/<business entity>/<row ID>?
contentMetadata=BVT
```

En la siguiente solicitud de ejemplo se recupera la información de BVT para un registro de persona con un ID de fila 123:

```
GET http://localhost:8080/cmx/cs/localhost-orcl-ORS/Person/123?contentMetadata=BVT
```

Respuesta generada al determinar los colaboradores del registro principal

En el siguiente ejemplo se muestra qué registro de referencias cruzadas ha colaborado en cada campo del registro principal:

```
{
  "firstName": "Joe",
  "lastName": "Smith",
  "BVT": {
    "firstName": {
      "rowidXref": 111
    },
    "lastName": {
      "rowidXref": 222
    }
  }
}
```

Obtener las puntuaciones de confianza de campos de registro de referencias cruzadas colaboradores

Puede usar servicios de entidad de negocio para obtener las puntuaciones de confianza de los campos de registro de referencias cruzadas que colaboran en el registro principal.

La dirección URL de la API de REST para determinar los colaboradores y obtener las puntuaciones de confianza tiene el siguiente formato:

```
http://<host>:<port>/<context>/<database ID>/<business entity>/<row ID>?
contentMetadata=TRUST
```

En la siguiente solicitud de ejemplo se indican las puntuaciones de confianza para un registro de persona con el ID de fila 123:

```
GET http://localhost:8080/cmx/cs/ors/Person/123?contentMetadata=TRUST
```

Respuesta generada al obtener las puntuaciones de confianza de campos de registro de referencias cruzadas colaboradores

En el siguiente ejemplo de respuesta se indican las puntuaciones de confianza para cada campo de un registro de entidad de negocio de persona:

```
{
  "firstName": "John",
  "lastName": "Smith",
  "TRUST": {
    "firstName": {
      "score": 75.0,
      "valid": true,
      "trustSetting" :{
        // custom settings
      }
    },
  }
}
```

Obtención de las puntuaciones de confianza de todos los campos de registro de referencias cruzadas

Use la API de REST con el parámetro contentMetadata establecido en XREF_TRUST para obtener las puntuaciones de confianza y los porcentajes de degradación de todos los campos de registro de referencias cruzadas.

Dirección URL de la solicitud para la API de REST de lectura para determinar los colaboradores y obtener las puntuaciones de confianza:

```
http://<host>:<port>/<context>/<database ID>/<business entity>/<row ID>?
contentMetadata=XREF_TRUST
```

En la siguiente solicitud de ejemplo se recuperan los datos de referencias cruzadas para un registro de persona con el ID de fila 123:

```
GET http://localhost:8080/cmx/cs/localhost-orcl-ORS/Person/123?contentMetadata=XREF_TRUST
```

Respuesta generada al obtener las puntuaciones de confianza de todos los campos de registro de referencias cruzadas

En el siguiente ejemplo se muestran las puntuaciones de confianza y los porcentajes de degradación de todos los campos de registro de referencias cruzadas para una entidad de negocio de persona:

```
{
  "firstName": "Sergey",
  "lastName": "Ivanov",
  "XREF": {
    "item": [{
      "rowidXref": 111,
      "firstName": "Sergey",
      "lastName": "Petrov",
      "TRUST": {
        "firstName": {
          "score": 75.0,
          "valid": true
        },
        "lastName": {
          "score": 60.0,
          "valid": false,
          "downgradePerCent": 20.0
        }
      }
    }, {
      "rowidXref": 222,
      "firstName": "Sergey",
      "lastName": "Ivanov",
      "TRUST": {
        "firstName": {
          "score": 10.0,
          "valid": true
        },
        "lastName": {
          "score": 80.0,
          "valid": true
        }
      }
    }
  ]
}
```


Obtener información sobre los sistemas de origen

Puede obtener información sobre los sistemas de origen de los que proceden los datos de referencias cruzadas y determinar el número de registros de referencias cruzadas en los que colaboran los sistemas de origen para todo el registro, para cada nodo o para cada registro.

Los siguientes parámetros se pueden especificar en la solicitud:

describe

Se establece en `true` para devolver la descripción del sistema de origen. Puede ser `true` o `false`. El valor predeterminado es `false`.

aggregate

Define el nivel en el que se devuelve la información de los sistemas de origen. Puede ser `ENTITY`, `NODE` y `RECORD`. El valor predeterminado es `ENTITY`.

recordStates

Especifica el estado de registro para el que se devuelven registros. Puede ser `ACTIVE`, `PENDING` o `DELETED`. El valor predeterminado es `ACTIVE`.

compact

Si se establece en `no`, especifica que los datos de nivel de entidad se devuelven al especificar el parámetro de agregado con `ENTITY` y otros niveles de agregado. Puede ser `yes` o `no`. Solo para solicitudes de API de REST. El valor predeterminado es `yes`.

Ejemplo de obtención de información sobre los sistemas de origen

En la siguiente solicitud de ejemplo se obtiene información sobre los sistemas de origen en el nivel de entidad y el nivel de nodo para la entidad de negocio de persona con el ID de fila 123:

```
GET http://localhost:8080/cmx/cs/ors/Person/123?
action=getSourceSystems&aggregate=ENTITY,NODE&compact=no
```

En la siguiente solicitud de ejemplo se obtienen información sobre los sistemas de origen en el nivel de registro y descripciones de los sistemas de origen para la entidad de negocio de persona con el ID de fila 456:

```
GET http://localhost:8080/cmx/cs/ors/Person/123/Address/456?
action=getSourceSystems&aggregate=ENTITY,NODE&compact=no
```

Respuesta generada al obtener información sobre los sistemas de origen

En el siguiente ejemplo se muestra información de nivel de entidad y de nivel de nodo para una entidad de negocio de persona:

```
{
  {
    "name": "Admin",
    "xrefCount": 120
  }
],
Person: {
  "rowidObject": "456",
  "sourceSystem":
  {
    {
      "name": "Admin",
      "xrefCount": 30
    }
  ]
}
}
```

Filtrado y paginación de respuestas

Puede seleccionar los campos que se van a devolver en la respuesta, filtrar los resultados por varios criterios y paginar los resultados.

Ejemplos de solicitudes de filtrado

En la siguiente tabla se incluyen solicitudes de ejemplo para la entidad de negocio de persona con diversos filtros aplicados y una descripción de los resultados que se devuelven en la respuesta:

Solicitud	Descripción de los resultados devueltos
/Person/123	Todos los campos definidos por el usuario
/Person/123?readSystemFields=true	Todos los campos definidos por el usuario y todos los campos del sistema
/Person/123?fields=firstName	Un solo campo definido por el usuario
/Person/123?fields=updatedAt	Un solo campo del sistema
Person/123?fields=firstName,updatedAt	Un solo campo definido por el usuario y un solo campo del sistema
/Person/123?fields=firstName&readsystemFields=true	Un solo campo definido por el usuario y todos los campos del sistema

Establecer la mejor versión de confianza

Una vez que un gestor de datos haya investigado los datos de origen de los registros de referencias cruzadas, puede realizar ajustes en el modo de consolidación de los datos de origen para garantizar que el registro maestro representa la mejor versión de confianza.

Puede usar servicios de entidad de negocio para realizar las siguientes acciones con el fin de establecer la mejor versión de confianza:

- Actualizar la configuración de confianza
- Eliminar los datos de origen no coincidentes
- Seleccionar el campo colaborador correcto
- Escribir el valor correcto en el registro principal

Seleccionar el campo colaborador correcto

Si el campo con la puntuación de confianza más alta no contiene la mejor versión de confianza, un gestor de datos puede seleccionar el campo que contiene los datos correctos para aportarlos al registro principal.

La dirección URL y el cuerpo de la solicitud para seleccionar el campo colaborador correcto según el nombre del sistema y la clave de origen tienen el siguiente formato:

```
POST http://<host>:<port>/<context>/<database ID>/<business entity>/<row ID>?
systemName=<source system name>
```

```

{
  BVT: {
    <field name>: {
      systemName: "<source system name>",
      sourceKey: "<source key>"
    }
  }
}

```

La dirección URL y el cuerpo de la solicitud para seleccionar el campo colaborador correcto según el ID de registro de referencias cruzadas tienen el siguiente formato:

```

POST http://<host>:<port>/<context>/<database ID>/<business entity>/<row ID>?
systemName=<source system name>
{
  BVT: {
    <field name>: {
      rowidXref: "<row ID>"
    }
  }
}

```

Ejemplo de selección del campo colaborador correcto

La siguiente dirección URL y el cuerpo de la solicitud permiten seleccionar el campo de nombre del registro de referencias cruzadas del sistema de origen de ventas con una clave de origen 0001 para colaborar en el registro principal:

```

POST http://localhost:8080/cmx/cs/localhost-orcl-ORS/Person/123?systemName=Admin
{
  BVT: {
    firstName: {
      systemName: "Sales",
      sourceKey: "0001"
    }
  }
}

```

La siguiente dirección URL y el cuerpo de la solicitud permiten seleccionar el campo de nombre del registro de referencias cruzadas con el ID de fila 789 para colaborar en el registro principal:

```

POST http://localhost:8080/cmx/cs/localhost-orcl-ORS/Person/123?systemName=Admin
{
  BVT: {
    firstName: {
      rowidXref: "789"
    }
  }
}

```

Escribir el valor correcto en el registro principal

Si usa una llamada de servicio de entidad de negocio para escribir un valor correcto en el registro principal, también puede establecer la configuración de confianza para el valor. Si no especifica la configuración de confianza, MDM Hub usa la configuración del sistema del administrador.

La dirección URL y el cuerpo de la solicitud para escribir el valor correcto con la configuración de confianza del administrador tienen el siguiente formato:

```

http://<host>:<port>/<context>/<database ID>/<business entity>/<row ID>?
systemName=<source system providing the correct value>{
  "<field name>": "<correct value>",
  "$original": {
    "<field name>": "<current value>",
  },
}

```

```

"TRUST": {
  "<field name>": {
    "trustSetting" : {
      custom: false
    }
  }
}

```

La dirección URL y el cuerpo de la solicitud para escribir el valor correcto con la configuración de confianza definida tienen el siguiente formato:

```

http://<host>:<port>/<context>/<database ID>/<business entity>/<row ID>?
systemName=<source system providing the correct value>{
  "<field name>": "<correct value>",
  "$original": {
    "<field name>": "<current value>",
  },
  "TRUST": {
    "firstName": {
      "trustSetting" : {
        custom: true, // if custom=true, all other trustSetting fields
                      //are mandatory. If they are not set,
                      //the service will return an error.
        minimumTrust: <minimum trust percent>,
        maximumTrust: <maximum trust percent>,
        timeUnit: "<units for measuring trust decay>",
        maximumTimeUnits: <number of units>,
        graphType: "<name of graph type>"
      }
    }
  }
}

```

Parámetros de confianza

Puede definir los siguientes parámetros de confianza:

minimumTrust

Nivel de confianza de un valor de datos cuando es antiguo (una vez transcurrido el período de caída). Este valor debe ser menor o igual que el valor de confianza máximo.

Nota: Si los valores máximo y mínimo de confianza son iguales, la curva de caída será una línea recta y el período y el tipo de caída no tendrán efecto.

maximumTrust

Nivel de confianza de un valor de datos justo después de cambiarlo. Por ejemplo, si el sistema de origen X cambia un campo de número de teléfono de 555-1234 a 555-4321, el nuevo valor tendrá el valor máximo de confianza del sistema X para el campo de número de teléfono. Al configurar un nivel máximo de confianza relativamente alto, se asegura de que los cambios que se realicen en los sistemas de origen se aplicarán al objeto base.

timeUnit

Especifica las unidades que se utilizan para calcular el período de caída: día, semana, mes, trimestre o año.

maximumTimeUnits

Especifica el número (de días, semanas, meses, trimestres o años) que se usa para calcular el período de caída.

graphType

La caída sigue un patrón según el cual el nivel de confianza disminuye durante el período de caída. Los tipos de gráfico pueden ser uno de los siguientes patrones de caída:

Parámetro de tipo de gráfico	Descripción
LINEAR	Es la caída más simple. La caída sigue una línea recta desde el valor máximo de confianza hasta el valor mínimo de confianza.
RISL	La mayor parte de la disminución se produce al inicio del período de caída. La caída sigue una curva cóncava. Si un sistema de origen incluye este tipo de gráfico, el nuevo valor del sistema será de confianza probablemente, pero es posible que dicho valor se reemplace.
SIRL	La mayor parte del aumento se produce al final del período de caída. La caída sigue una curva convexa. Si un sistema de origen incluye este tipo de gráfico, es relativamente poco probable que otro sistema reemplace el valor en el registro principal hasta que el valor esté cerca del final de su período de caída.

Ejemplo de escritura del valor correcto en el registro principal

Ejemplo 1

Desea cambiar el nombre del registro principal de Sam Brown a John Smith. El cambio se atribuye al sistema de origen de ventas. La configuración de confianza se establece en la configuración de confianza del administrador.

En el siguiente código se muestran la dirección URL y el comando para el ejemplo 1.

```
POST http://localhost:8080/cmx/cs/localhost-orcl-ORS/Person/123?systemName=Sales
{
  "firstName": "John",
  "lastName": "Smith"
  "$original": {
    "firstName": "Sam",
    "lastName": "Brown"
  },
  "TRUST": {
    "firstName": {
      "trustSetting" : {
        custom: false
      }
    }
    "lastName": {
      "trustSetting" : {
        custom: false
      }
    }
  }
}
```

Ejemplo 2

Desea cambiar el nombre del registro principal de Sam Brown a John Smith. El cambio se atribuye al sistema de origen SFA. La configuración de confianza se establece en un valor de confianza mínimo de 60 y un valor de confianza máximo de 90, y la confianza disminuye de forma lineal durante un período de caída de tres meses.

En el siguiente código se muestran la dirección URL y el comando para el ejemplo 2.

```
POST http://localhost:8080/cmx/cs/localhost-orcl-ORS/Person/123?systemName=SFA
{
  "firstName": "John",
```

```

    "lastName": "Smith"
    "$original": {
      "firstName": "Sam",
      "lastName": "Brown"
    },
    "TRUST": {
      "firstName": {
        "trustSetting" : {
          custom: true,
          minimumTrust: 60,
          maximumTrust: 90,
          timeUnit: "Month",
          maximumTimeUnits: 3,
          graphType: "LINEAR"
        }
      }
      "lastName": {
        "trustSetting" : {
          custom: true,
          minimumTrust: 60,
          maximumTrust: 90,
          timeUnit: "Month",
          maximumTimeUnits: 3,
          graphType: "LINEAR"
        }
      }
    }
  }
}

```

Eliminar los datos de origen no coincidentes

Si un registro de referencias cruzadas está asociado de forma incorrecta a un registro principal determinado, un gestor de datos puede anular la fusión del registro de referencias cruzadas. Se crea, entonces, un nuevo registro principal a partir del registro de referencias cruzadas no fusionado.

Solo se puede anular la fusión de un único registro de referencias cruzadas en una llamada de anulación de fusión. Si es necesario anular la fusión de varios registros de referencias cruzadas, realice una llamada de anulación de fusión para cada registro de referencias cruzadas.

Si se configura un activador para un evento no fusionado, se crea una tarea de anulación de fusión. En caso contrario, se anula la fusión del registro de referencias cruzadas.

La dirección URL y el comando para anular la fusión de un registro basado en el nombre del sistema y la clave de origen tienen el siguiente formato:

```

POST http://<host>:<port>/<context>/<database ID>/<business entity>/<row ID>?
action=unmerge&systemName=<source system name>
{
  name: "<object name>",
  key: {rowid: "<rowid value>", sourcekey: "<source key>", systemName: "<source system
name>" }
}

```

La dirección URL y el comando para anular la fusión de un registro basado en el ID de registro de referencias cruzadas tienen el siguiente formato:

```

POST http://<host>:<port>/<context>/<database ID>/<business entity>/<row ID>?
action=unmerge&systemName=<source system name>
{
  name: "<object name>",
  key: {rowid: "<rowid value>", rowidXref: "<row ID of xref>"}
}

```

Ejemplo de eliminación de datos de origen no coincidentes

Ejemplo de API de REST

En el siguiente código se muestran la dirección URL y el comando para anular la fusión del registro de referencias cruzadas en el nivel secundario de un registro de dirección:

```
POST http://localhost:8080/cmx/cs/localhost-orcl-MDM_SAMPLE/Person/181921?
action=unmerge&systemName=Admin
{
  "name": "Person.Address",
  "key": {
    "rowid": "41721 ",
    "rowidXref": 41722
  }
}
```

Donde:

- El registro de referencias cruzadas para el que se va a anular la fusión tiene el ID de fila 41722.
- El ID de fila del registro principal desde el que se va a anular la fusión del registro de referencias cruzadas es 41721.
- El ID de fila del registro raíz es 181921.

Ejemplo de SOAP/EJB

En el siguiente código se muestran la dirección URL y el comando para anular la fusión del registro de referencias cruzadas en el nivel secundario de un registro de dirección:

```
<ns9:UnMerge xmlns:ns2="urn:co-base.informatica.mdm" xmlns:ns7="urn:co-
meta.informatica.mdm" xmlns:ns3="http://services.dnb.com/LinkageServiceV2.0"
xmlns:ns8="urn:task-base.informatica.mdm" xmlns:ns6="urn:co-ors.informatica.mdm"
xmlns:ns1="urn:cs-base.informatica.mdm" xmlns:ns9="urn:cs-ors.informatica.mdm"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="ns9:UnMerge">
  <ns9:parameters>
    <ns9:businessEntityKey name="Person">
      <ns1:key>
        <ns1:rowid>181921</ns1:rowid>
      </ns1:key>
    </ns9:businessEntityKey>
    <ns9:unmergeKey name="Person.TelephoneNumbers">
      <ns1:key>
        <ns1:rowid>41721 </ns1:rowid>
        <ns1:rowidXref>41722</ns1:rowidXref>
      </ns1:key>
    </ns9:unmergeKey>
    <ns9:treeUnmerge>true</ns9:treeUnmerge>
  </ns9:parameters>
</ns9:UnMerge>
```

Donde:

- El registro de referencias cruzadas para el que se va a anular la fusión tiene el ID de fila 41722.
- El ID de fila del registro principal desde el que se va a anular la fusión del registro de referencias cruzadas es 41721.
- El ID de fila del registro raíz es 181921.

Respuesta de anulación de fusión

La respuesta de anulación de fusión contiene el ID de fila del objeto base que se crea a partir del registro de referencias cruzadas sin fusionar.

Ejemplo de respuesta 1

En el siguiente ejemplo se muestra la respuesta si usa un registro de referencias cruzadas de un nodo raíz de persona.

```
{
  Person: {
    rowidObject: "7777"
  }
}
```

Ejemplo de respuesta 2

En el siguiente ejemplo se muestra la respuesta si usa un registro de referencias cruzadas de un nodo secundario de persona.

