



Informatica® Cloud Application Integration
January 2025

Function Calling using Google Vertex AI

© Copyright Informatica LLC 2025

This software and documentation contain proprietary information of Informatica LLC and are provided under a license agreement containing restrictions on use and disclosure and are also protected by copyright law. Reverse engineering of the software is prohibited. No part of this document may be reproduced or transmitted in any form, by any means (electronic, photocopying, recording or otherwise) without prior consent of Informatica LLC. This Software may be protected by U.S. and/or international Patents and other Patents Pending.

Use, duplication, or disclosure of the Software by the U.S. Government is subject to the restrictions set forth in the applicable software license agreement and as provided in DFARS 227.7202-1(a) and 227.7702-3(a) (1995), DFARS 252.227-7013(1)(ii) (OCT 1988), FAR 12.212(a) (1995), FAR 52.227-19, or FAR 52.227-14 (ALT III), as applicable.

The information in this product or documentation is subject to change without notice. If you find any problems in this product or documentation, please report them to us in writing.

Informatica, Informatica Platform, Informatica Data Services, PowerCenter, PowerCenterRT, PowerCenter Connect, PowerCenter Data Analyzer, PowerExchange, PowerMart, Metadata Manager, Informatica Data Quality, Informatica Data Explorer, Informatica B2B Data Transformation, Informatica B2B Data Exchange Informatica On Demand, Informatica Identity Resolution, Informatica Application Information Lifecycle Management, Informatica Complex Event Processing, Ultra Messaging, Informatica Master Data Management, and Live Data Map are trademarks or registered trademarks of Informatica LLC in the United States and in jurisdictions throughout the world. All other company and product names may be trade names or trademarks of their respective owners.

Portions of this software and/or documentation are subject to copyright held by third parties, including without limitation: Copyright DataDirect Technologies. All rights reserved. Copyright © Sun Microsystems. All rights reserved. Copyright © RSA Security Inc. All Rights Reserved. Copyright © Ordinal Technology Corp. All rights reserved. Copyright © Aandacht c.v. All rights reserved. Copyright Genivia, Inc. All rights reserved. Copyright Isomorphic Software. All rights reserved. Copyright © Meta Integration Technology, Inc. All rights reserved. Copyright © Intalio. All rights reserved. Copyright © Oracle. All rights reserved. Copyright © Adobe Systems Incorporated. All rights reserved. Copyright © DataArt, Inc. All rights reserved. Copyright © ComponentSource. All rights reserved. Copyright © Microsoft Corporation. All rights reserved. Copyright © Rogue Wave Software, Inc. All rights reserved. Copyright © Teradata Corporation. All rights reserved. Copyright © Yahoo! Inc. All rights reserved. Copyright © Glyph & Cog, LLC. All rights reserved. Copyright © Thinkmap, Inc. All rights reserved. Copyright © Clearpace Software Limited. All rights reserved. Copyright © Information Builders, Inc. All rights reserved. Copyright © OSS Nokalva, Inc. All rights reserved. Copyright Edifecs, Inc. All rights reserved. Copyright Cleo Communications, Inc. All rights reserved. Copyright © International Organization for Standardization 1986. All rights reserved. Copyright © ej-technologies GmbH. All rights reserved. Copyright © Jaspersoft Corporation. All rights reserved. Copyright © International Business Machines Corporation. All rights reserved. Copyright © yWorks GmbH. All rights reserved. Copyright © Lucent Technologies. All rights reserved. Copyright © University of Toronto. All rights reserved. Copyright © Daniel Veillard. All rights reserved. Copyright © Unicode, Inc. Copyright IBM Corp. All rights reserved. Copyright © MicroQuill Software Publishing, Inc. All rights reserved. Copyright © PassMark Software Pty Ltd. All rights reserved. Copyright © LogiXML, Inc. All rights reserved. Copyright © 2003-2010 Lorenzi Davide, All rights reserved. Copyright © Red Hat, Inc. All rights reserved. Copyright © The Board of Trustees of the Leland Stanford Junior University. All rights reserved. Copyright © EMC Corporation. All rights reserved. Copyright © Flexera Software. All rights reserved. Copyright © Jinfonet Software. All rights reserved. Copyright © Apple Inc. All rights reserved. Copyright © Telerix Inc. All rights reserved. Copyright © BEA Systems. All rights reserved. Copyright © PDFlib GmbH. All rights reserved. Copyright © Orientation in Objects GmbH. All rights reserved. Copyright © Tanuki Software, Ltd. All rights reserved. Copyright © Ricebridge. All rights reserved. Copyright © Sencha, Inc. All rights reserved. Copyright © Scalable Systems, Inc. All rights reserved. Copyright © jQWidgets. All rights reserved. Copyright © Tableau Software, Inc. All rights reserved. Copyright © MaxMind, Inc. All Rights Reserved. Copyright © TMate Software s.r.o. All rights reserved. Copyright © MapR Technologies Inc. All rights reserved. Copyright © Amazon Corporate LLC. All rights reserved. Copyright © Highsoft. All rights reserved. Copyright © Python Software Foundation. All rights reserved. Copyright © BeOpen.com. All rights reserved. Copyright © CNRI. All rights reserved.

This product includes software developed by the Apache Software Foundation (<http://www.apache.org/>), and/or other software which is licensed under various versions of the Apache License (the "License"). You may obtain a copy of these Licenses at <http://www.apache.org/licenses/>. Unless required by applicable law or agreed to in writing, software distributed under these Licenses is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the Licenses for the specific language governing permissions and limitations under the Licenses.

This product includes software which was developed by Mozilla (<http://www.mozilla.org/>), software copyright The JBoss Group, LLC, all rights reserved; software copyright © 1999-2006 by Bruno Lowagie and Paulo Soares and other software which is licensed under various versions of the GNU Lesser General Public License Agreement, which may be found at <http://www.gnu.org/licenses/lgpl.html>. The materials are provided free of charge by Informatica, "as-is", without warranty of any kind, either express or implied, including but not limited to the implied warranties of merchantability and fitness for a particular purpose.

The product includes ACE(TM) and TAO(TM) software copyrighted by Douglas C. Schmidt and his research group at Washington University, University of California, Irvine, and Vanderbilt University, Copyright (©) 1993-2006, all rights reserved.

This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit (copyright The OpenSSL Project. All Rights Reserved) and redistribution of this software is