```
{
  Person: {
    Address: {
      item: [
        rowidObject: "55555"
      ]
    }
  }
}
```


CAPÍTULO 7

Compatibilidad del servicio de vinculación empresarial

Este capítulo incluye los siguientes temas:

- [Resumen, 297](#)
- [Servicios de entidad de negocio para la importación y actualización de DaaS, 297](#)
- [Configuración de la compatibilidad de la vinculación, 298](#)
- [Aplicación personalizada para la división de datos de vinculación, 298](#)

Resumen

El servicio de vinculación empresarial de Duns & Bradstreet (D&B) devuelve el elemento principal de una organización solicitada y todas sus entidades relacionadas. Puede usar el servicio de vinculación de D&B para obtener información de todas las sucursales y divisiones de una organización. Puede crear y actualizar registros con los datos del servicio de vinculación.

Puede importar los datos de vinculación empresarial en MDM Hub. Debe desarrollar una aplicación personalizada que pueda usar el servicio de vinculación del componente personalizado de proveedor de DaaS en una vista de entidad.

Necesita un servicio de entidad de negocio para importar los datos del servicio de D&B y crear un registro con los datos. En caso de producirse un cambio en los datos almacenados externamente, debe poder realizar los cambios correspondientes en su registro. D&B ofrece un servicio de supervisión para informarle de los cambios en los datos. Necesita un servicio que pueda aceptar datos antes y después de un cambio, y aplicar los cambios al registro correspondiente.

Servicios de entidad de negocio para la importación y actualización de DaaS

El servicio de entidad de negocio de importación de DaaS acepta datos del servicio de vinculación en formato XML y los convierte en un registro. El servicio de entidad de negocio de actualización de DaaS acepta datos del servicio externo en forma de dos archivos XML. Estos dos archivos XML se corresponden

con los datos antes y después de un cambio. El servicio de actualización aplica los cambios al registro correspondiente.

TEMAS RELACIONADOS

- [“Importación de DaaS” en la página 256](#)
- [“Actualización de DaaS” en la página 259](#)

Configuración de la compatibilidad de la vinculación

Con el fin de poder usar el servicio de vinculación de D&B para crear y actualizar registros, debe añadir la configuración en la herramienta de aprovisionamiento y crear una aplicación personalizada para dividir la respuesta del servicio de vinculación.

Realice las siguientes tareas para configurar la compatibilidad del servicio de vinculación de D&B:

1. Use la herramienta de aprovisionamiento para cargar WSDL para el servicio de vinculación.
2. Use la herramienta de aprovisionamiento para configurar un documento XML para la transformación de entidad de negocio y su exposición como un servicio. Cuando expone la transformación como un servicio, el proceso crea los servicios de importación y actualización de DaaS.
3. Cree una aplicación personalizada que pueda solicitar datos del servicio de vinculación y dividir la respuesta en detalles de registro y detalles de vinculación.
4. Desarrolle una interfaz de usuario que envíe la solicitud a la aplicación personalizada.

Nota: Para obtener más información sobre cómo cargar WSDL y configurar XML para la transformación de entidad de negocio, consulte el capítulo Integración de Data as a Service de la *Guía de la herramienta de aprovisionamiento de Multidomain MDM*.

Aplicación personalizada para la división de datos de vinculación

Para poder usar los servicios de vinculación de D&B, debe diseñar una aplicación personalizada que pueda dividir la información de vinculación en detalles de registro y detalles de vinculación.

La aplicación personalizada debe realizar las siguientes funciones:

1. Aceptar la solicitud para un servicio de vinculación en la vista de entidad.
2. Enviar la solicitud a D&B y recibir la respuesta.
3. Convertir la respuesta en XML.
4. Dividir la respuesta en detalles de registro y detalles de vinculación.
5. Enviar la información XML a servicios de entidad de negocio para guardarla como un registro en la base de datos.
6. Supervisar los cambios en los datos y llamar a la función de aviso de cambio en lista del servicio externo.

CAPÍTULO 8

Llamadas externas para limpiar, analizar y transformar datos

Este capítulo incluye los siguientes temas:

- [Resumen, 299](#)
- [Fases de servicio, 300](#)
- [Parámetros del método de proceso, 302](#)
- [Prácticas recomendadas para llamadas externas, 303](#)
- [Creación de llamadas externas, 304](#)
- [Ejemplo de llamada externa, 304](#)
- [Obtención de ID de fila de origen con la llamada externa MergeCO.BeforeEverything, 316](#)
- [Ejemplo: Validación y lógica personalizadas para servicios de entidad de negocio, 319](#)
- [Ejemplo: realizar una llamada al marco de integración de servicios desde una llamada externa, 324](#)

Resumen

Los proveedores externos proporcionan servicios web para limpiar, analizar y transformar datos de registro. Use los servicios web externos para la validación personalizada (por ejemplo, para comprobar si el campo de dirección está en blanco al añadir un registro). Use los servicios web externos para que la lógica personalizada transforme los datos de registro. Por ejemplo, si fusiona dos registros, puede fusionar las direcciones, pero no los números de teléfono.

Un servicio web externo expone una o varias operaciones a las que pueden llamar los servicios de entidad de negocio. Cada operación tiene un tipo de solicitud y respuesta. Los servicios de entidad de negocio envían los datos de registro con los parámetros de servicio requeridos para los servicios externos. Puede configurar llamadas a los servicios web externos para determinados pasos de la lógica de ejecución. En función de la lógica que implemente, las solicitudes salen de Data Director para actualizar los datos de registro. Es posible que los servicios externos modifiquen los datos si es necesario.

En la herramienta de aprovisionamiento, configure la entidad de negocio y los eventos para los que desea llamar al servicio externo. En la herramienta de aprovisionamiento, cargue el archivo WSDL para el servicio externo y registre el servicio SOAP y la operación. Enlace el servicio con entidades de negocio y eventos específicos.

Use el archivo WSDL del kit de recursos para comprender el servicio, las operaciones, los métodos y los tipos de datos que intercambian los métodos de servicio. El archivo `custom-logic-service.wsdl` para los

servicios web externos se encuentra en la siguiente ubicación del kit recursos: `C:\<infamdm installation directory>\hub\resourcekit\samples\BESEExternalCall\source\resources\webapp\WEB-INF\wsdl\`

El kit de recursos incluye código de ejemplo para implementar la lógica y la validación personalizadas. Si instala el kit de recursos, el archivo `bes-external-call.ear` para la lógica y la validación personalizadas de ejemplo se implementa en el servidor de aplicaciones.

Fases de servicio

Un servicio de entidad de negocio consta de múltiples fases de servicio. Puede configurar una fase de servicio para invocar una llamada externa y desencadenar un evento admitido.

Las fases de servicio se activan en el siguiente orden:

1. **BeforeEverything.** Invoca una llamada externa y activa un evento admitido antes de realizar la lógica del servicio, como validación, localización y proyección. Un caso de uso de ejemplo podría implicar la configuración de esta fase de servicio para cambiar el comportamiento de fusión predeterminado para evitar la fusión de un atributo de dos entidades de negocio. Por ejemplo, durante una fusión de dos o más registros de persona, puede evitar la fusión de los registros de números de teléfono secundarios de un registro de persona principal.

Nota: La proyección es el proceso de añadir información de referencia cruzada a respuestas de servicios de entidades de negocio específicas. Puede considerar la proyección como un posible paso adicional que ocurre antes o después de una fase de servicio en particular.

2. **BeforeValidate.** Invoca una llamada externa y activa un evento admitido antes de validar los datos de entrada. Un caso de uso de ejemplo puede implicar que configure esta fase de servicio para validar la lógica antes de validar los datos de entrada cuando persiste una entidad de negocio. Por ejemplo, la validación de la dirección o el número de teléfono que se produce en un servicio externo de terceros antes de conservar un registro de persona.
3. **AfterValidate.** Invoca una llamada externa y activa un evento admitido inmediatamente después de validar los datos de entrada. Un caso de uso de ejemplo puede implicar que configure esta fase de servicio para validar la lógica después de validar los datos de entrada cuando persiste una entidad de negocio. Por ejemplo, la validación de la dirección o el número de teléfono que se produce en un servicio externo de terceros después de conservar un registro de persona.

Nota: Cuando se activa el servicio `AfterValidate`, los datos de entrada pueden ser diferentes porque la fase de servicio anterior modificó o procesó los datos.

4. **AfterEverything.** Invoca una llamada externa y activa un evento admitido después de realizar la lógica del servicio, como validación, localización y proyección. Un caso de uso de ejemplo puede implicar que configure esta fase de servicio para activar la notificación después de que el servicio realice toda la lógica. Por ejemplo, enviar una notificación por correo electrónico después de una fusión exitosa de dos o más registros de persona.

Nota: Cuando se activa la fase de servicio `AfterEverything`, una llamada externa tiene acceso a los datos de la respuesta del servicio y no a los datos de entrada.

Servicios que admiten las fases del servicio

Puede utilizar los siguientes servicios internos que admiten las fases del servicio:

WriteCO

Persiste y actualiza las entidades de negocio. El servicio WriteCO admite las siguientes fases de servicio:

- BeforeEverything
- BeforeValidate
- AfterValidate
- AfterEverything

Estas fases de servicio se activan antes, durante y después de que persista una entidad de negocio.

WriteView

Persiste y actualiza las vistas de entidad de negocio. El servicio WriteView admite las siguientes fases de servicio:

- BeforeEverything
- BeforeValidate
- AfterValidate
- AfterEverything

Estas fases de servicio se activan antes, durante y después de que persista una entidad de negocio.

El servicio WriteView activa todas las fases del servicio WriteCO compatibles en el siguiente orden:

1. WriteView.BeforeEverything
2. WriteView.BeforeValidate
3. WriteView.AfterValidate
4. WriteCO.BeforeEverything
5. WriteCO.BeforeValidate
6. WriteCO.AfterValidate
7. WriteCO.AfterEverything
8. WriteView.AfterEverything

ReadCO

Recupera entidades de negocio. El servicio ReadCO admite las siguientes fases de servicio:

- BeforeEverything
- AfterEverything

Estas fases de servicio se activan antes y después de la lectura de una entidad de negocio.

ReadView

Recupera vistas de entidad de negocio. El servicio ReadView admite las siguientes fases de servicio:

- BeforeEverything
- AfterEverything

Estas fases de servicio se activan antes y después de la lectura de una vista de entidad de negocio.

Nota: El servicio ReadView activa todas las fases del servicio ReadCO compatibles en el siguiente orden:

1. ReadView.BeforeEverything
2. ReadCO.BeforeEverything

3. `ReadCO.AfterEverything`
4. `ReadView.AfterEverything`

PreviewMergeCO

Recupera el resultado de la vista previa de la fusión de varias entidades de negocio. El servicio `PreviewMergeCO` admite las siguientes fases de servicio:

- `BeforeEverything`
- `AfterEverything`

Estas fases de servicio se activan antes y después de la generación de la vista previa de una entidad de servicio fusionada.

MergeCO

Fusiona múltiples entidades de negocio. El servicio `MergeCO` admite las siguientes fases de servicio:

- `BeforeEverything`
- `AfterEverything`

Estas fases de servicio se activan antes y después de la fusión de una entidad de negocio.

Parámetros del método de proceso

Cuando implementa la lógica empresarial para una llamada externa, puede utilizar los parámetros del método de proceso.

Utilice los siguientes parámetros en una interfaz Java lógica personalizada:

helperContext

Define el contexto de ejecución auxiliar que requiere la operación de objetos de servicio de datos (SDO).

inputSdo

Especifica el objeto de datos de servicio que representa la entidad de negocio, incluidos los registros principales y secundarios.

inParams

Especifica las propiedades de entrada de la solicitud anterior y depende directamente de la fase de servicio definida dentro de la llamada externa. El parámetro `inParams` también puede incluir los parámetros `outParams` de una llamada anterior.

Por ejemplo, `validateOnly` es una propiedad de entrada que indica que se debe realizar la validación en un registro principal, un registro secundario o un nivel de entidad de negocio.

La propiedad `validateOnly` puede tener los siguientes valores:

- `true`. Realiza la validación de un registro principal o secundario. Por ejemplo, puede especificar esta propiedad para validar 10 números de teléfono individualmente.
- `null` o `false`. Realiza validación a nivel de la entidad de negocio. Por ejemplo, puede especificar esta propiedad para realizar la validación de la entidad de negocio cuando un usuario guarda el registro.

Otro ejemplo es la propiedad `servicePhase` que especifica el nombre de la llamada externa.

outParams

Agrega o sobrescribe los parámetros inputSdo e inParams admitidos. Por ejemplo, una llamada externa que incluye el evento Merge_CO_Before_Everything podría actualizar el parámetro keysAndOverrides para añadir más claves de entidad de negocio o eliminar claves para una fusión.

En el valor de retorno, cuando hay un cambio de datos en inputSdo, la llamada externa devuelve uno de los siguientes valores:

- null. Cuando no hay cambios de datos en el parámetro inputSdo.
- El parámetro inputSdo modificado. Cuando hay un cambio de datos en el parámetro inputSdo y desea reflejar el cambio en la entidad de negocio.

El siguiente código de ejemplo muestra una interfaz Java lógica personalizada que incluye los parámetros del método de proceso:

```
public class ValidateOrgWriteBeforeEverything implements CustomLogic {
    @Override
    public DataObject process(HelperContext helperContext, DataObject inputSdo,
        Map<String, Object> inParams,
        Map<String, Object> outParams) throws StepException {
        // your implementation
        return null;
        //or
        return inputSdo;
    }
}
```

Prácticas recomendadas para llamadas externas

Considere las siguientes mejores prácticas al implementar llamadas externas:

- Implemente una llamada externa cuando no exista una función existente para limpiar, analizar o transformar datos.
- Tenga cuidado al implementar una llamada externa porque la llamada puede afectar el rendimiento de MDM Hub y la respuesta a la llamada de servicio de la entidad de negocio.
- Identifique cuándo desea utilizar la llamada externa. Por ejemplo, ejecute una llamada externa cuando un usuario navegue de un registro secundario a otro o cuando el usuario haga clic en **Guardar**.
- Defina una clase separada para cada llamada externa.
- Si incluye el parámetro del método de proceso inParams con la propiedad validateOnly en una llamada externa, asegúrese de establecer la propiedad validateOnly en uno de los siguientes valores:
 - false o null. Ejecuta la llamada externa después de que el usuario hace clic en **Guardar**.
 - true. Ejecuta la llamada externa después de que el usuario actualiza un registro principal o secundario y luego se desplaza a otro registro secundario.
- Si incluye propiedades personalizadas en llamadas externas posteriores hasta el evento AfterEverything, añada las propiedades personalizadas en el parámetro del método de proceso outParams.
- El evento WriteCO.AfterEverything final durante una operación de guardado devuelve los ID de fila de los registros principales y secundarios. Si necesita otros datos, llame a los servicios de entidad de negocio de lectura.
- No implemente llamadas externas para operaciones de lectura porque las operaciones de lectura ocurren durante las siguientes operaciones:
 - Buscar

- Leer antes de escribir
- Leer después de escribir
- Validar registros mediante funciones de limpieza.
- Si implementa una validación adicional, realice la validación adicional en el evento `WriteCO.AfterEverything` cuando la propiedad `validateOnly` se establezca en uno de los siguientes valores:
 - `false` o `null`. Para un objeto base.
 - `true`. Para un registro secundario.
- Realice una llamada de servicios de entidad de negocio desde una llamada externa similar a cualquier aplicación externa que utilice llamadas de servicios de entidad de negocio para leer o escribir datos.

Creación de llamadas externas

Un servicio de entidad de negocio cuenta con pasos de servicio. Una solicitud entrante pasa a través de cada paso de servicio. Puede crear llamadas externas a servicios externos para determinados pasos de la lógica de ejecución del servicio de entidad de negocio. Cree una llamada externa cuando no exista una función para limpiar, analizar o transformar datos.

Para crear una llamada externa, realice los siguientes pasos:

1. Comprenda e implemente el servicio web de ejemplo.
2. Inicie sesión en la herramienta de aprovisionamiento.
3. Cargue el archivo WSDL.
4. Registre el servicio de SOAP.
5. Configure la llamada externa.
6. Publique las configuraciones.

Después de crear una llamada externa, para asegurarse de que funciona como se esperaba, pruébela.

Para ver una llamada externa de ejemplo que incluya estos pasos, consulte [“Ejemplo de llamada externa” en la página 304](#).

Ejemplo de llamada externa

Para aprender a crear una llamada externa, revise el ejemplo de `BESEExternalCall` que está disponible en el Kit de recursos. Las siguientes secciones incluyen información sobre los pasos necesarios para crear y probar el ejemplo de `BESEExternalCall`.

El ejemplo de `BESEExternalCall` implementa la base más simple para una configuración exitosa de una llamada externa.

El ejemplo de `BESEExternalCall` tiene la siguiente estructura de código:

- Implementaciones de `CustomLogic` en los archivos `ValidatePerson.java` y `MergePerson.java`
- Implementación de `CustomLogicFactory` en el archivo `CustomLogicFactoryImpl.java`
- Implementación de `WebService` en el archivo `CustomLogicService.java`

Implementaciones de CustomLogic en los archivos ValidatePerson.java y MergePerson.java

Las implementaciones de CustomLogic para una llamada externa incluyen la lógica empresarial personalizada, como la validación y manipulación de datos adicionales.

Las implementaciones de CustomLogic deben incluir una interfaz `com.informatica.mdm.spi.externalcall.CustomLogic`, que está disponible en el archivo `mdm-spi.jar`. Esta interfaz expone el método de proceso donde puede definir la lógica empresarial personalizada. El siguiente código de clase `ValidatePerson.java` es un ejemplo:

```
public class ValidatePerson implements CustomLogic {
    private CompositeServiceClient besClient;
    private CallContext callContext;
    public ValidatePerson(CompositeServiceClient besClient, CallContext callContext) {
        this.besClient = besClient;
        this.callContext = callContext;
    }
    @Override
    public DataObject process(HelperContext helperContext, DataObject inputSdo,
        Map < String, Object > inParams,
        Map < String, Object > outParams) throws StepException {
        try {
            List < ValidationError > errorList = new ArrayList < > ();
            DataFactory dataFactory = helperContext.getDataFactory();

            // Ensure that the Person SDO has at least one Address item
            List<?> list = inputSdo.getList("Person/Addresses/item");
            if ((list == null) || list.isEmpty()) {
                String personId = inputSdo.getString("Person/rowidObject");

                if(personId != null) { // Verify if an address already exists
                    DataObject readEntity = dataFactory.create(ReadEntity.class);
                    DataObject personFilter =

readEntity.createDataObject("parameters").createDataObject("coFilter").createDataObject("
object");
                    personFilter.setString("name", "Person");
                    personFilter.createDataObject("key").setString("rowid",
personId);
                    DataObject addressFilter =
personFilter.createDataObject("object");
                    addressFilter.setString("name", "Addresses");

addressFilter.createDataObject("pager").setInt("recordsToReturn", 1);
                    DataObject readEntityReturn = besClient.process(callContext,
readEntity);
                    List existingList = readEntityReturn.getList("object/Person/
Addresses/item");
                    if (existingList == null || existingList.isEmpty()) {
                        errorList.add(createValidationError(dataFactory,
"CUSTOM-00001", "You must enter at least one Address.", "Person.Addresses"));
                    }
                }
            }
        }
    }
}
```

Implementación de CustomLogicFactory en el archivo CustomLogicFactoryImpl.java

Una implementación de CustomLogicFactory proporciona las instancias de CustomLogic para que se ejecuten en la fase esperada, se ejecuten para la entidad de negocio de destino o ambas.

Una implementación de CustomLogicFactory debe incluir una interfaz `com.informatica.mdm.spi.externalcall.CustomLogicFactory`, que está disponible en el archivo `mdm-spi.jar`. Esta interfaz expone el método de creación, que la solicitud de llamada externa invoca en las primeras etapas. La lógica esperada en este método filtra y dirige la solicitud de llamada externa a la implementación lógica personalizada esperada. El siguiente código `CustomLogicFactoryImpl.java` es un ejemplo:

```
public class CustomLogicFactoryImpl implements CustomLogicFactory {

    public static final String PERSON = "Person";
}
```

```

        private static final CustomLogic EMPTY_LOGIC = new EmptyLogic();

        private CompositeServiceClient besClient;

        public CustomLogicFactoryImpl(CompositeServiceClient besClient)
        { this.besClient = besClient;
        }

        @Override
        public CustomLogic create(ExternalCallRequest externalCallRequest,
        CallContext callContext) throws StepException {
            // Interest is in the Person business entity. The logic can handle a few
        service phases.

            Trigger trigger = externalCallRequest.getTrigger();
            String businessEntity = trigger.getBusinessEntity();
            ServicePhase phase = trigger.getServicePhase();

            switch (phase) {
                case WRITE_CO_BEFORE_VALIDATE:
                    if (PERSON.equals(businessEntity)) {
                        return new ValidatePerson(besClient, callContext);
                    }
                    break;
                case PREVIEW_MERGE_CO_BEFORE_EVERYTHING:
                case MERGE_CO_BEFORE_EVERYTHING:
                    if (PERSON.equals(businessEntity)) {
                        return new MergePerson();
                    }
                    break;
                default:
                    //
            }
            return EMPTY_LOGIC; // This one does nothing
        }
    }
}

```

Una implementación de CustomLogicFactory puede incluir los siguientes parámetros de método de creación:

externalCallRequest

Una instancia de la solicitud que representa la llamada externa como se define en el archivo bes-external-call.xsd.

callContext

Una instancia del contexto de servicio compuesto que las solicitudes de CompositeServiceClient usan en implementaciones lógicas personalizadas.

Implementación de WebService en el archivo CustomLogicService.java

Una implementación de WebService expone el servicio completo que invoca una llamada externa y es el punto de entrada del flujo de llamadas externas.

El siguiente código de ejemplo CustomLogicService.java implementa el servicio web que define el archivo bes-external-call.xsd en función de la API de WebService nativa de Java:

```

/**
 * Web service implementation.
 * It must accept a {urn:bes-external-call.informatica.mdm}ExternalCallRequest as an
 * operation input
 * and return a {urn:bes-external-call.informatica.mdm}ExternalCallResponse as output.
 */
@WebServiceProvider(
    targetNamespace = "http://cs.sample.mdm.informatica.com/",
    serviceName = "CustomLogicService",
    portName = "CustomLogicServicePort",
    wsdlLocation = "WEB-INF/wsdl/custom-logic-service.wsdl"
)
@ServiceMode(Mode.PAYLOAD)
public class CustomLogicService implements Provider<Source> {

```

```

@Override
public Source invoke(Source request) {

    CompositeServiceClient compositeServiceClient =
createCompositeServiceClient();
    CustomLogicFactory customLogicFactory = new
CustomLogicFactoryImpl(compositeServiceClient);
    String appName = "<trusted_user>"; // replace with proper application user name

    // Create a processor instance and let the instance perform the job.
    // Provide a custom logic factory implementation.
    ExternalCallProcessor externalCallProcessor =
        new ExternalCallProcessor(compositeServiceClient, appName,
customLogicFactory);

    return externalCallProcessor.invoke(request);
}

```

El método de invocación de la instancia ExternalCallProcessor maneja la solicitud entrante.

Nota: La instancia de ExternalCallProcessor se basa en un usuario de la aplicación de confianza. Para obtener más información sobre los usuarios de aplicaciones de confianza, consulte *Guía de seguridad de Multidomain MDM*.

Ejemplo de BESExternalCall

La llamada externa BESExternalCall al servicio web puede realizar las siguientes acciones:

- Al crear o actualizar una persona, la llamada valida si la persona tiene al menos una dirección. Si la validación falla, la llamada devuelve un mensaje de error personalizado.
- Al generar una vista previa de los registros fusionados, la llamada no fusiona los números de teléfono.
- Al fusionar registros, la llamada no fusiona números de teléfono.

La siguiente tabla describe los archivos y las ubicaciones de los archivos que admiten el ejemplo de BESExternalCall:

Nombre de archivo	Descripción	Ubicación
bes-external-call.xsd	Define los tipos de solicitud y respuesta para todas las llamadas externas. Una operación de servicio web externo debe usar los tipos de solicitud y respuesta para los elementos de entrada y salida.	<MDM installation directory>/hub/server/lib/mdm-spi.jar
custom-logic-service.wsdl	Archivo WSDL para el servicio web de ejemplo. Define los servicios, operaciones, métodos y tipos de datos externos que intercambian los métodos de servicio.	<Directorio de instalación de MDM>/hub/resourcekit/samples/BESExternalCall/source/resources/webapp/WEB-INF/wsdl/
ValidatePerson.java y MergePerson.java	Ejemplos de operaciones personalizadas de validación y fusión que se ejecutan con el servicio web de ejemplo. Nota: Este ejemplo usa archivos .java, pero su servicio web externo puede requerir un estándar de codificación diferente.	<Directorio de instalación de MDM>/hub/resourcekit/samples/BESExternalCall/source/java/

Nombre de archivo	Descripción	Ubicación
build.xml	Construye el archivo <code>bes-external-call.ear</code> .	<Directorio de instalación de MDM>/hub/resourcekit/samples/BESEExternalCall/
bes-external-call.ear	El archivo <code>.ear</code> que implementa en el servidor de aplicaciones para ejecutar este ejemplo.	<Directorio de instalación de MDM>/hub/resourcekit/samples/BESEExternalCall/build/

Paso 1. Comprender e implementar el servicio web de ejemplo

El ejemplo de BESEExternalCall incluye un archivo WSDL y los archivos Java con lógica personalizada que utilizan las llamadas externas. Revise los archivos del ejemplo. Para usar el ejemplo, cree e implemente el archivo `.ear` al servidor de aplicaciones que ejecuta MDM Hub.

Cuando diseñe llamadas externas, primero desarrolle la lógica personalizada en el marco del servicio externo. Luego, cree el archivo WSDL que define los servicios, operaciones, métodos y tipos de datos externos que intercambian los métodos de servicio. Asegúrese de que el archivo WSDL utilice los tipos de solicitud y respuesta que se definen en el archivo `bes-external-call.xsd`.

1. Para comprender los requisitos para los tipos de solicitud y respuesta, revise el archivo `bes-external-call.xsd`.
2. Para comprender la implementación de los tipos de solicitud y respuesta para el servicio web de ejemplo, revise el archivo `custom-logic-service.wsdl`.
3. Para comprender la lógica personalizada que utiliza el servicio web de ejemplo, revise los archivos Java en el kit de recursos.
4. Para construir el archivo de ejemplo `.ear`, use el archivo `build.xml` con una herramienta de compilación ANT.
Para obtener más información sobre la creación del archivo EAR, consulte *Guía del kit de recursos de Multidomain MDM*.
5. Implemente el archivo `bes-external-call.ear` en el servidor de aplicaciones que ejecuta MDM Hub.
Para obtener más información sobre la implementación del archivo EAR, consulte *Guía del kit de recursos de Multidomain MDM*.

Paso 2. Iniciar sesión en la herramienta de aprovisionamiento

Configure el servicio web y la llamada externa en la herramienta de aprovisionamiento.

Para iniciar sesión, necesita la URL de la herramienta de aprovisionamiento y sus credenciales de usuario. La URL incluye el nombre de host y el número de puerto del servidor de MDM Hub. Si no dispone de esta información, póngase en contacto con el administrador de MDM Hub.

MDM Hub debe estar en ejecución en el mismo servidor de aplicaciones que la herramienta de aprovisionamiento. Para obtener más información acerca de la herramienta de aprovisionamiento, consulte la *Guía de la herramienta de aprovisionamiento de Multidomain MDM*.

1. Abra un navegador compatible.
2. Introduzca la URL de la herramienta de aprovisionamiento.

La URL tiene los siguientes formatos:

- **Conexiones seguras.** `https://<nombre de host del servidor de MDM Hub>:<número de puerto del servidor de MDM>/provisioning/`
- **Conexiones no seguras.** `http://<nombre de host del servidor de MDM Hub>:<número de puerto del servidor de MDM Hub>/provisioning/`

Se abre la página **Iniciar sesión**.

3. Introduzca el nombre de usuario y la contraseña.
4. Haga clic en **Iniciar sesión**.
5. Cuando se le solicite una ORS, seleccione la muestra de MDM ORS.
La herramienta de aprovisionamiento se abre y muestra la página **Inicio**.

Paso 3. Cargar el archivo WSDL de muestra

Cargue el archivo WSDL para el servicio web de ejemplo BESEExternalCall.

Para obtener más información sobre la carga del archivo WSDL, consulte *Guía de la herramienta de aprovisionamiento de Multidomain MDM*.

1. En la herramienta de aprovisionamiento, haga clic en **Entidad de negocio > Extensiones**.
La página **Extensiones** se abre.
2. Seleccione **WSDL** y, a continuación, haga clic en **Crear**.
3. En el panel Propiedades, especifique las siguientes propiedades:

Propiedad	Descripción	Ejemplo Valor
Nombre	Un nombre descriptivo para el servicio web de ejemplo.	besexternal
URL	URL del archivo WSDL.	<code>https://<host>:<port>/bes-external-call/customLogicService?wsdl</code>

4. Haga clic en **Aplicar**.
El panel **Vista de árbol** muestra los detalles.

Paso 4. Registrar el servicio de SOAP

Registre el servicio y las operaciones SOAP que describe WSDL.

Para obtener más información sobre cómo registrar el servicio SOAP, consulte *Guía de la herramienta de aprovisionamiento de Multidomain MDM*.

1. En la página **Extensiones**, seleccione **Servicios de SOAP** y, a continuación, haga clic en **Crear**.
Las propiedades aparecen en el panel de propiedades.

- En el panel de propiedades, especifique las siguientes propiedades:

Propiedad	Descripción	Ejemplo Valor
Nombre	Nombre del servicio de SOAP.	userexit
WSDL	Nombre que especificó cuando cargó el archivo WSDL de ejemplo. Los detalles del servicio web aparecen en los campos Espacio de nombres, Nombre del servicio, Nombre del puerto y Dirección de extremo.	besexternal
Encabezado SOAP	Cualquier información adicional que necesite para ejecutar el servicio web, como el nombre de usuario y la contraseña.	En el ejemplo de BESEExternalCall, no hay encabezados SOAP.

- Haga clic en **Aplicar**.
El panel **Vista de árbol** muestra los detalles.

Paso 5. Configurar la llamada externa

La llamada externa utiliza el servicio y las operaciones SOAP configuradas, y la llamada se activa para tres fases de servicio.

Para obtener más información sobre la configuración de las llamadas externas, consulte *Guía de la herramienta de aprovisionamiento de Multidomain MDM*.

- En la página **Extensiones**, seleccione **Llamadas externas** y, a continuación, haga clic en **Crear**.
- En el panel **Propiedades**, especifique las siguientes propiedades:

Propiedad	Descripción	Ejemplo Valor
Nombre	Nombre de la llamada externa.	validateperson
Servicio de SOAP	Servicio de SOAP que configuró.	userexit
Operación SOAP	Operación SOAP para realizar.	validate

- Haga clic en **Aplicar**.
El panel **Vista de árbol** se completa según el contenido del servicio y la operación SOAP.
- En el panel **Vista de árbol**, seleccione **Entidades de negocio**, haga clic en **Crear** y luego seleccione **Persona**.
- Seleccione **Fase de servicio**, haga clic en **Crear** y luego seleccione las siguientes fases de servicio para el código de muestra:
 - WriteCO.BeforeValidate
 - PreviewMergeCO.BeforeEverything
 - MergeCO.BeforeEverything

Nota: Según su lógica personalizada, puede seleccionar las fases del servicio.

6. Haga clic en **Aplicar**.

Paso 6. Publicar las configuraciones

Las configuraciones para el archivo WSDL, el servicio de SOAP y las llamadas externas se almacenan en una caché. Publique las configuraciones en MDM Hub.

En la herramienta de aprovisionamiento, haga clic en **Publicar**.

El proceso de publicación guarda las configuraciones en MDM Hub.

Probar la validación de direcciones

Utilice las API REST para verificar que el paso de validación funcione. Cree un registro de persona sin una dirección. Si la configuración de la llamada externa es correcta, aparece un error de validación que indica que un registro de persona debe tener al menos una dirección.

Para usar las URL de REST, necesita el ID de la base de datos para el ORS de muestra de MDM y un nombre de usuario de MDM.

1. Busque el ID de la base de datos para el ORS de muestra de MDM:
 - a. Inicie sesión en la consola del concentrador.
 - b. En el entorno de trabajo **Configuración**, haga clic en **Bases de datos**.
 - c. Seleccione la base de datos de muestra de MDM.
 - d. En el panel **Propiedades de la base de datos**, copie el valor de **ID de base de datos**. Utilice este valor en la URL de REST.
2. Use la API Create para crear la siguiente entidad de negocio de persona sin ninguna dirección.

```
POST https://<host>:<port>/cmx/cs/<database id>/Person?systemName=<user name>
{
    firstName: "John"
}
```

La prueba de validación falla y muestra el error. No se crea el registro de persona.

3. Cree una entidad de negocio de persona con una dirección.

```
POST https://<host>:<port>/cmx/cs/<database id>/Person?systemName=<user name>
{
    firstName: "John",
    Addresses: {
        item: [
            {
                cityName: "Toronto"
            }
        ]
    }
}
```

La prueba de validación se aprueba sin errores. Se crea el registro de persona.

Probar la lógica de fusión

Para probar la lógica de fusión que evita que los números de teléfono se fusionen, use la API MergePreview.

1. Para crear entidades de negocio de dos personas, utilice la API Create.

```
POST https://<host>:<port>/cmx/cs/<ors>/Person?systemName=Admin
{
    firstName: "John",
    Addresses: {
```

```

        item: [
            {
                cityName: "Toronto"
            }
        ],
        TelephoneNumbers: {
            item:[
                {
                    phoneNum: "111-11-11"
                }
            ]
        }
    }
}
POST https://<host>:<port>/cmx/cs/<ors>/Person?systemName=Admin
{
    firstName: "John",
    Addresses: {
        item: [
            {
                cityName: "Ottawa"
            }
        ]
    },
    TelephoneNumbers: {
        item:[
            {
                phoneNum: "222-22-22"
            }
        ]
    }
}

```

La respuesta incluye los siguientes ID de fila de muestra:

- Persona: 161923, 161924
- Direcciones: 2123, 2124
- Números de teléfono: 101723, 101724

2. Para obtener una vista previa de las entidades de negocio de persona fusionadas, ejecute la API PreviewMerge.

```

POST https://<host>:<port>/cmx/cs/<ors>/Person/161923?action=previewMerge&depth=2
{
    keys: [
        {
            rowid: "161924"
        }
    ]
}

```

La respuesta incluye dos direcciones y dos números de teléfono.

```

{
  "Person": {
    "rowidObject": "161923",
    "creator": "admin",
    "createDate": "2016-10-20T09:50:35.878-04:00",
    "updatedBy": "admin",
    "lastUpdateDate": "2016-10-20T09:50:35.879-04:00",
    "consolidationInd": 4,
    "lastRowidSystem": "SYS0",
    "hubStateInd": 1,
    "label": "Person: , Bill",
    "partyType": "Person",
    "displayName": "Bill",
    "firstName": "Bill",
    "TelephoneNumbers": {
      "link": [],

```



```

"firstRecord": 1,
"recordCount": 2,
"pageSize": 2,
"item": [
  {
    "rowidObject": "101723",
    "creator": "admin",
    "createDate": "2016-10-20T09:50:35.904-04:00",
    "updatedBy": "admin",
    "lastUpdateDate": "2016-10-20T09:50:35.905-04:00",
    "consolidationInd": 4,
    "lastRowidSystem": "SYS0",
    "hubStateInd": 1,
    "label": "PhoneNumbers",
    "phoneNum": "111-1111",
    "phoneCountryCd": "1"
  },
  {
    "rowidObject": "101724",
    "creator": "admin",
    "createDate": "2016-10-20T09:50:54.768-04:00",
    "updatedBy": "admin",
    "lastUpdateDate": "2016-10-20T09:50:54.769-04:00",
    "consolidationInd": 4,
    "lastRowidSystem": "SYS0",
    "hubStateInd": 1,
    "label": "PhoneNumbers",
    "phoneNum": "222-2222",
    "phoneCountryCd": "1"
  }
],
"Addresses": {
  "link": [],
  "firstRecord": 1,
  "recordCount": 2,
  "pageSize": 2,
  "item": [
    {
      "rowidObject": "2123",
      "creator": "admin",
      "createDate": "2016-10-20T09:50:37.956-04:00",
      "updatedBy": "admin",
      "lastUpdateDate": "2016-10-20T09:50:37.956-04:00",
      "consolidationInd": 4,
      "lastRowidSystem": "SYS0",
      "hubStateInd": 1,
      "label": "Addresses",
      "Address": {
        "rowidObject": "2121",
        "creator": "admin",
        "createDate": "2016-10-20T09:50:36.922-04:00",
        "updatedBy": "admin",
        "lastUpdateDate": "2016-10-20T09:50:37.923-04:00",
        "consolidationInd": 4,
        "lastRowidSystem": "SYS0",
        "hubStateInd": 1,
        "label": "Address",
        "cityName": "Toronto"
      }
    },
    {
      "rowidObject": "2124",
      "creator": "admin",
      "createDate": "2016-10-20T09:50:54.790-04:00",
      "updatedBy": "admin",
      "lastUpdateDate": "2016-10-20T09:50:54.790-04:00",
      "consolidationInd": 4,
      "lastRowidSystem": "SYS0"
    }
  ]
}

```



```

"hubStateInd": 1,
"label": "Person: , Bill",
"partyType": "Person",
"displayname": "Bill",
"firstName": "Bill",
"TelephoneNumbers": {
  "link": [],
  "firstRecord": 1,
  "recordCount": 2,
  "pageSize": 2,
  "item": [
    {
      "rowidObject": "101723",
      "creator": "admin",
      "createDate": "2016-10-20T09:50:35.904-04:00",
      "updatedBy": "admin",
      "lastUpdateDate": "2016-10-20T09:50:35.905-04:00",
      "consolidationInd": 4,
      "lastRowidSystem": "SYS0",
      "hubStateInd": 1,
      "label": "PhoneNumbers",
      "phoneNum": "111-1111",
      "phoneCountryCd": "1"
    },
    {
      "rowidObject": "101724",
      "creator": "admin",
      "createDate": "2016-10-20T09:50:54.768-04:00",
      "updatedBy": "admin",
      "lastUpdateDate": "2016-10-20T09:50:54.769-04:00",
      "consolidationInd": 4,
      "lastRowidSystem": "SYS0",
      "hubStateInd": 1,
      "label": "PhoneNumbers",
      "phoneNum": "222-2222",
      "phoneCountryCd": "1"
    }
  ]
},
"Addresses": {
  "link": [],
  "firstRecord": 1,
  "recordCount": 1,
  "pageSize": 1,
  "item": [
    {
      "rowidObject": "2123",
      "creator": "admin",
      "createDate": "2016-10-20T09:50:37.956-04:00",
      "updatedBy": "admin",
      "lastUpdateDate": "2016-10-20T09:50:37.956-04:00",
      "consolidationInd": 4,
      "lastRowidSystem": "SYS0",
      "hubStateInd": 1,
      "label": "Addresses"
    }
  ]
},
"PersonDetails": {
  "link": [],
  "recordCount": 0,
  "pageSize": 0,
  "item": []
}
}

```

Obtención de ID de fila de origen con la llamada externa MergeCO.BeforeEverything

Durante una acción de combinación, los ID de fila de origen ayudan a rastrear los objetos base de origen. Puede obtener los ID de fila de origen del parámetro del método de proceso inputSDO con la llamada externa MergeCO.BeforeEverything.

Para obtener los ID de las filas de origen, realice los siguientes pasos:

1. Implemente la llamada externa MergeCO.BeforeEverything.

El siguiente código de ejemplo del archivo MergePerson.java obtiene los ID de fila de origen con la llamada externa MergeCO.BeforeEverything:

```
package com.informatica.mdm.sample.cs;

import java.util.List;
import java.util.Map;

import com.informatica.mdm.sdo.co.base.BaseObject;
import com.informatica.mdm.sdo.co.base.CompositeObjectBase;
import com.informatica.mdm.sdo.co.base.KeysAndOverrides;
import com.informatica.mdm.sdo.co.base.MergedRecord;
import com.informatica.mdm.sdo.co.base.MergedRecordPager;
import com.informatica.mdm.sdo.co.base.Pager;
import com.informatica.mdm.sdo.cs.base.BusinessEntityKey;
import com.informatica.mdm.sdo.cs.base.Key;
import com.informatica.mdm.spi.cs.StepException;
import com.informatica.mdm.spi.externalcall.CustomLogic;
import commonj.sdo.ChangeSummary;
import commonj.sdo.DataObject;
import commonj.sdo.Property;
import commonj.sdo.helper.HelperContext;
import org.eclipse.persistence.sdo.SDOChangeSummary;

/**
 * Call this piece of code before the Person business entity record merge or the
 * previewMerge API call.
 */
public class MergePerson implements CustomLogic {

    public static final String KEYS_AND_OVERRIDES = "keysAndOverrides";
    public static final String OVERRIDES = "overrides";
    public static final String MERGE = "MERGE";

    @Override
    public DataObject process(HelperContext helperContext, DataObject inputSdo,
        Map<String, Object> inParams,
        Map<String, Object> outParams) throws StepException {

        DataObject keysAndOverrides = (DataObject) inParams.get(KEYS_AND_OVERRIDES);

        if (keysAndOverrides != null) {

            explainMerge(inParams);

            DataObject overrides = keysAndOverrides.getDataObject(OVERRIDES);
            if (overrides != null) {
                // if "overrides" are provided, then user is not simply merging
                several records, but
                // tries to merge some child records or provides some winner override
                List<DataObject> list = overrides.getList("Person/TelephoneNumbers/
item");
                if ((list != null) && !list.isEmpty()) {
                    SDOChangeSummary changeSummary = (SDOChangeSummary)
overrides.getChangeSummary();
                }
            }
        }
    }
}
```

```

        changeSummary.resumeLogging();
        // remove "MERGE" element from all "Telephone" items, this will
prevent mering of "Telephone" child records
        for (DataObject dataObject : list) {
            if (dataObject.isSet(MERGE)) {
                dataObject.unset(MERGE);
            }
        }
        // send updated "overrides" back to Hub
        outParams.put(KEYS_AND_OVERRIDES, keysAndOverrides);
    }
}
return inputSdo;
}

private void explainMerge(Map<String, Object> inParams) {
    BusinessEntityKey businessEntityKey = (BusinessEntityKey)
inParams.get("businessEntityKey");
    System.out.println(String.format("%s:%s", businessEntityKey.getName(),
businessEntityKey.getKey().getRowid()));
    KeysAndOverrides keysAndOverrides = (KeysAndOverrides)
inParams.get("keysAndOverrides");
    List<Key> keys = keysAndOverrides.getKeys();
    for (Key key : keys) {
        System.out.println(String.format(" | <- %s", key.getRowid()));
    }
    CompositeObjectBase overrides = keysAndOverrides.getOverrides();
    if(overrides != null) {
        DataObject co = (DataObject) overrides;
        ChangeSummary changeSummary = co.getChangeSummary();
        DataObject root = co.getDataObject(co.getType().getName());
        dumpOne("", co.getType().getName(), root, changeSummary);
    }
}

private void dumpOne(String prefix, String name, DataObject dataObject,
ChangeSummary changeSummary) {
    String rowidObject = dataObject.getString("rowidObject");
    if(rowidObject != null) {
        System.out.println(String.format("%s%s:%s", prefix, name, rowidObject));
    }
    List<MergedRecord> mergeItems = dataObject.getList("MERGE/item");
    if(mergeItems != null) {
        for (MergedRecord mergeItem : mergeItems) {
            System.out.println(String.format("%s | <- %s", prefix,
mergeItem.getKey().getRowid()));
        }
    }
    List<Property> properties = dataObject.getInstanceProperties();
    for (Property property : properties) {
        Object o = dataObject.get(property);
        if((o instanceof Pager) && !(o instanceof MergedRecordPager)) {
            DataObject pager = (DataObject) o;
            System.out.println(prefix + property.getName());
            List<DataObject> items = pager.getList("item");
            for (int i = 0; i < items.size(); i++) {
                DataObject item = items.get(i);
                dumpOne(prefix + "  ", "[" + i + "]", item, changeSummary);
            }
        }
        if(o instanceof BaseObject) {
            dumpOne(prefix + "  ", property.getName(), (DataObject) o,
changeSummary);
        }
    }
    if(changeSummary.isModified(dataObject)) {
        List<ChangeSummary.Setting> oldValues =
changeSummary.getOldValues(dataObject);
    }
}

```


El siguiente resultado de ejemplo muestra los ID de fila de origen:

```
2020-03-20 11:59:49,631 INFO [stdout] (default task-55) Person:401924
2020-03-20 11:59:49,631 INFO [stdout] (default task-55) | <- 401925
2020-03-20 11:59:49,632 INFO [stdout] (default task-55) | <- 401926
2020-03-20 11:59:49,632 INFO [stdout] (default task-55) TelephoneNumbers
2020-03-20 11:59:49,632 INFO [stdout] (default task-55) [0]:1724
2020-03-20 11:59:49,632 INFO [stdout] (default task-55) | <- 1725
2020-03-20 11:59:49,633 INFO [stdout] (default task-55) | <- 1726
2020-03-20 11:59:49,633 INFO [stdout] (default task-55) manual override:
phoneNum = 777-77-77
2020-03-20 11:59:49,633 INFO [stdout] (default task-55) Addresses
2020-03-20 11:59:49,633 INFO [stdout] (default task-55) [0]:2124
2020-03-20 11:59:49,633 INFO [stdout] (default task-55) | <- 2125
2020-03-20 11:59:49,634 INFO [stdout] (default task-55) | <- 2126
```

Ejemplo: Validación y lógica personalizadas para servicios de entidad de negocio

Puede probar la validación y la lógica personalizadas cuando añada y fusione los registros de persona. La validación personalizada comprueba si el registro de persona tiene una dirección. La lógica personalizada no permite fusionar dos números de teléfono. Use las API de REST para crear y fusionar registros de persona.

1. Para comprobar la validación al crear un registro de persona, realice los siguientes pasos:
 - a. Use la API de creación para crear un registro de persona sin ninguna dirección. Se genera un error de validación.
 - b. Use la API de creación para crear un registro de persona con una dirección. La operación se realiza correctamente.
2. Para comprobar la lógica personalizada al fusionar registros, realice los siguientes pasos:
 - a. Use la API de creación para crear dos registros de persona con direcciones y números de teléfono.
 - b. Use la API de vista previa de fusión para fusionar los dos registros de persona. Añada `overrides` a la solicitud de vista previa de fusión para fusionar las direcciones y los teléfonos. En la respuesta se muestran una sola dirección y dos números de teléfono. La lógica personalizada impide la fusión de los números de teléfono.

Requisitos previos

Para comprobar la lógica y la validación personalizadas, debe cargar el archivo WSDL en la herramienta de aprovisionamiento. Debe registrar el servicio SOAP y la operación. Enlace el servicio con las entidades de negocio y los eventos para los que desea usar la lógica y la validación personalizadas. Puede probar la lógica y la validación para las entidades de negocio y los eventos especificados.

Paso 1. Probar la validación personalizada

Use la API de creación para crear el siguiente registro de persona sin ninguna dirección:

```
POST http://localhost:8080/cmxcsllocalhost-orcl-mdm_Sample/Person?systemName=Admin
{
    firstName: "John"
}
```

Se genera un error de validación.

Use la API de creación para crear el siguiente registro de persona con una dirección:

```
POST http://localhost:8080/cmxcsllocalhost-orcl-mdm_Sample/Person?systemName=Admin
{
  firstName: "John",
  Addresses: {
    item: [
      {
        cityName: "Toronto"
      }
    ]
  }
}
```

La solicitud crea un registro de persona.

Paso 2. Probar la lógica personalizada

Realice los siguientes pasos para probar la lógica personalizada:

1. Use la API de creación para crear dos registros de persona con direcciones y teléfonos:

```
POST http://localhost:8080/cmxcsllocalhost-orcl-mdm_sample/Person?systemName=Admin
{
  firstName: "John",
  Addresses: {
    item: [
      {
        cityName: "Toronto"
      }
    ]
  },
  TelephoneNumbers: {
    item: [
      {
        phoneNum: "111-11-11"
      }
    ]
  }
}

POST http://localhost:8080/cmxcsllocalhost-orcl-mdm_sample/Person?systemName=Admin
{
  firstName: "John",
  Addresses: {
    item: [
      {
        cityName: "Ottawa"
      }
    ]
  },
  TelephoneNumbers: {
    item: [
      {
        phoneNum: "222-22-22"
      }
    ]
  }
}
```

La respuesta contiene los siguientes valores de rowId:

- Persona: 161923, 161924
- Direcciones: 2123, 2124
- Números de teléfono: 101723, 101724

2. Ejecute la API de vista previa de fusión para fusionar ambos registros de persona:

```
POST http://localhost:8080/cmx/cs/localhost-orcl-mdm_sample/Person/161923?
action=previewMerge&depth=2
{
  keys: [
    {
      rowid: "161924"
    }
  ]
}
```

La respuesta es un registro de persona con dos direcciones y dos números de teléfono.

```
{
  "Person": {
    "rowidObject": "161923",
    "creator": "admin",
    "createDate": "2016-10-20T09:50:35.878-04:00",
    "updatedBy": "admin",
    "lastUpdateDate": "2016-10-20T09:50:35.879-04:00",
    "consolidationInd": 4,
    "lastRowidSystem": "SYS0",
    "hubStateInd": 1,
    "label": "Person, Bill",
    "partyType": "Person",
    "displayName": "Bill",
    "firstName": "Bill",
    "TelephoneNumbers": {
      "link": [],
      "firstRecord": 1,
      "recordCount": 2,
      "pageSize": 2,
      "item": [
        {
          "rowidObject": "101723",
          "creator": "admin",
          "createDate": "2016-10-20T09:50:35.904-04:00",
          "updatedBy": "admin",
          "lastUpdateDate": "2016-10-20T09:50:35.905-04:00",
          "consolidationInd": 4,
          "lastRowidSystem": "SYS0",
          "hubStateInd": 1,
          "label": "PhoneNumbers",
          "phoneNum": "111-1111",
          "phoneCountryCd": "1"
        },
        {
          "rowidObject": "101724",
          "creator": "admin",
          "createDate": "2016-10-20T09:50:54.768-04:00",
          "updatedBy": "admin",
          "lastUpdateDate": "2016-10-20T09:50:54.769-04:00",
          "consolidationInd": 4,
          "lastRowidSystem": "SYS0",
          "hubStateInd": 1,
          "label": "PhoneNumbers",
          "phoneNum": "222-2222",
          "phoneCountryCd": "1"
        }
      ]
    }
  },
  "Addresses": {
    "link": [],
    "firstRecord": 1,
    "recordCount": 2,
    "pageSize": 2,
    "item": [
      {
        "rowidObject": "2123",
        "creator": "admin",

```



```

    },
    TelephoneNumbers: {
      item: [
        {
          rowidObject: "101723",
          MERGE: {
            item: [{key: {rowid: "101724"}}],
            $original: {
              item: [null]
            }
          }
        }
      ]
    }
  }
}

```

En la respuesta se muestran una sola dirección y dos números de teléfono.

```

{
  "Person": {
    "rowidObject": "161923",
    "creator": "admin",
    "createDate": "2016-10-20T09:50:35.878-04:00",
    "updatedBy": "admin",
    "lastUpdateDate": "2016-10-20T09:50:35.879-04:00",
    "consolidationInd": 4,
    "lastRowidSystem": "SYS0",
    "hubStateInd": 1,
    "label": "Person: , Bill",
    "partyType": "Person",
    "displayName": "Bill",
    "firstName": "Bill",
    "TelephoneNumbers": {
      "link": [],
      "firstRecord": 1,
      "recordCount": 2,
      "pageSize": 2,
      "item": [
        {
          "rowidObject": "101723",
          "creator": "admin",
          "createDate": "2016-10-20T09:50:35.904-04:00",
          "updatedBy": "admin",
          "lastUpdateDate": "2016-10-20T09:50:35.905-04:00",
          "consolidationInd": 4,
          "lastRowidSystem": "SYS0",
          "hubStateInd": 1,
          "label": "PhoneNumbers",
          "phoneNum": "111-1111",
          "phoneCountryCd": "1"
        },
        {
          "rowidObject": "101724",
          "creator": "admin",
          "createDate": "2016-10-20T09:50:54.768-04:00",
          "updatedBy": "admin",
          "lastUpdateDate": "2016-10-20T09:50:54.769-04:00",
          "consolidationInd": 4,
          "lastRowidSystem": "SYS0",
          "hubStateInd": 1,
          "label": "PhoneNumbers",
          "phoneNum": "222-2222",
          "phoneCountryCd": "1"
        }
      ]
    }
  },
  "Addresses": {
    "link": [],
    "firstRecord": 1,
  }
}

```

```

"recordCount": 1,
"pageSize": 1,
"item": [
  {
    "rowidObject": "2123",
    "creator": "admin",
    "createDate": "2016-10-20T09:50:37.956-04:00",
    "updatedBy": "admin",
    "lastUpdateDate": "2016-10-20T09:50:37.956-04:00",
    "consolidationInd": 4,
    "lastRowidSystem": "SYS0",
    "hubStateInd": 1,
    "label": "Addresses"
  }
],
"PersonDetails": {
  "link": [],
  "recordCount": 0,
  "pageSize": 0,
  "item": []
}
}

```

Ejemplo: realizar una llamada al marco de integración de servicios desde una llamada externa

Por razones de seguridad, MDM no transfiere la información de la carga útil y la contraseña a una llamada externa. El ejemplo de BESEExternalCallWithSIF muestra cómo llamar a un marco de integración de servicios (SIF) desde una llamada externa.

La llamada externa en este ejemplo realiza la siguiente validación personalizada:

1. Comprueba si un número de teléfono en un registro de persona tiene caracteres especiales en el número de extensión del teléfono.
2. Si el número de teléfono incluye caracteres especiales, lo corrige.
3. Comprueba si solo un número de teléfono tiene el tipo PRI.

Una vez que la llamada externa completa la validación de que solo existe un número de teléfono con el tipo PRI, la llamada externa realiza una actualización utilizando la llamada SIF Put.

Requisitos previos

Configure un usuario de la aplicación para llamadas del servicio externo a MDM Hub. Este ejemplo utiliza e360 como usuario de la aplicación. Reemplace este usuario de la aplicación con el usuario que configuró. Para obtener más información, consulte la *Guía de seguridad de Multidomain MDM*.

Hacer una llamada SIF desde una llamada externa

1. Cree e implemente el archivo bes-external-call.ear.
2. En la herramienta de aprovisionamiento, registre el servicio web como un servicio de SOAP.

La siguiente tabla describe las propiedades avanzadas que puede configurar para el servicio de SOAP:

Propiedad	Valor de ejemplo
Nombre	besexternal
URL	https://<host>:<puerto>/bes-external-call/CustomLogicService?wsdl

3. En la herramienta de aprovisionamiento, registre la llamada externa.
La siguiente tabla describe las propiedades que puede configurar para la llamada externa:

Propiedad	Valor de ejemplo
Operación SOAP	validate
Entidad de negocio	Persona
Fase de servicio	WriteCO.BeforeValidate

- Operación: validar
 - Entidad de negocio: persona
 - Fases del servicio: WriteCO.BeforeValidate
4. Cree un registro de persona con un número de teléfono que tenga caracteres especiales en el número de extensión.

La siguiente solicitud POST de ejemplo incluye los caracteres especiales @ en la extensión del número de teléfono:

```
POST /cmx/cs/<ors>/Person?systemName=Admin
{
  "firstName": "John",
  "TelephoneNumbers": {
    "item": [
      {
        "phoneExtNum": "e@1",
        "phoneNum": "1234567",
        "phoneTypeCd": {
          "phoneType": "PRI"
        }
      }
    ]
  }
}
```

La siguiente respuesta muestra el error de validación:

```
<?xml version="1.0" encoding="UTF-8"?>
<ns1:CsFault xmlns:ns1="urn:cs-base.informatica.mdm" xmlns:xsi="http://
www.w3.org/2001/XMLSchema-instance">
  <ns1:errorMessage>SIP-50022: Validation failed.</ns1:errorMessage>
  <ns1:errorCode>SIP-50022</ns1:errorCode>
  <ns1:details xsi:type="ns1:ValidationErrors">
    <ns1:error>
      <ns1:code>CUSTOM-00001</ns1:code>
      <ns1:message>The Phone Ext Number has special symbols.</ns1:message>
      <ns1:field>Person.TelephoneNumbers.phoneExtNum</ns1:field>
    </ns1:error>
  </ns1:details>
</ns1:CsFault>
```

El siguiente ejemplo es una solicitud POST válida que no genera un error de validación:

```
POST /cmx/cs/<ors>/Person?systemName=Admin
{
  "firstName": "John",
  "TelephoneNumbers": {
    "item": [
      {
        "phoneExtNum": "e1",
        "phoneNum": "1234567",
        "phoneTypeCd": {
          "phoneType": "PRI"
        }
      }
    ]
  }
}
```

5. Cree un registro de persona con un número de teléfono que incluya una letra, como o. Cuando se ejecuta la solicitud POST, el número 0 reemplaza a la letra o.

La siguiente solicitud POST de ejemplo incluye un número de teléfono con la letra o:

```
POST /cmx/cs/<ors>/Person?systemName=Admin
{
  "firstName": "John",
  "TelephoneNumbers": {
    "item": [
      {
        "phoneExtNum": "e1",
        "phoneNum": "123o4567",
        "phoneTypeCd": {
          "phoneType": "PRI"
        }
      }
    ]
  }
}
```

6. Considere los siguientes escenarios que demuestran cómo se establece el valor del tipo de teléfono en función de si una solicitud POST incluye uno o varios registros telefónicos:

- Escenario 1. Este escenario incluye dos solicitudes POST que crean un registro telefónico cada una. La siguiente solicitud POST crea el primer registro telefónico con el tipo de teléfono de PRI:

```
POST /cmx/cs/<ors>/Person?systemName=Admin
{
  "firstName": "John",
  "TelephoneNumbers": {
    "item": [
      {
        "phoneExtNum": "e1",
        "phoneNum": "12304567",
        "phoneTypeCd": {
          "phoneType": "PRI"
        }
      }
    ]
  }
}
```

Cuando se ejecuta la siguiente solicitud de publicación para crear el segundo registro telefónico, el tipo de teléfono se establece en PRI para el segundo registro telefónico y el tipo de teléfono de la primera solicitud de publicación se cambia a OTH.

```
POST /cmx/cs/<ors>/Person/rowidObject?systemName=Admin
{
  "TelephoneNumbers": {
    "item": [
      {

```

```

        "phoneExtNum": "e2",
        "phoneNum": "23045678",
        "phoneTypeCd": {
            "phoneType": "PRI"
        }
    }
]
}
}

```

- Escenario 2. Este escenario incluye una solicitud POST que crea dos registros telefónicos. La siguiente solicitud POST de ejemplo incluye dos registros telefónicos con el tipo de teléfono de PRI:

```

POST /cmx/cs/<ors>/Person?systemName=Admin
{
  "firstName": "John",
  "TelephoneNumbers": {
    "item": [
      {
        "phoneExtNum": "e1",
        "phoneNum": "12304567",
        "phoneTypeCd": {
          "phoneType": "PRI"
        }
      },
      {
        "phoneExtNum": "e2",
        "phoneNum": "23045678",
        "phoneTypeCd": {
          "phoneType": "PRI"
        }
      }
    ]
  }
}
}

```

Cuando se ejecuta esta solicitud POST, el tipo de teléfono para el primer registro telefónico se establece en PRI. Para el segundo registro telefónico, el tipo de teléfono se establece en OTH.

APÉNDICE A

Uso de las API de REST para añadir registros

Este apéndice incluye los siguientes temas:

- [Resumen sobre el uso de las API de REST para añadir registros, 328](#)
- [Estructura de una entidad de negocio de persona, 329](#)
- [Paso 1. Obtener información sobre el esquema, 329](#)
- [Paso 2. Crear un registro, 335](#)
- [Paso 3. Leer el registro, 337](#)

Resumen sobre el uso de las API de REST para añadir registros

Después de crear un modelo de entidad de negocio y configurar la estructura de entidad de negocio, puede usar las API de REST para añadir registros.

Las secciones siguientes utilizan el ejemplo de la entidad de negocio de persona para ilustrar cómo se pueden añadir registros mediante las API de REST. La entidad de negocio de persona contiene datos de los empleados de su empresa.

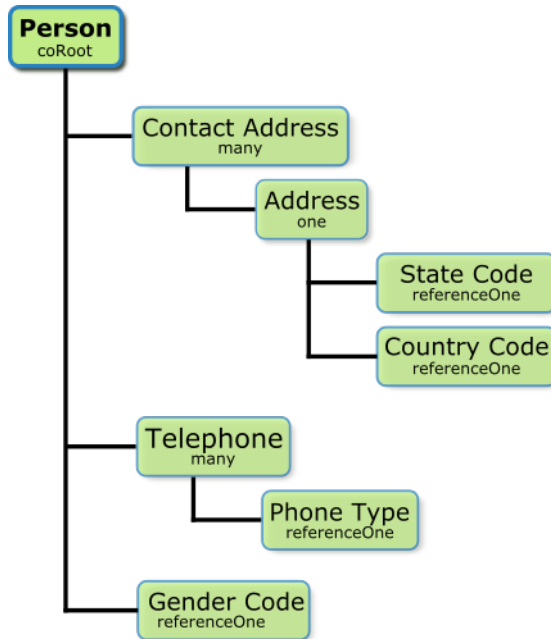
Utilice las siguientes API para añadir detalles de sus empleados:

1. Obtenga información sobre el esquema. Utilice la API de REST Obtener metadatos para conseguir información sobre la estructura de datos de una entidad de negocio, incluida la estructura, la lista de campos, los tipos de campos y los detalles de los campos de búsqueda. O puede acceder a los archivos de definición de esquema XML (XSD) que describen qué elementos y atributos se pueden utilizar. Los archivos XSD residen en `http://<host>:<puerto>/cmx/csfiles`.
2. Cree un registro. Utilice la API de REST de creación de registros para crear el registro.
3. Lea los datos del registro que haya añadido. Utilice la API de REST de lectura de registros para recuperar los datos del registro.

Estructura de una entidad de negocio de persona

Agregaremos un registro de persona mediante las API de REST. El nodo raíz de Persona es el nodo que ocupa la posición más alta en la estructura de entidades de negocio de persona. Debajo del nodo raíz Persona existen nodos para los detalles de los empleados, como el sexo, la dirección y el teléfono.

La siguiente imagen muestra la estructura de la entidad de negocio de Persona:



Persona es el nodo raíz de la entidad de negocio de Persona. El tipo de nodo, que aparece debajo del nombre de nodo, indica la relación entre el nodo superior y el nodo secundario. Existe una relación de uno a uno entre la Dirección de contacto y la Dirección, que indica que cada dirección de contacto solo puede tener asociada una dirección. Existe una relación de uno a varios entre la Persona y el Teléfono, que indica que un registro de persona puede tener asociados varios números de teléfono. Existe una relación de uno a uno entre la Persona y el Sexo, que indica que un registro de persona solo puede tener un valor de sexo. Los valores de sexo residen en una tabla de búsqueda. Del mismo modo, los valores del código de provincia/estado y de país residen en varias tablas de búsqueda.

Paso 1. Obtener información sobre el esquema

Utilice la API de REST para la obtención de metadatos para conseguir información sobre un esquema. La API para la obtención de metadatos devuelve la estructura de datos de una entidad de negocio. Los metadatos enumeran los campos de entidad de negocio, los tipos de campos y los detalles de los campos de búsqueda.

La URL para la obtención de metadatos tiene el siguiente formato:

```
http://<host>:<port>/<context>/<database ID>/<business entity>?action=meta
```

La siguiente solicitud de ejemplo recupera la información de metadatos para la entidad de negocio de persona:

```
GET http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/Person?action=meta
```

Obtener la respuesta de los metadatos

El siguiente ejemplo muestra algunos fragmentos de la estructura de datos de la entidad de negocio de persona:

```
{
  "object": {
    "field": [
      {
        "allowedValues": [
          "Person"
        ],
        "name": "partyType",
        "label": "Party Type",
        "dataType": "String",
        "length": 255,
        "totalDigits": 0,
        "fractionDigits": 0,
        "readOnly": false,
        "required": false,
        "system": false
      },
      {
        "name": "imageUrl",
        "label": "Image URL",
        "dataType": "ImageURL",
        "length": 255,
        "totalDigits": 0,
        "fractionDigits": 0,
        "readOnly": false,
        "required": false,
        "system": false
      },
      {
        "name": "statusCd",
        "label": "Status Cd",
        "dataType": "String",
        "length": 2,
        "totalDigits": 0,
        "fractionDigits": 0,
        "readOnly": false,
        "required": false,
        "system": false
      },
      {
        "name": "displayName",
        "label": "Display Name",
        "dataType": "String",
        "length": 200,
        "totalDigits": 0,
        "fractionDigits": 0,
        "readOnly": false,
        "required": false,
        "system": false
      },
      {
        "name": "birthdate",
        "label": "Birthdate",
        "dataType": "Date",
        "length": 0,
        "totalDigits": 0,
        "fractionDigits": 0,
        "readOnly": false,
        "required": false,
        "system": false
      },
      {
        "name": "firstName",
        "label": "First Name",
        "dataType": "String",
        "length": 50,

```

```

        "totalDigits": 0,
        "fractionDigits": 0,
        "readOnly": false,
        "required": false,
        "system": false
    },
    {
        "name": "lastName",
        "label": "Last Name",
        "dataType": "String",
        "length": 50,
        "totalDigits": 0,
        "fractionDigits": 0,
        "readOnly": false,
        "required": false,
        "system": false
    },
    {
        "name": "middleName",
        "label": "Middle Name",
        "dataType": "String",
        "length": 50,
        "totalDigits": 0,
        "fractionDigits": 0,
        "readOnly": false,
        "required": false,
        "system": false
    },
    {
        "name": "dirtyIndicator",
        "label": "Dirty Indicator",
        "dataType": "Integer",
        "length": 38,
        "totalDigits": 0,
        "fractionDigits": 0,
        "readOnly": true,
        "required": false,
        "system": true
    },
    {
        "name": "hubStateInd",
        "label": "Hub State Ind",
        "dataType": "Integer",
        "length": 38,
        "totalDigits": 0,
        "fractionDigits": 0,
        "readOnly": true,
        "required": false,
        "system": true
    },
    {
        "name": "cmDirtyInd",
        "label": "Content metadata dirty Ind",
        "dataType": "Integer",
        "length": 38,
        "totalDigits": 0,
        "fractionDigits": 0,
        "readOnly": true,
        "required": false,
        "system": true
    },
    {
        "name": "lastRowidSystem",
        "label": "Last Rowid System",
        "dataType": "String",
        "length": 14,
        "totalDigits": 0,
        "fractionDigits": 0,
        "readOnly": true,
        "required": false,
        "system": true
    }

```

```

    },
    -----
    {
        "name": "genderCd",
        "label": "Gender Cd",
        "dataType": "lookup",
        "readOnly": false,
        "required": false,
        "system": false,
        "lookup": {
            "link": [
                {
                    "href": "http://localhost:8080/cmxc/cs/localhost-hub101-
ds_uil/LUGender?action=list&idlabel=genderCode%3AgenderDisp",
                    "rel": "lookup"
                },
                {
                    "href": "http://localhost:8080/cmxc/cs/localhost-hub101-
ds_uil/LUGender?action=list",
                    "rel": "list"
                }
            ],
            "object": "LUGender",
            "key": "genderCode",
            "value": "genderDisp"
        }
    }
},
    -----

```

```

    "child": [
        {
            "field": [
                {
                    "name": "cityName",
                    "label": "City Name",
                    "dataType": "String",
                    "length": 100,
                    "totalDigits": 0,
                    "fractionDigits": 0,
                    "readOnly": false,
                    "required": false,
                    "system": false
                },
                {
                    "name": "addressLine2",
                    "label": "Address Line2",
                    "dataType": "String",
                    "length": 100,
                    "totalDigits": 0,
                    "fractionDigits": 0,
                    "readOnly": false,
                    "required": false,
                    "system": false
                },
                {
                    "name": "addressLine1",
                    "label": "Address Line1",
                    "dataType": "String",
                    "length": 100,
                    "totalDigits": 0,
                    "fractionDigits": 0,
                    "readOnly": false,
                    "required": false,
                    "system": false
                },
                {
                    "name": "isValidInd",

```

```

        "label": "Is Valid Ind",
        "dataType": "String",
        "length": 1,
        "totalDigits": 0,
        "fractionDigits": 0,
        "readOnly": false,
        "required": false,
        "system": false
    },
    {
        "name": "postalCd",
        "label": "Postal Cd",
        "dataType": "String",
        "length": 10,
        "totalDigits": 0,
        "fractionDigits": 0,
        "readOnly": false,
        "required": false,
        "system": false
    }
},

```

```

    {
        "name": "countryCode",
        "label": "Country Code",
        "dataType": "lookup",
        "readOnly": false,
        "required": false,
        "system": false,
        "dependents": [
            "Person.Address.Address.stateCd"
        ],
        "lookup": {
            "link": [
                {
                    "href": "http://localhost:8080/cmx/cs/localhost-hub101-ds_ui1/LUCountry?action=list",
                    "rel": "list"
                },
                {
                    "href": "http://localhost:8080/cmx/cs/localhost-hub101-ds_ui1/LUCountry?action=list&idlabel=countryCode%3AcountryNameDisp",
                    "rel": "lookup"
                }
            ],
            "object": "LUCountry",
            "key": "countryCode",
            "value": "countryNameDisp"
        }
    },
    {
        "name": "stateCd",
        "label": "State Cd",
        "dataType": "lookup",
        "readOnly": false,
        "required": false,
        "system": false,
        "parents": [
            "Person.Address.Address.countryCode"
        ],
        "lookup": {
            "link": [
                {
                    "href": "http://localhost:8080/cmx/cs/localhost-hub101-ds_ui1/LUCountry/{Person.Address.Address.countryCode}/LUState?action=list",
                    "rel": "list"
                }
            ],
            "object": "LUState",
            "key": "stateCd",
            "value": "stateNameDisp"
        }
    }
}

```

```

                "href": "http://localhost:8080/cmx/cs/localhost-hub101-ds_ui1/LUCountry/{Person.Address.Address.countryCode}/LUState?action=list&idlabel=stateAbbreviation%3AstateNameDisp",
                "rel": "lookup"
            }
        ],
        "object": "LUCountry.LUState",
        "key": "stateAbbreviation",
        "value": "stateNameDisp"
    }
}
],
"name": "Address",
"label": "Address",
"many": false
}
],
"name": "Address",
"label": "Contact Address",
"many": true
},
{
    "field": [
        {
            "name": "phoneNum",
            "label": "Phone Number",
            "dataType": "String",
            "length": 13,
            "totalDigits": 0,
            "fractionDigits": 0,
            "readOnly": false,
            "required": false,
            "system": false
        },
        {
            "name": "phoneTypeCd",
            "label": "Phone Type",
            "dataType": "lookup",
            "readOnly": false,
            "required": false,
            "system": false,
            "lookup": {
                "link": [
                    {
                        "href": "http://localhost:8080/cmx/cs/localhost-hub101-ds_ui1/LUPhoneType?action=list&idlabel=phoneType%3AphoneTypeDisp",
                        "rel": "lookup"
                    },
                    {
                        "href": "http://localhost:8080/cmx/cs/localhost-hub101-ds_ui1/LUPhoneType?action=list",
                        "rel": "list"
                    }
                ],
                "object": "LUPhoneType",
                "key": "phoneType",
                "value": "phoneTypeDisp"
            }
        }
    ],
    "name": "Telephone",
    "label": "Telephone",
    "many": true
}
],
"name": "Person",
"label": "Person",

```

```

    "many": false
  }
}

```

Paso 2. Crear un registro

Utilice la API de REST de creación de registros para crear un registro. El nombre de la entidad de negocio y el nombre del sistema de origen son parámetros obligatorios. Envíe los datos para el registro en el cuerpo de la solicitud.

La dirección URL de creación de registros tiene el siguiente formato:

```

http://<host>:<port>/<context>/<database ID>/<business entity>?systemName=<name of the source system>

```

El parámetro `systemName` es un parámetro obligatorio y especifica el nombre del sistema de origen.

La entidad de negocio de persona tiene el nodo raíz `Persona` y los nodos de dirección de segundo nivel, sexo y teléfono.

La siguiente solicitud de ejemplo crea un registro de persona:

```

POST http://localhost:8080/cmxc/cs/localhost-hub101-ds_uil/Person?systemName=Admin
{
  "firstName": "Boris",
  "lastName": "Isaac",
  "genderCd": {
    "genderCode": "M"
  },
  "Address": {
    "item": [
      {
        "Address": {
          "addressLine1": "B-203, 101 Avenue, New York",
          "stateCd": {
            "stateAbbreviation": "NY"
          },
          "countryCode": {
            "countryCode": "US"
          }
        }
      }
    ]
  },
  "Telephone": {
    "item": [
      {
        "phoneNum": "1234567",
        "phoneTypeCd": {
          "phoneType": "HOM"
        }
      },
      {
        "phoneNum": "7654321",
        "phoneTypeCd": {
          "phoneType": "MOB"
        }
      }
    ]
  }
}

```

El cuerpo de la solicitud especifica los detalles siguientes del registro de persona:

- Nombre.

- Apellido.
- Sexo.
- Dirección con el código de provincia/estado y el código de país.
- Números de teléfono junto con el tipo de teléfono como, por ejemplo, fijo o móvil.

Crear respuestas de registro

La siguiente respuesta de ejemplo muestra la respuesta después de crear correctamente un registro de persona:

```
{
  "Person": {
    "key": {
      "rowid": "658248",
      "sourceKey": "66240000025000"
    },
    "rowidObject": "658248",
    "genderCd": {
      "key": {
        "rowid": "2"
      },
      "rowidObject": "2"
    },
    "Address": {
      "link": [],
      "item": [
        {
          "key": {
            "rowid": "101526",
            "sourceKey": "66240000028000"
          },
          "rowidObject": "101526",
          "Address": {
            "key": {
              "rowid": "121506",
              "sourceKey": "66240000027000"
            },
            "rowidObject": "121506",
            "countryCode": {
              "key": {
                "rowid": "233"
              },
              "rowidObject": "233"
            },
            "stateCd": {
              "key": {
                "rowid": "52"
              },
              "rowidObject": "52"
            }
          }
        }
      ]
    },
    "Telephone": {
      "link": [],
      "item": [
        {
          "key": {
            "rowid": "20967",
            "sourceKey": "66240000029000"
          },
          "rowidObject": "20967",
          "phoneTypeCd": {
            "key": {

```


Nota: Si un flujo de trabajo está iniciado cuando crea un registro, este tendrá el estado pendiente. De forma predeterminada, la solicitud de lectura de registros lee los registros activos. Utilice el parámetro `recordStates` para especificar el estado pendiente del registro.

La siguiente solicitud de ejemplo lee los detalles de un registro consolidado:

```
GET http://localhost:8080/cmx/cs/localhost-hub101-ds_ui1/Person/658248?
depth=3&recordStates=PENDING
```

Leer la respuesta del registro

La siguiente respuesta de ejemplo muestra los detalles del registro que ha añadido:

```
{
  "link": [
    {
      "href": "http://localhost:8080/cmx/cs/localhost-hub101-ds_ui1/Person/658248",
      "rel": "self"
    },
    {
      "href": "http://localhost:8080/cmx/cs/localhost-hub101-ds_ui1/Person/658248?
depth=2",
      "rel": "children"
    }
  ],
  "rowidObject": "658248",
  "label": "Person",
  "partyType": "Person",
  "displayName": "BORIS ISAAC",
  "firstName": "BORIS",
  "lastName": "ISAAC",
  "genderCd": {
    "link": [
      {
        "href": "http://localhost:8080/cmx/cs/localhost-hub101-ds_ui1/Person/
658248/genderCd/2",
        "rel": "self"
      },
      {
        "href": "http://localhost:8080/cmx/cs/localhost-hub101-ds_ui1/Person/
658248",
        "rel": "parent"
      },
      {
        "href": "http://localhost:8080/cmx/cs/localhost-hub101-ds_ui1/Person/
658248/genderCd/2?depth=2",
        "rel": "children"
      }
    ]
  },
  "rowidObject": "2",
  "label": "LU Gender",
  "genderCode": "M",
  "genderDisp": "MALE"
},
"Address": {
  "link": [
    {
      "href": "http://localhost:8080/cmx/cs/localhost-hub101-ds_ui1/Person/
658248/Address",
      "rel": "self"
    },
    {
      "href": "http://localhost:8080/cmx/cs/localhost-hub101-ds_ui1/Person/
658248",
      "rel": "parent"
    }
  ]
},
"firstRecord": 1,
"pageSize": 10,
```

```

"searchToken": "SVR1.PCWJ",
"item": [
  {
    "link": [
      {
        "href": "http://localhost:8080/cmxcsllocalhost-hub101-ds_ui1/
Person/658248/Address",
        "rel": "parent"
      },
      {
        "href": "http://localhost:8080/cmxcsllocalhost-hub101-ds_ui1/
Person/658248/Address/101526?depth=2",
        "rel": "children"
      },
      {
        "href": "http://localhost:8080/cmxcsllocalhost-hub101-ds_ui1/
Person/658248/Address/101526",
        "rel": "self"
      }
    ],
    "rowidObject": "101526",
    "label": "Contact Address",
    "Address": {
      "link": [
        {
          "href": "http://localhost:8080/cmxcsllocalhost-hub101-
ds_ui1/Person/658248/Address/101526/Address/121506?depth=2",
          "rel": "children"
        },
        {
          "href": "http://localhost:8080/cmxcsllocalhost-hub101-
ds_ui1/Person/658248/Address/101526",
          "rel": "parent"
        },
        {
          "href": "http://localhost:8080/cmxcsllocalhost-hub101-
ds_ui1/Person/658248/Address/101526/Address/121506",
          "rel": "self"
        }
      ],
      "rowidObject": "121506",
      "label": "Address",
      "addressLine1": "B-203, 101 Avenue, New York",
      "countryCode": {
        "link": [
          {
            "href": "http://localhost:8080/cmxcsllocalhost-hub101-
ds_ui1/Person/658248/Address/101526/Address/121506/countryCode",
            "rel": "self"
          },
          {
            "href": "http://localhost:8080/cmxcsllocalhost-hub101-
ds_ui1/Person/658248/Address/101526/Address/121506",
            "rel": "parent"
          },
          {
            "href": "http://localhost:8080/cmxcsllocalhost-hub101-
ds_ui1/Person/658248/Address/101526/Address/121506/countryCode?depth=2",
            "rel": "children"
          }
        ],
        "countryCode": "US"
      },
      "stateCd": {
        "link": [
          {
            "href": "http://localhost:8080/cmxcsllocalhost-hub101-
ds_ui1/Person/658248/Address/101526/Address/121506",
            "rel": "parent"
          }
        ]
      }
    }
  }
]

```

```

                "href": "http://localhost:8080/cmxc/cs/localhost-hub101-
ds_ui1/Person/658248/Address/101526/Address/121506/stateCd?depth=2",
                "rel": "children"
            },
            {
                "href": "http://localhost:8080/cmxc/cs/localhost-hub101-
ds_ui1/Person/658248/Address/101526/Address/121506/stateCd",
                "rel": "self"
            }
        ],
        "stateAbbreviation": "NY"
    }
}
]
},
"Telephone": {
    "link": [
        {
            "href": "http://localhost:8080/cmxc/cs/localhost-hub101-ds_ui1/Person/
658248",
            "rel": "parent"
        },
        {
            "href": "http://localhost:8080/cmxc/cs/localhost-hub101-ds_ui1/Person/
658248/Telephone",
            "rel": "self"
        }
    ],
    "firstRecord": 1,
    "pageSize": 10,
    "searchToken": "SVR1.PCWK",
    "item": [
        {
            "link": [
                {
                    "href": "http://localhost:8080/cmxc/cs/localhost-hub101-ds_ui1/
Person/658248/Telephone",
                    "rel": "parent"
                },
                {
                    "href": "http://localhost:8080/cmxc/cs/localhost-hub101-ds_ui1/
Person/658248/Telephone/20967",
                    "rel": "self"
                },
                {
                    "href": "http://localhost:8080/cmxc/cs/localhost-hub101-ds_ui1/
Person/658248/Telephone/20967?depth=2",
                    "rel": "children"
                }
            ],
            "rowidObject": "20967",
            "label": "Telephone",
            "phoneNum": "1234567",
            "phoneTypeCd": {
                "link": [
                    {
                        "href": "http://localhost:8080/cmxc/cs/localhost-hub101-
ds_ui1/Person/658248/Telephone/20967",
                        "rel": "parent"
                    },
                    {
                        "href": "http://localhost:8080/cmxc/cs/localhost-hub101-
ds_ui1/Person/658248/Telephone/20967/phoneTypeCd/8",
                        "rel": "self"
                    },
                    {
                        "href": "http://localhost:8080/cmxc/cs/localhost-hub101-
ds_ui1/Person/658248/Telephone/20967/phoneTypeCd/8?depth=2",
                        "rel": "children"
                    }
                ]
            }
        }
    ]
}

```

```

    ],
    "rowidObject": "8",
    "label": "LU Phone Type",
    "phoneTypeDisp": "HOME",
    "phoneType": "HOM"
  }
},
{
  "link": [
    {
      "href": "http://localhost:8080/cmx/cs/localhost-hub101-ds_ui1/Person/658248/Telephone/20968",
      "rel": "self"
    },
    {
      "href": "http://localhost:8080/cmx/cs/localhost-hub101-ds_ui1/Person/658248/Telephone/20968?depth=2",
      "rel": "children"
    },
    {
      "href": "http://localhost:8080/cmx/cs/localhost-hub101-ds_ui1/Person/658248/Telephone",
      "rel": "parent"
    }
  ],
  "rowidObject": "20968",
  "label": "Telephone",
  "phoneNum": "7654321",
  "phoneTypeCd": {
    "link": [
      {
        "href": "http://localhost:8080/cmx/cs/localhost-hub101-ds_ui1/Person/658248/Telephone/20968/phoneTypeCd/6",
        "rel": "self"
      },
      {
        "href": "http://localhost:8080/cmx/cs/localhost-hub101-ds_ui1/Person/658248/Telephone/20968",
        "rel": "parent"
      },
      {
        "href": "http://localhost:8080/cmx/cs/localhost-hub101-ds_ui1/Person/658248/Telephone/20968/phoneTypeCd/6?depth=2",
        "rel": "children"
      }
    ]
  },
  "rowidObject": "6",
  "label": "LU Phone Type",
  "phoneTypeDisp": "MOBILE",
  "phoneType": "MOB"
}
]
}
}

```

APÉNDICE B

Uso de las API de REST para cargar archivos

Este apéndice incluye los siguientes temas:

- [Resumen sobre el uso de las API de REST para cargar archivos, 342](#)
- [API de REST para archivos, 343](#)
- [Componentes de archivo, 343](#)
- [Tipos de almacenamiento, 344](#)
- [Adjuntar archivos a registros, 345](#)
- [Adjuntar archivos a tareas, 346](#)
- [Cargar archivos de paquete de recursos, 349](#)

Resumen sobre el uso de las API de REST para cargar archivos

Puede usar las API de REST para cargar archivos en un tipo de almacenamiento. Tras cargar un archivo, puede adjuntarlo a un registro o a una tarea, o bien puede usarlo para localizar la interfaz de usuario de Data Director.

En función de cómo quiera usar los archivos, podría variar la combinación de las API de REST, los componentes de archivo y el tipo de almacenamiento que utilice. Por ejemplo, para adjuntar archivos a registros o tareas, debe crear los metadatos del archivo y crear el archivo en un almacenamiento temporal. Tras cargar el archivo, puede adjuntarlo a un registro en una base de datos o adjuntarlo a una tarea en el almacenamiento BPM. Para localizar la interfaz de usuario de Data Director, descargue el archivo ZIP, modifique los archivos comprimidos y, a continuación, cargue el paquete ZIP modificado en el almacenamiento BUNDLE.

API de REST para archivos

Es posible usar un conjunto de diversas API de REST con fines generales para cargar y administrar archivos para adjuntos o su localización.

En la siguiente tabla se enumeran las API de REST para los archivos:

API de REST	Descripción	Tipo de almacenamiento compatible
Enumerar metadatos de archivo	Devuelve una lista de los metadatos de archivo en un almacenamiento.	BPM o TEMP
Crear metadatos de archivo	Crea los metadatos de un archivo en un almacenamiento.	DB o TEMP
Obtener metadatos de archivo	Devuelve los metadatos de un archivo.	BPM, BUNDLE, DB o TEMP
Actualizar metadatos de archivo	Actualiza los metadatos de un archivo.	DB o TEMP
Cargar contenido de archivo	Carga el contenido de un archivo en un almacenamiento.	BUNDLE, DB o TEMP
Obtener contenido de archivo	Descarga el contenido de un archivo.	BPM, BUNDLE, DB o TEMP
Suprimir archivo	Elimina un archivo en un almacenamiento, incluidos los componentes asociados al archivo, como su contenido o sus metadatos.	BUNDLE, DB o TEMP

Componentes de archivo

Para adjuntar archivos a registros o tareas, cree metadatos para el archivo y, a continuación, cargue el contenido del archivo. Para localizar la interfaz de usuario de Data Director, descargue el archivo de paquete de recursos y, a continuación, cargue el archivo de paquete de recursos que se ha modificado.

Metadatos de archivo

Información sobre el archivo, como el nombre del archivo, el tipo de archivo y su contenido. En función del tipo de almacenamiento, es posible que se tengan que incluir otros parámetros, como el creador, la hora de creación y la fecha de carga.

Contenido de archivo

El contenido del archivo. Por ejemplo, el texto, la imagen, el documento o el paquete de recursos.

Tipos de almacenamiento

Cargue y almacene archivos en una implementación de almacenamiento compatible. El tipo de almacenamiento que utilice dependerá de si quiere localizar la interfaz de usuario de Data Director o adjuntar archivos a registros o tareas.

En la siguiente lista se describen los tipos de almacenamiento compatibles:

BPM

Almacena archivos adjuntos a tareas junto con los datos de la tarea. Cuando adjunte un archivo a una tarea, el proceso almacenará el archivo en un almacenamiento BPM del almacenamiento TEMP.

Los archivos almacenados en el almacenamiento BPM utilizan el formato de ID de archivo siguiente:

`taskId::filename.`

Nota: Para adjuntar un archivo a tareas activadas o tareas existentes, en la herramienta de aprovisionamiento, permita los adjuntos en activadores de tarea, tipos de tarea y acciones de tarea. Para obtener más información, consulte la *Guía de la herramienta de aprovisionamiento de Multidomain MDM*.

BUNDLE

Almacena archivos de paquete de recursos que localiza la interfaz de usuario de Data Director.

Los archivos almacenados en el almacenamiento BUNDLE utilizan el formato de ID de archivo siguiente:

`besMetadata.`

DB

Almacena adjuntos de archivo para registros en la tabla C_REPOS_ATTACHMENTS. Cuando adjunte un archivo a un registro, el proceso almacenará el archivo en un almacenamiento DB del almacenamiento TEMP.

Los archivos almacenados en el almacenamiento DB utilizan el formato de ID de archivo siguiente:

`DB_<RowID>.`

Nota: Para adjuntar un archivo a un registro, en la herramienta de aprovisionamiento, configure un campo con `FileAttachment` como el tipo de dato. Si desea más información sobre cómo se configura el tipo de dato, consulte la *Guía de la herramienta de aprovisionamiento de Multidomain MDM*.

TEMP

Almacena temporalmente los archivos en la tabla C_REPOS_ATTACHMENTS y marca los archivos como TEMP. Los archivos se eliminan del almacenamiento TEMP después de que se hayan cargado correctamente al almacenamiento BPM o DB o después de que haya pasado el tiempo de caducidad preconfigurado.

Los archivos almacenados en el almacenamiento TEMP utilizan el formato de ID de archivo siguiente:

`TEMP_<ROWID_ATTACHMENT>.`

Si desea más información sobre cómo se configura el tiempo de caducidad, consulte la *Guía de configuración de Multidomain MDM*.

Adjuntar archivos a registros

Antes de adjuntar un archivo a un registro, cree los metadatos del archivo y, a continuación, cargue el archivo en el almacenamiento temporal.

1. Para crear los metadatos de un archivo, utilice la API de REST de creación de metadatos de archivos con TEMP como tipo de almacenamiento.

Por ejemplo, con la siguiente solicitud se crean los metadatos del archivo `Document_3.pdf`:

```
POST http://localhost:8080/cmx/file/localhost-orcl-MDM_SAMPLE/TEMP
Content-Type: application/json
{
  "fileName": "Document_3.pdf",
  "fileType": "pdf",
  "fileContentType": "application/pdf"
}
```

Nota: Cree siempre los metadatos de archivos en el almacenamiento TEMP.

La API de REST de creación de metadatos de archivos devuelve un ID para el archivo. El ID de archivo se presenta en el siguiente formato: <Tipo de almacenamiento>_<RowID>. Donde RowID hace referencia al ID de fila del archivo que se va a cargar en el almacenamiento.

En el ejemplo, la llamada de API devuelve el siguiente ID para el archivo `Document_3.pdf`:

```
TEMP_SVR1.OJU3
```

El ID de archivo se puede usar para cargar el archivo, adjuntarlo, actualizarlo, descargarlo y eliminarlo.

2. Para cargar el archivo, utilice la API de REST de carga de contenido de archivo con TEMP como tipo de almacenamiento.

Por ejemplo, con la siguiente solicitud se carga el archivo en el almacenamiento TEMP:

```
PUT http://localhost:8080/cmx/file/localhost-orcl-MDM_SAMPLE/TEMP/TEMP_SVR1.OJU3/
content
Content-Type: application/octet-stream
<file object (upload using REST client)>
```

Nota: Después de cargar un archivo, el almacenamiento TEMP almacena el archivo durante un período de tiempo preconfigurado de 60 minutos. El archivo se debe adjuntar a un registro antes de que caduque el período preconfigurado.

3. Para crear un registro y adjuntar el archivo a un registro nuevo, utilice la API de REST de creación de registros.

Por ejemplo, con la siguiente solicitud se crea un registro y se adjunta el archivo con el ID de archivo

```
TEMP_SVR1.OJU3:
```

```
POST http://localhost:8080/cmx/cs/localhost-orcl-MDM_SAMPLE/Person?systemName=Admin
Content-Type: application/json
{
  "frstNm": "John",
  "lstNm": "Smith",
  "addrLn1": "2100 Breverly Road",
  "addrTyp": {
    "addrTyp": "Billing",
    "addrTypDesc": "Billing"
  },
  "cntryCd": {
    "cntryCd": "AX",
    "cntryDesc": "Aland"
  },
  "attachments": {
    "item": [
      {
        "fileId": "TEMP_SVR1.OJU3"
      }
    ]
  }
}
```

```
}  
}
```

Nota: Cuando adjunte un archivo a un registro, el proceso almacenará el archivo en la base de datos. El ID del archivo cambiará a DB_<RowID>, donde DB indica que el archivo se encuentra almacenado en la base de datos.

4. Para reemplazar un archivo que está adjunto a un registro, utilice la API de REST de carga de contenido de archivo con DB como tipo de almacenamiento.

Por ejemplo, con la siguiente solicitud se reemplaza el archivo adjunto con el ID de archivo

DB_SVR1.OJU3 de la base de datos:

```
PUT http://localhost:8080/cmx/file/localhost-orcl-MDM_SAMPLE/DB/DB_SVR1.OJU3/content  
Content-Type: application/octet-stream  
<file object (upload using REST client)>
```

Nota: El tipo de almacenamiento de la URL de la solicitud es DB.

5. Para editar los metadatos de un archivo después de adjuntarlo a un registro, utilice la API de REST de actualización de metadatos de archivo con DB como tipo de almacenamiento.

Por ejemplo, con la siguiente solicitud se actualizan los metadatos de un archivo asociado al ID de archivo DB_SVR1.OJU3 de la base de datos:

```
PUT http://localhost:8080/cmx/file/localhost-orcl-MDM_SAMPLE/DB/DB_SVR1.OJU3  
Content-Type: application/json  
{  
  "fileName": "Document_4.pdf",  
  "fileType": "pdf",  
  "fileContentType": "application/pdf"  
}
```

6. Para descargar un archivo que está adjunto a un registro, utilice la API de REST para obtener el contenido de archivo con DB como tipo de almacenamiento.

Por ejemplo, con la siguiente solicitud se descarga un archivo que está asociado al ID de archivo

DB_SVR1.OJU3 desde la base de datos:

```
GET http://localhost:8080/cmx/file/localhost-orcl-MDM_SAMPLE/DB/DB_SVR1.OJU3/content
```

7. Para eliminar un archivo que está adjunto a un registro, utilice la API de REST de eliminación de archivos con DB como tipo de almacenamiento.

Por ejemplo, con la siguiente solicitud se elimina un archivo que está asociado al ID de archivo

DB_SVR1.OJU3 de la base de datos:

```
DELETE http://localhost:8080/cmx/file/localhost-orcl-MDM_SAMPLE/DB/DB_SVR1.OJU3
```

Adjuntar archivos a tareas

Cree los metadatos de un archivo y, a continuación, cargue el contenido del archivo en un almacenamiento temporal. Después de cargar el archivo, adjúntelo a una tarea activada o a una tarea existente.

Nota: Para adjuntar un archivo a tareas activadas o tareas existentes, en la herramienta de aprovisionamiento, permita los adjuntos en activadores de tarea, tipos de tarea y acciones de tarea. Para obtener más información, consulte la *Guía de la herramienta de aprovisionamiento de Multidomain MDM*.

1. Para crear los metadatos de un archivo, utilice la API de REST de creación de metadatos de archivos con TEMP como tipo de almacenamiento.

Por ejemplo, con la siguiente solicitud se crean los metadatos del archivo file1.txt:

```
POST http://localhost:8080/cmx/file/localhost-orcl-MDM_SAMPLE/TEMP  
  
{  
  "fileName": "file1.txt",
```

```

    "fileType": "text",
    "fileContentType": "text/plain"
  }

```

Nota: Cree siempre los metadatos de archivos en el almacenamiento TEMP.

La API de REST de creación de metadatos de archivos devuelve un ID para el archivo. El ID de archivo se presenta en el siguiente formato: <Tipo de almacenamiento>_<RowID>. Donde RowID hace referencia al ID de fila del archivo que se va a cargar en el almacenamiento.

En el ejemplo, la llamada de API devuelve el siguiente ID para file1.txt: TEMP_SVR1.1VDVS

El ID de archivo se puede usar para cargar el archivo, adjuntarlo, actualizarlo y eliminarlo.

2. Para cargar el archivo, utilice la API de REST de carga de contenido de archivo con TEMP como tipo de almacenamiento.

Por ejemplo, con la siguiente solicitud se carga el archivo en el almacenamiento TEMP:

```

PUT http://localhost:8080/cmx/file/localhost-orcl-MDM_SAMPLE/TEMP/TEMP_SVR1.1VDVS/
content

Test attachment content: file 1

```

Nota: Después de cargar un archivo, el almacenamiento TEMP almacena el archivo durante un período de tiempo preconfigurado de 60 minutos. El archivo se debe adjuntar a una tarea antes de que caduque el período preconfigurado.

3. Adjunte el archivo a la tarea que se activó mientras administraba los registros.
 - Para adjuntar el archivo a la tarea que se activa al crear un registro, use la API de REST de creación de entidades de negocio con el parámetro `taskattachments`.

Por ejemplo, con la siguiente solicitud se crea un registro y se adjunta el archivo con el ID de archivo TEMP_SVR1.1VDVS:

```

POST http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/Person?
systemName=Admin&taskAttachments=TEMP_SVR1.1VDVS
Content-Type: application/json

{
  firstName: "John",
  lastName: "Smith",
  Phone: {
    item: [
      {
        phoneNumber: "111-11-11"
      }
    ]
  }
}

```

- Para adjuntar el archivo a la tarea que se activa al actualizar un registro, use la API de REST de actualización de entidades de negocio con el parámetro `taskattachments`.

Por ejemplo, con la siguiente solicitud se actualiza un registro y se adjunta el archivo con el ID de archivo TEMP_SVR1.1VDVS:

```

PUT http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/Person/233?
systemName=Admin&taskAttachments=TEMP_SVR1.1VDVS
{
  rowidObject: "233",
  firstName: "BOB",
  lastName: "LLOYD",
  Phone: {
    item: [
      {
        rowidObject: "164",
        phoneNumber: "777-77-77",
        $original: {
          phoneNumber: "(336) 366-4936"
        }
      }
    ]
  }
}

```

```

    }
  ]
},
$original: {
  firstName: "DUNN"
}
}

```

- Para adjuntar el archivo a la tarea que se activa al fusionar un registro, use la API de REST de fusión de entidades de negocio con el parámetro `taskattachments`.

Por ejemplo, con la siguiente solicitud se fusiona un registro y se adjunta el archivo con el ID de archivo `TEMP_SVR1.1VDVS`:

```

POST http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/Person/2478245?
action=merge&taskAttachments=TEMP_SVR1.1VDVS
Content-Type: application/<json/xml>
{
  keys: [
    {
      rowid: "2478246"
    }
  ],
  overrides: {
    Person: {
      firstName: "Charlie"
    }
  }
}

```

- Para adjuntar el archivo a la tarea que se activa al anular la fusión de un registro, use la API de REST de anulación de fusión de entidades de negocio con el parámetro `taskattachments`.

Por ejemplo, con la siguiente solicitud se anula la fusión de un registro y se adjunta el archivo con el ID de archivo `TEMP_SVR1.1VDVS`:

```

POST http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/Person/2478248?
action=unmerge&taskAttachments=TEMP_SVR1.1VDVS
{
  rowid: "4880369"
}

```

4. Adjunte el archivo a una tarea existente.

- Para adjuntar el archivo al actualizar una tarea, use la API de REST de actualización de tareas con `adjuntos` en el cuerpo de la solicitud.

Por ejemplo, con la siguiente solicitud se actualiza una tarea y se adjunta el archivo con el ID de archivo `TEMP_SVR1.1VDVS`:

```

PUT http://localhost:8080/cmx/cs/localhost-orcl-MDM_SAMPLE/task/urn:b4p2:15934
{
  taskType: {
    name: "UpdateWithApprovalWorkflow"
  },
  taskId: "urn:b4p2:15934",
  owner: "John",
  title: "Smoke test task - updated",
  comments: "Smoke testing - updated",
  "attachments": [
    {
      "id": "TEMP_SVR1.1VDVS"
    }
  ],
  ...
}

```

- Para adjuntar el archivo al ejecutar una acción de tarea, use la API de REST de ejecución de acciones de tarea con `adjuntos` en el cuerpo de la solicitud.

Por ejemplo, con la siguiente solicitud se ejecuta una acción de tarea y se adjunta el archivo con el ID de archivo TEMP_SVR1.1VDVS:

```
POST http://localhost:8080/cmx/cs/localhost-orcl-MDM_SAMPLE/task/urn:b4p2:15934?
taskAction=Cancel
{
  taskType: {
    name:"UpdateWithApprovalWorkflow",
    taskAction: [{name: "Cancel"}]
  },
  taskId: "urn:b4p2:15934",
  owner: "manager",
  title: "Smoke test task 222",
  comments: "Smoke testing",
  "attachments": [
    {
      "id": "TEMP_SVR1.1VDVS"
    }
  ],
  ...
}
```

Después de adjuntar un archivo a una tarea, el proceso mueve el archivo del almacenamiento TEMP y lo almacena junto a los datos de tarea en el almacenamiento de BPM. El ID del archivo cambia a taskId::filename.

Cargar archivos de paquete de recursos

Para localizar la interfaz de usuario de Data Director, descargue el archivo ZIP de paquete de recursos, modifique los archivos del archivo ZIP y, a continuación, cargue el archivo ZIP modificado en el almacenamiento BUNDLE.

1. Para descargar el archivo ZIP de paquete de recursos, utilice la API de REST de obtención de contenido de archivo con BUNDLE como tipo de almacenamiento.

Por ejemplo, la siguiente solicitud descarga el archivo ZIP de paquete de recursos:

```
GET http://localhost:8080/cmx/file/localhost-orcl-MDM_SAMPLE/BUNDLE/besMetadata/
content
```

2. Modifique el archivo ZIP añadiendo archivos de paquete específicos de idioma.
Por ejemplo, para localizar los nombres de campo, las etiquetas y los nombres de las tablas a ruso, añada el archivo `besMetadata_ru.properties`.

3. Para cargar el archivo ZIP de paquete de recursos modificado, utilice la API de REST de carga de contenido de archivo con BUNDLE como tipo de almacenamiento.

Por ejemplo, la siguiente solicitud carga el archivo ZIP de paquete de recursos:

```
PUT http://localhost:8080/cmx/file/localhost-orcl-MDM_SAMPLE/BUNDLE/besMetadata/
content
Content-Type: application/octet-stream
Body: binary stream - zip file with besMetadata bundle
```

APÉNDICE C

Uso de las API de REST para administrar informes

Este apéndice incluye los siguientes temas:

- [Resumen sobre el uso de las API de REST para administrar informes, 350](#)
- [API de REST para informes, 351](#)
- [Configuración del informe, 351](#)
- [Datos del informe, 352](#)
- [Informes de detalle, 353](#)
- [Informes de fábrica, 354](#)
- [Informes personalizados, 357](#)
- [Solución de problemas con API de informes, 363](#)

Resumen sobre el uso de las API de REST para administrar informes

Puede usar las API de REST para administrar los informes de fábrica y personalizarlos para que recopilen información acerca de sus datos principales. Más adelante, en la herramienta de aprovisionamiento, puede rellenar los componentes del cuadro con los datos del informe. A continuación, en Data Director, los usuarios pueden ver los cuadros para analizar información acerca de sus datos principales.

Los informes de fábrica recopilan información acerca de criterios predefinidos. Si quiere recopilar más información acerca de los datos principales, use los informes personalizados.

API de REST para informes

Puede usar un conjunto de API de REST para administrar informes.

En la siguiente tabla se enumeran las API de REST para los informes:

API de REST	Descripción
Enumerar informes	Devuelve una lista de informes registrados y de la configuración de los informes.
Obtener configuración y datos del informe	Devuelve los datos y la configuración del informe.
Obtener configuración del informe e informes de detalle	Devuelve la configuración del informe e informes de detalle asociados.
Registrar informe	Registra informes personalizados. Devuelve el ID para el informe, que puede usarse en otras API.
Actualizar la configuración del informe	Actualiza la configuración de un informe.
Agregar o actualizar datos de informe	Agrega o actualiza entradas de datos en un informe.
Eliminar informe	Elimina un informe.
Ejecutar el trabajo de actualización de informe	Inicia los trabajos en lotes asociados con los informes de fábrica.
Obtener estado de trabajo de actualización de informe	Devuelve el estado de un trabajo de actualización de informe.

Configuración del informe

La configuración del informe contiene varios parámetros, como el nombre del informe, su descripción, sus dimensiones y el nombre del indicador. En el caso de los informes de fábrica está preconfigurada. Si quiere usar informes personalizados, debe especificar la configuración del informe de estos.

En la tabla siguiente se describen los parámetros en una configuración de informe:

Parámetro	Descripción
ROWID_RPT_CONFIG	ID de un informe.
DIMENSION_NAME_1	El nombre de una dimensión de datos en el informe. Por ejemplo, una dimensión puede ser nombre de usuario, tipo de tarea de revisión o tipo de entidad de negocio.
DIMENSION_NAME_2	Opcional. El nombre de una dimensión de datos en el informe. Por ejemplo, una dimensión puede ser nombre de usuario, tipo de tarea de revisión o tipo de entidad de negocio.
TIMEPERIOD_NAME	Opcional. Nombre del período de tiempo al que se aplican los datos. Por ejemplo, puede especificar Mes.

Parámetro	Descripción
RPT_NAME	Nombre del informe.
METRIC_NAME	Etiqueta del tipo de datos recabados por el informe. De forma predeterminada, el valor especificado para este parámetro aparece como el nombre del eje y en el cuadro. Por ejemplo, puede tener un informe que recaba información acerca de tareas cerradas por el usuario. Puede usar el nombre del indicador <code>Número de tareas cerradas</code> .
RPT_DESC	Descripción del informe.
RPT_TYPE	Indica si el informe es un informe de detalle. Si el valor es <code>null</code> , el informe es un informe raíz. Si el valor no es <code>null</code> , el informe es un informe de detalle.

Datos del informe

Agregue datos a un informe para poder completar cuadros con datos del informe. Para los informes de fábrica, use los trabajos de actualización de informes asociadas con los informes para recabar y agregar datos del informe. Si usa informes personalizados, debe agregar manualmente las entradas de datos en ellos.

En la siguiente tabla se describen los parámetros que debe configurar para cada entrada de datos de un informe:

Parámetro	Descripción
DIMENSION_NAME_1	El nombre de una dimensión de datos en el informe. Por ejemplo, una dimensión puede ser nombre de usuario, tipo de tarea de revisión o tipo de entidad de negocio.
DIMENSION_NAME_2	Opcional. El nombre de una dimensión de datos en el informe. Por ejemplo, una dimensión puede ser nombre de usuario, tipo de tarea de revisión o tipo de entidad de negocio.
TIMEPERIOD_NAME	Opcional. Nombre del período de tiempo al que se aplican los datos. Por ejemplo, puede especificar <code>Mes</code> .
METRIC_VALUE	Número que representa las dimensiones de datos. Por ejemplo, puede tener un informe que recaba información acerca de tareas cerradas por el usuario. Una dimensión del informe es el usuario John Smith, y otra dimensión es el tipo de tarea Revisión. El valor del indicador puede ser 5. Esto significa que John Smith cerró cinco tareas del tipo de tarea Revisión.
DRILLDOWN_RPT_ID	ID del informe de detalle. Si el valor es <code>null</code> , no se configura ningún informe de detalle.

En el siguiente ejemplo se muestran tres entradas de datos en un informe:

```
[
  {
    "DIMENSION_VALUE_1": "High",
    "DIMENSION_VALUE_2": "AVOSBeMerge",
    "TIMEPERIOD_VALUE": "null",
    "METRIC_VALUE": "3",
    "DRILLDOWN_RPT_ID": "null"
  },
  {
    "DIMENSION_VALUE_1": "High",
```



```

    "DIMENSION_VALUE_2": "AVOSBeReviewNoApprove",
    "TIMEPERIOD_VALUE": "null",
    "METRIC_VALUE": "0",
    "DRILLDOWN_RPT_ID": "null"
  },
  {
    "DIMENSION_VALUE_1": "High",
    "DIMENSION_VALUE_2": "AVOSBeUpdate",
    "TIMEPERIOD_VALUE": "null",
    "METRIC_VALUE": "0",
    "DRILLDOWN_RPT_ID": "null"
  }
  ...
]

```

Informes de detalle

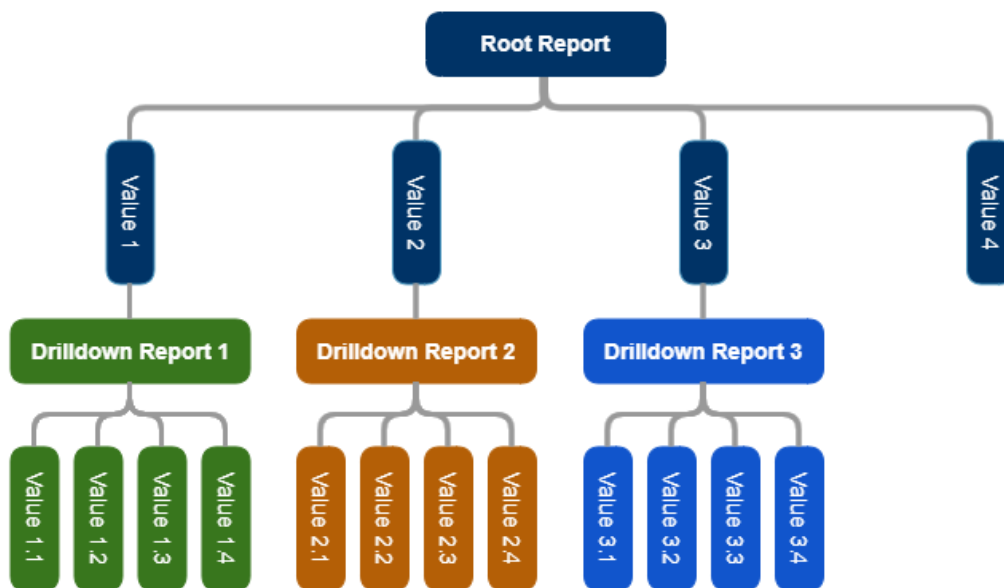
Un informe de detalle contiene otra capa de datos acerca de una entrada de datos en un informe. Puede usar los informes de detalle asociados a los informes de fábrica o configurar los informes de detalle para los informes personalizados.

Cuando se crean informes personalizados, se registra un informe raíz. Puede crear y asociar informes de detalle a entradas de datos en el informe raíz o entradas de datos en otros informes de detalle.

Por ejemplo, puede tener un informe que recaba información acerca del número de tareas por cada estado. Puede asociar un informe de detalle con el número de tareas cerradas. El informe de detalle contiene el número de tareas cerradas de cada usuario. También puede asociar un informe de detalle al número de tareas aprobadas. El informe de detalle contiene el número de tareas aprobadas de cada usuario.

En Data Director, los usuarios pueden ver un gráfico principal que muestra el número de tareas por cada estado. Cuando seleccionan el número de tareas cerradas, van hasta un gráfico de detalle que muestra el número de tareas cerradas de cada usuario.

En la siguiente imagen se muestra la estructura del informe de detalle:



Para obtener información acerca de los informes de detalle asociados con los informes de fábrica, consulte ["Informes de fábrica" en la página 354.](#)

Informes de fábrica

Los informes de fábrica recopilan información acerca de los datos principales. Algunos informes de fábrica también tienen informes de detalle. Los informes de detalle contienen otra capa de datos acerca de los datos del informe.

Por ejemplo, el informe Resumen de tareas por estado recopila información acerca del número de tareas por estado. Los informes de detalle asociados con el informe muestran las tareas asignadas por los usuarios para cada estado.

Para usar los informes de fábrica, debe ejecutar el trabajo de actualización de informe asociado con aquel informe que quiera usar. El trabajo de actualización de informe registra el informe y recopila información acerca de los datos principales según los criterios de reportación. A continuación, en la herramienta de aprovisionamiento, puede rellenar los componentes del cuadro con los datos del informe. Si existen informes de detalle asociados con el informe de fábrica, puede vincular el gráfico con un informe de detalle.

En la siguiente tabla, se describen los informes de fábrica y se enumeran los ID de cada informe:

ID de informe	Informe	Descripción
MDM.RPT.1	Resumen de tareas por estado	Muestra el número total de tareas y desglosa las tareas por estado. Puede analizar en detalle los usuarios asignados para cada estado de tarea.
MDM.RPT.2	Resumen de tareas por prioridad	Muestra el número total de tareas y desglosa las tareas por prioridad. Puede analizar en detalle los usuarios asignados para cada prioridad de tarea.
MDM.RPT.4	Entidades de negocio agregadas por año	Muestra el número de registros agregados para cada tipo de entidad de negocio y desglosa los registros por año.
MDM.RPT.5	Tiempo de incorporación de clientes	Muestra cuánto llevó calificar los registros.
MDM.RPT.6	Entidades de negocio por sistemas de origen	Muestra el número total de registros para cada tipo de entidad de negocio que proviene de cada sistema de origen. El informe detallado desglosa los registros para cada par de tipo de entidad de negocio y sistema de origen por año.
MDM.RPT.7	Tareas asignadas por usuarios	Muestra el número total de tareas asignadas por cada usuario.
MDM.RPT.8	Tareas abiertas por funciones de usuario	Muestra el número total de tareas abiertas por funciones de usuario.
MDM.RPT.10	Tareas cerradas por usuarios	Muestra el número total de tareas cerradas por funciones de usuario.

Administrar informes de fábrica

Para usar los informes de fábrica para recopilar datos acerca de los datos principales, ejecute los trabajos de actualización asociados con los informes de fábrica. Después de ejecutar los trabajos de actualización, en la herramienta de aprovisionamiento, puede rellenar los componentes del cuadro con los datos del informe.

Antes de comenzar, revise los informes de fábrica y determine qué informes quiere usar. Para obtener más información, consulte ["Informes de fábrica" en la página 354](#).

1. Para ejecutar el trabajo de actualización asociado con el informe de fábrica, en la barra de direcciones del explorador, introduzca la URL de REST de la API de REST de ejecución del trabajo de actualización de informes.

Por ejemplo, la siguiente URL de REST ejecuta el trabajo de actualización asociado con el informe Resumen de tareas por estado:

```
http://localhost:8080/cmx/report/localhost-orcl-DS_UI1/data/collect/MDM.RPT.1
```

Registró el informe e inició la recopilación de datos según los criterios de informe. La siguiente respuesta de ejemplo muestra el estado de un trabajo de actualización de informe:

```
{
  "status": "PROCESSING"
}
```

2. Para recuperar el estado del trabajo de actualización de informe, en la barra de direcciones del explorador, introduzca la URL de REST desde la API de REST de obtención de estado de trabajo de actualización del informe.

Por ejemplo, la siguiente URL de REST recupera el estado del trabajo de actualización:

```
http://localhost:8080/cmx/report/localhost-orcl-DS_UI1/data/collect/MDM.RPT.1/status
```

La siguiente respuesta de ejemplo muestra el estado de un trabajo de actualización de informe:

```
{
  "status": "COMPLETED_SUCCESSFULLY",
  "jobId": "SVR1.2X9N1",
  "lastUpdateDate": "2019-10-31T14:37:11.846-04:00",
  "startRunDate": "2019-10-31T14:37:09.120-04:00"
}
```

3. Para recuperar los datos y la configuración del informe, en la barra de direcciones del explorador, introduzca la URL de REST desde la API de REST de obtención de configuración y datos del informe.

Por ejemplo, la siguiente URL de REST recupera los datos y la configuración del informe:

```
http://localhost:8080/cmx/report/localhost-orcl-DS_UI1/data/MDM.RPT.1
```

La siguiente respuesta de ejemplo muestra los datos y la configuración del informe:

```
{
  "metadata":{
    "fieldsMetadata":[
      "DIMENSION_VALUE_1",
      "DIMENSION_VALUE_2",
      "TIMEPERIOD_VALUE",
      "METRIC_VALUE",
      "DRILLDOWN_RPT_ID"
    ],
    "ROWID_RPT_CONFIG":"MDM.RPT.1",
    "DIMENSION_NAME 1":"Task Status",
    "METRIC_NAME":"Number of tasks",
    "DIMENSION_NAME 2":"Task Type",
    "TIMEPERIOD_NAME":null,
    "RPT_NAME":"Task Status/Type Report",
    "RPT_DESC":"Metrics for task status/type",
    "RPT_TYPE":null
  },
  "data":[
    [
```

```

    "Overdue",
    "AVOSBeMerge",
    null,
    "36",
    "SVR1.4FCA2    "
  ],
  [
    "Overdue",
    "AVOSBeReviewNoApprove",
    null,
    "189",
    "SVR1.4FC8E    "
  ],
  [
    "Overdue",
    "AVOSBeUpdate",
    null,
    "1",
    "SVR1.4FCAQ    "
  ],
  [
    "Overdue",
    "AVOSBeFinalReview",
    null,
    "1",
    "SVR1.4FC9W    "
  ]
]
...
}

```

4. Para recuperar los informes de detalle y la configuración del informe, use la API de REST de obtención de configuración del informe e informes de detalle.

Por ejemplo, la siguiente URL de REST recupera los Informes de detalle y la configuración del informe:

```
http://localhost:8080/cm/./report/localhost-orcl-DS_UI1/meta/MDM.RPT.1
```

La siguiente respuesta de ejemplo muestra la configuración del informe y los informes de detalle:

```

{
  "ROWID_RPT_CONFIG": "MDM.RPT.1",
  "DIMENSION_NAME 1": "Task Status",
  "METRIC_NAME": "Number of tasks",
  "DIMENSION NAME 2": "Task Type",
  "TIMEPERIOD_NAME": null,
  "RPT_NAME": "Task Status/Type Report",
  "RPT_DESC": "Metrics for task status/type",
  "RPT_TYPE": null,
  "DRILLDOWN": [
    {
      "RPT_NAME": "Task's Owner per Task Status and Task Type",
      "DETAILED_RPT_IDS": [
        "SVR1.4FCA3",
        "SVR1.4FCA4",
        "SVR1.4FCA5",
        "SVR1.4FCA6",
        "SVR1.4FCA7",
        "SVR1.4FC8F",
        "SVR1.4FC8G",
        "SVR1.4FC8H",
        "SVR1.4FC8I",
        "SVR1.4FC8J",
        "SVR1.4FCAR",
        "SVR1.4FCAS",
        "SVR1.4FCAT",
        "SVR1.4FCAU",
        "SVR1.4FCAV",
        "SVR1.4FC9X",
        "SVR1.4FC9Y",
        "SVR1.4FC9Z",
        "SVR1.4FCA0",
      ]
    }
  ]
}

```

```

        "SVR1.4FCA1",
        "SVR1.4FC9L",
        "SVR1.4FC9M",
        "SVR1.4FC9N",
        "SVR1.4FC9O",
        "SVR1.4FC9P",
        "SVR1.4FC8L",
        "SVR1.4FC8M",
        "SVR1.4FC8N",
        "SVR1.4FC8O"
    ],
    "CONFIG_RPT_IDS": [
        "SVR1.4FCA2",
        "SVR1.4FC8E",
        "SVR1.4FCAQ",
        "SVR1.4FC9W",
        "SVR1.4FC9K",
        "SVR1.4FC8K",
        "SVR1.4FCAW",
        "SVR1.4FC9E",
        "SVR1.4FC88",
        "SVR1.4FC8W",
        "SVR1.4FC82",
        "SVR1.4FC92",
        "SVR1.4FC98",
        "SVR1.4FCAE",
        "SVR1.4FCA8",
        "SVR1.4FCB8",
        "SVR1.4FC9Q",
        "SVR1.4FCAK",
        "SVR1.4FC8Q",
        "SVR1.4FCB2"
    ]
}
]
}
}

```

Informes personalizados

Si los informes de fábrica no cumplen con sus requisitos de negocio, puede configurar informes personalizados para recabar otra información acerca de los datos principales. Por ejemplo, es posible que quiera recabar información acerca del número de tareas cerradas por prioridad.

Para usar informes personalizados, debe configurar y registrar los informes, y a continuación, agregar entradas de datos al informe. Para asociar informes de detalle con los datos de informe, configure y registre los informes de detalle. A continuación, puede asociar el informe de detalle con una entrada de datos en el informe personalizado.

Administrar informes personalizados

Registre y configure los informes personalizados que quiera usar para recopilar información acerca de los datos principales. A continuación, en la herramienta de aprovisionamiento, puede rellenar el componente del cuadro con los datos del informe.

Antes de comenzar, determine los datos que quiera recopilar.

1. Para recuperar los datos registrados, use la API de REST de enumeración de informes. Por ejemplo, la siguiente solicitud enumera los informes raíz registrados:

```
GET http://localhost:8080/cmxc/report/localhost-orcl-DS_UI1/list
```

2. Para registrar un informe personalizado, use la API de REST de registro de informe.

Por ejemplo, la siguiente solicitud registra un informe personalizado:

```
POST http://localhost:8080/cmz/report/localhost-orcl-DS_UI1/list
{
  "DIMENSION_NAME_1": "Task Priority",
  "DIMENSION_NAME_2": "Task Type",
  "TIMEPERIOD_NAME": "null",
  "RPT_NAME": "Task Priority/Type Report",
  "RPT_DESC": "Metrics for task status/type",
  "METRIC_NAME": "Number of tasks"
}
```

La siguiente respuesta de ejemplo muestra el informe registrado:

```
{
  "ROWID_RPT_CONFIG": "SVR1.2X9N0",
  "DIMENSION_NAME_1": "Task Priority",
  "DIMENSION_NAME_2": "Task Type",
  "TIMEPERIOD_NAME": "null",
  "RPT_NAME": "Task Priority/Type Report",
  "RPT_DESC": "Metrics for task status/type",
  "METRIC_NAME": "Number of tasks",
  "RPT_TYPE": "null"
}
```

La solicitud devuelve el ID del informe en el parámetro ROWID_RPT_CONFIG.

3. Para agregar entradas de datos en un informe, use la API de REST para agregar o actualizar datos de informe.

Por ejemplo, la siguiente solicitud agrega entradas de datos en un informe:

```
POST http://localhost:8080/cmz/report/localhost-orcl-DS_UI1/data/SVR1.2X9N0
[
  {
    "DIMENSION_VALUE_1": "High",
    "DIMENSION_VALUE_2": "AVOSBeMerge",
    "TIMEPERIOD_VALUE": "null",
    "METRIC_VALUE": "3",
    "DRILLDOWN_RPT_ID": "null"
  },
  {
    "DIMENSION_VALUE_1": "High",
    "DIMENSION_VALUE_2": "AVOSBeReviewNoApprove",
    "TIMEPERIOD_VALUE": "null",
    "METRIC_VALUE": "0",
    "DRILLDOWN_RPT_ID": "null"
  },
  {
    "DIMENSION_VALUE_1": "High",
    "DIMENSION_VALUE_2": "AVOSBeUpdate",
    "TIMEPERIOD_VALUE": "null",
    "METRIC_VALUE": "0",
    "DRILLDOWN_RPT_ID": "null"
  }
  ...
]
```

4. Para recuperar los datos y la configuración del informe, use la API de REST de obtención de detalles y configuración del informe.

Por ejemplo, la siguiente solicitud recupera los datos y la configuración del informe:

```
GET http://localhost:8080/cmz/report/localhost-orcl-DS_UI1/data/SVR1.2X9N0
```

La siguiente respuesta de ejemplo muestra los datos y la configuración del informe:

```
{
  "metadata":{
    "fieldsMetadata":[
      "DIMENSION_VALUE_1",
      "DIMENSION_VALUE_2",

```

```

        "TIMEPERIOD_VALUE",
        "METRIC_VALUE",
        "DRILLDOWN_RPT_ID"
    ],
    "ROWID_RPT_CONFIG": "MDM.RPT.2",
    "DIMENSION_NAME_1": "Task Priority",
    "METRIC_NAME": "Number of tasks",
    "DIMENSION_NAME_2": "Task Type",
    "TIMEPERIOD_NAME": "null",
    "RPT_NAME": "Task Priority/Type Report",
    "RPT_DESC": "Metrics for task status/type",
    "RPT_TYPE": "null"
},
"data": [
    [
        "High",
        "AVOSBeMerge",
        "null",
        "3",
        "SVR1.48P5G"
    ],
    [
        "High",
        "AVOSBeReviewNoApprove",
        "null",
        "0",
        "null"
    ],
    [
        "High",
        "AVOSBeUpdate",
        null,
        "0",
        "null"
    ],
    [
        "High",
        "AVOSBeFinalReview",
        null,
        "0",
        "null"
    ]
    ...
]
}

```

5. Para eliminar un informe, use la API de REST de eliminación de informe. Por ejemplo, la siguiente solicitud elimina un informe:

```
DELETE http://localhost:8080/cmz/report/localhost-orcl-DS_UI1/data/SVR1.2X9N0
```

Administrar informes personalizados con informes de detalle

Registre los informes personalizados que quiera usar y agregue entradas de datos al informe. A continuación, registre los informes de detalle y agregue entradas de datos a los informes de detalle. Cuando agrega o actualiza entradas de datos en los informes raíz, se pueden asociar los informes de detalle con las entradas de datos. A continuación, en la herramienta de aprovisionamiento, puede rellenar el componente del cuadro con los datos del informe y vincular el cuadro raíz con un cuadro de detalle.

Antes de comenzar, determine los datos que quiera recopilar. También, determine los informes de detalle que quiera configurar y asícielo con las entradas de datos en el informe personalizado.

1. Para recuperar los datos registrados, use la API de REST de enumeración de informes. Por ejemplo, la siguiente solicitud enumera los informes raíz e informes de detalle registrados:

```
GET http://localhost:8080/cmz/report/localhost-orcl-DS_UI1/list?show=all
```

2. Para registrar un informe personalizado, use la API de REST de registro de informe.

Por ejemplo, la siguiente solicitud registra un informe personalizado:

```
POST http://localhost:8080/cmx/report/localhost-orcl-DS_UI1/list
{
  "DIMENSION_NAME_1": "Task Priority",
  "DIMENSION_NAME_2": "Task Type",
  "TIMEPERIOD_NAME": "null",
  "RPT_NAME": "Task Priority/Type Report",
  "RPT_DESC": "Metrics for task status/type",
  "METRIC_NAME": "Number of tasks"
}
```

La siguiente respuesta de ejemplo muestra el informe registrado:

```
{
  "ROWID_RPT_CONFIG": "SVR1.2X9N0",
  "DIMENSION_NAME_1": "Task Priority",
  "DIMENSION_NAME_2": "Task Type",
  "TIMEPERIOD_NAME": "null",
  "RPT_NAME": "Task Priority/Type Report",
  "RPT_DESC": "Metrics for task status/type",
  "METRIC_NAME": "Number of tasks",
  "RPT_TYPE": "null"
}
```

La solicitud devuelve el ID del informe en el parámetro ROWID_RPT_CONFIG.

3. Para agregar entradas de datos en un informe raíz, use la API de REST para agregar o actualizar datos de informe.

Más tarde, puede asociar un informe de detalle con una entrada de datos.

Por ejemplo, la siguiente solicitud agrega entradas de datos en un informe:

```
POST http://localhost:8080/cmx/report/localhost-orcl-DS_UI1/data/SVR1.2X9N0
[
  {
    "DIMENSION_VALUE_1": "High",
    "DIMENSION_VALUE_2": "AVOSBeMerge",
    "TIMEPERIOD_VALUE": "null",
    "METRIC_VALUE": "3",
    "DRILLDOWN_RPT_ID": "null"
  },
  {
    "DIMENSION_VALUE_1": "High",
    "DIMENSION_VALUE_2": "AVOSBeReviewNoApprove",
    "TIMEPERIOD_VALUE": "null",
    "METRIC_VALUE": "0",
    "DRILLDOWN_RPT_ID": "null"
  },
  {
    "DIMENSION_VALUE_1": "High",
    "DIMENSION_VALUE_2": "AVOSBeUpdate",
    "TIMEPERIOD_VALUE": "null",
    "METRIC_VALUE": "0",
    "DRILLDOWN_RPT_ID": "null"
  }
  ...
]
```

4. Para registrar un informe de detalle, use la API de REST de registro de informe.

Por ejemplo, la siguiente solicitud registra un informe de detalle:

```
POST http://localhost:8080/cmx/report/localhost-orcl-DS_UI1/list
{
  "DIMENSION_NAME_1": "Task's Owner",
  "DIMENSION_NAME_2": "null",
  "TIMEPERIOD_NAME": "null",
  "RPT_NAME": "Task's Owner per Task Priority and Task Type",
  "RPT_DESC": "Number of tasks for each users for Task Priority/Task Type",
  "METRIC_NAME": "Number of Tasks"
}
```


La siguiente respuesta de ejemplo muestra el informe de detalle registrado:

```
{
  "ROWID_RPT_CONFIG": "SVR1.48P5G",
  "DIMENSION_NAME_1": "Task's Owner",
  "DIMENSION_NAME_2": "null",
  "TIMEPERIOD_NAME": "null",
  "RPT_NAME": "Task's Owner per Task Priority and Task Type",
  "RPT_DESC": "Number of tasks for each users for Task Priority/Task Type",
  "METRIC_NAME": "Number of Tasks",
  "RPT_TYPE": "High/AVOSBeMerge"
}
```

La solicitud devuelve el ID del informe de detalle en el parámetro ROWID_RPT_CONFIG.

5. Para agregar entradas de datos en un informe de detalle, use la API de REST para agregar o actualizar datos de informe.

Por ejemplo, la siguiente solicitud agrega entradas de datos en un informe de detalle:

```
POST http://localhost:8080/cmz/report/localhost-orcl-DS_UI1/data/SVR1.48P5G
[
  {
    "DIMENSION_VALUE_1": "admin",
    "DIMENSION_VALUE_2": "null",
    "TIMEPERIOD_VALUE": "null",
    "METRIC_VALUE": "0",
    "DRILLDOWN_RPT_ID": "null"
  },
  {
    "DIMENSION_VALUE_1": "srmgr2",
    "DIMENSION_VALUE_2": "null",
    "TIMEPERIOD_VALUE": "null",
    "METRIC_VALUE": "0",
    "DRILLDOWN_RPT_ID": "null"
  },
  {
    "DIMENSION_VALUE_1": "mgr2",
    "DIMENSION_VALUE_2": "null",
    "TIMEPERIOD_VALUE": "null",
    "METRIC_VALUE": "0",
    "DRILLDOWN_RPT_ID": "null"
  }
]
```

6. Para asociar un informe de detalle a una entrada de datos en un informe, use la API de REST para agregar o actualizar datos de informe. Especifique el ID del informe de detalle en el parámetro DRILLDOWN_RPT_ID.

Por ejemplo, la siguiente solicitud asocia un informe de detalle con la entrada de datos:

```
POST http://localhost:8080/cmz/report/localhost-orcl-DS_UI1/data/SVR1.2X9N0
[
  {
    "DIMENSION_VALUE_1": "High",
    "DIMENSION_VALUE_2": "AVOSBeMerge",
    "TIMEPERIOD_VALUE": "null",
    "METRIC_VALUE": "5",
    "DRILLDOWN_RPT_ID": "SVR1.48P5G"
  }
]
```

7. Para recuperar los datos y la configuración del informe, use la API de REST de obtención de detalles y configuración del informe.

Por ejemplo, la siguiente solicitud recupera los datos y la configuración del informe:

```
GET http://localhost:8080/cmz/report/localhost-orcl-DS_UI1/data/SVR1.2X9N0
```

La siguiente respuesta de ejemplo muestra los datos y la configuración del informe:

```
{
  "metadata":{
```

```

"fieldsMetadata":[
  "DIMENSION_VALUE_1",
  "DIMENSION_VALUE_2",
  "TIMEPERIOD_VALUE",
  "METRIC_VALUE",
  "DRILLDOWN_RPT_ID"
],
"ROWID_RPT_CONFIG": "SVR1.2X9N0",
"DIMENSION_NAME 1": "Task Priority",
"METRIC_NAME": "Number of tasks",
"DIMENSION NAME 2": "Task Type",
"TIMEPERIOD_NAME": "null",
"RPT_NAME": "Task Priority/Type Report",
"RPT_DESC": "Metrics for task status/type",
"RPT_TYPE": "null"
},
"data":[
  [
    "High",
    "AVOSBeMerge",
    "null",
    "3",
    "SVR1.48P5G"
  ],
  [
    "High",
    "AVOSBeReviewNoApprove",
    "null",
    "0",
    "null"
  ],
  [
    "High",
    "AVOSBeUpdate",
    "null",
    "0",
    "null"
  ],
  [
    "High",
    "AVOSBeFinalReview",
    "null",
    "0",
    "null"
  ]
]
...
]
}

```

8. Para recuperar los informes registrados, incluso los informes personalizados e informes de detalle, use la API de REST de enumeración de informes.

Por ejemplo, la siguiente solicitud enumera los informes raíz e informes de detalle registrados:

```
GET http://localhost:8080/cm/report/localhost-orcl-DS_UI1/list?show=all
```

La siguiente respuesta de ejemplo muestra informes raíz e informes de detalle registrados:

```

[
  {
    "ROWID_RPT_CONFIG": "SVR1.2X9N0",
    "DIMENSION_NAME 1": "Task Priority",
    "METRIC_NAME": "Number of tasks",
    "DIMENSION NAME 2": "Task Type",
    "TIMEPERIOD_NAME": "null",
    "RPT_NAME": "Task Priority/Type Report",
    "RPT_DESC": "Metrics for task status/type",
    "RPT_TYPE": "null"
  },
  {
    "ROWID_RPT_CONFIG": "SVR1.48P5G",

```

```

        "DIMENSION_NAME_1": "Task's Owner",
        "DIMENSION_NAME_2": "null",
        "TIMEPERIOD_NAME": "null",
        "RPT_NAME": "Task's Owner per Task Priority and Task Type",
        "RPT_DESC": "Number of tasks for each users for Task Priority/Task Type",
        "METRIC_NAME": "Number of Tasks",
        "RPT_TYPE": "High/AVOSBeMerge"
    }
}
]

```

9. Para eliminar un informe, use la API de REST de eliminación de informe.
Por ejemplo, la siguiente solicitud elimina un informe:

```
DELETE http://localhost:8080/cm/report/localhost-orcl-DS_UI1/data/SVR1.2X9N0
```

Solución de problemas con API de informes

Si tiene problemas con las API de informes, consulte la información siguiente para encontrar una solución.

No pudo ejecutar trabajos de actualización asociados con los informes de fábrica.

Los servicios de informes tal vez no estén conectados con ActiveVOS. Si editó los archivos EAR de ActiveVOS para personalizar la cadena de búsqueda de JNDI, debe agregar la propiedad `activevos.jndi` al archivo `cmxserver.properties` y especificar la cadena de búsqueda de JNDI personalizada. Para obtener más información, consulte la *Guía de configuración de Multidomain MDM*.

INDICE

A

- actualizar registros
 - cuerpo de la respuesta [64](#)
 - encabezado de respuesta [64](#)
 - parámetros de URL [62](#)
- API de REST
 - acción masiva de tareas [112](#)
 - Actualización de DaaS [259](#)
 - actualizar la configuración del informe [240](#)
 - actualizar metadatos de archivo [121](#)
 - actualizar registros [61](#)
 - actualizar registros coincidentes [221](#)
 - actualizar tarea [97](#)
 - actualizar una relación [173](#)
 - agregar o actualizar detalles de informe [241](#)
 - anular fusión de registros [157](#)
 - asignación masiva de tareas [109](#)
 - buscar registros [69](#)
 - búsqueda de DaaS [249](#)
 - cambios masivos de relación [206](#)
 - crear metadatos de archivo [120](#)
 - crear registros [58](#)
 - crear tarea [95](#)
 - crear una relación [172](#)
 - cuerpo [33](#)
 - cuerpo de la solicitud [34](#)
 - edición masiva de tareas [111](#)
 - ejecutar acción de tarea [103](#)
 - ejecutar el trabajo de actualización de informe [243](#)
 - eliminar informe [242](#)
 - eliminar pendiente [143](#)
 - eliminar registros [65](#)
 - eliminar registros coincidentes [221](#)
 - eliminar una relación [175](#)
 - encabezado [33](#)
 - encabezado de solicitud [33](#)
 - enumerar columnas de coincidencia [50](#)
 - enumerar informes [233](#)
 - enumerar jerarquías [180](#)
 - enumerar metadatos de archivo [119](#)
 - enumerar posibles propietarios de tareas [116](#)
 - enumerar posibles propietarios de una tarea [117](#)
 - enumerar registros [66](#)
 - exportar elementos principales y secundarios directos [199](#)
 - exportar jerarquía [197](#)
 - fusión pendiente [150](#)
 - fusionar registros [155](#)
 - Importación de DaaS [256](#)
 - lectura de DaaS [253](#)
 - leer registros [51](#)
 - leer relación [169](#)
 - leer tarea [93](#)
 - lista de metadatos [45](#)
 - lista de tareas [87](#)
 - lista de usuarios asignables [105](#)

- API de REST (*continuado*)
 - obtener acciones de tarea [114](#)
 - obtener cambios de jerarquía [200](#)
 - obtener configuración del informe e informes de detalle [237](#)
 - obtener configuración y datos del informe [235](#)
 - obtener contenido de archivo [123](#)
 - obtener detalles de evento [231](#)
 - obtener elemento secundario [190](#)
 - obtener elementos principales [188](#)
 - obtener estado de trabajo de actualización de informe [243](#)
 - obtener eventos de historial de registros [228](#)
 - obtener metadatos [42](#)
 - obtener metadatos de archivo [121](#)
 - obtener metadatos de BPM [86](#)
 - obtener metadatos de DaaS [248](#)
 - obtener metadatos de jerarquía [181](#)
 - obtener registros coincidentes [219](#)
 - obtener registros relacionados [175](#)
 - obtener ruta de jerarquía [186](#)
 - promoción masiva [209](#)
 - promover fusión [151](#)
 - promover registro [143](#)
 - proveedor de sugerencias [75](#)
 - rechazo masivo [214](#)
 - reclamación masiva de tareas [106](#)
 - registrar informe [238](#)
 - SearchMatch [81](#)
 - SearchQuery [76](#)
 - suprimir archivo [124](#)
 - tarea completada [101](#)
 - tareas de liberación masiva [108](#)
 - vista previa de fusión [144](#)
 - vista previa de promoción [141](#)
- API SOAP
 - autenticación [276](#)
 - respuesta [279](#)
 - solicitud [279](#)
 - WSDL [278](#)
- autenticación
 - HTTP básico [29](#)
 - método [29](#)
 - uso de cookies [29](#), [276](#)

C

- confianza
 - WriteBE [13](#)
- crear registros
 - cuerpo de la respuesta [60](#)
 - encabezado de respuesta [60](#)
 - parámetros de URL [59](#)
 - URL de solicitud [58](#)
- crear tarea
 - URL de solicitud [95](#)

Cuerpo de REST
Formato JSON [35](#)
formato XML [34](#)

D

datos de vinculación
aplicación personalizada [298](#)
dividir [298](#)

E

ejecutar acción de tarea
cuerpo de la solicitud [103](#)
ejemplo
llamadas externas [319](#)
lógica personalizada [319](#)
eliminar registros
parámetro de URL [66](#)
URL de solicitud [65](#)
eventos compatibles
list [300](#)

F

formato de fecha
acerca de [36](#)

I

inicio de sesión
Herramienta de aprovisionamiento [308](#)

L

leer registros
parámetros de consulta [53](#)
leer tarea
URL de solicitud [93](#)
lista de tareas
parámetros de clasificación [89](#)
parámetros de consulta [88](#)
URL de solicitud [87](#)
llamadas externas
configurar [310](#)
resumen [299](#)

M

métodos REST
compatibles [28](#)
DELETE [28](#)
GET [28](#)
PATCH [28](#)
POST [28](#)
PUT [28](#)

O

obtener detalles de evento
parámetros de consulta [232](#)

obtener eventos de historial de registros
parámetros de consulta [229](#)

P

parámetros de consulta
depth [35](#)
firstRecord [35](#)
recordsToReturn [35](#)
returnTotal [35](#)
searchToken [35](#)
pasos del servicio de entidad de negocio
SearchBE [14](#)
WriteBE [13](#)
periodos efectivos
WriteBE [13](#)
probar llamadas externas
requisitos previos [319](#)
prólogo [10](#)

R

ReadBE
servicio de entidad de negocio [13](#)
registro eliminado
restaurar [13](#)
registros
añadir [328](#)
uso de las API de REST [328](#)
registros raíz
identificación [15](#)
restaurar
registro eliminado parcialmente [13](#)

S

SearchBE
acerca de [14](#)
SearchMatch
exportar resultados [86](#)
SearchQuery
exportar resultados [80](#)
servicio de entidad de negocio
ReadBE [13](#)
Servicio de SOAP
registro [309](#)
servicio de vinculación
configuración [298](#)
servicios de entidad de negocio
Actualización de DaaS [298](#)
extremo de SOAP [15](#)
Extremo EJB [14](#)
Extremo REST [14](#)
extremos [14](#)
extremos de SOAP [275](#)
Extremos REST [27](#)
Importación de DaaS [298](#)
Referencia de la API de REST [41](#)

U

UTC
acerca de [36](#)

V

vinculación empresarial
compatibilidad [297](#)

Z

zona horaria
acerca de [36](#)

W

WriteBE
paso del servicio de entidad de negocio [13](#)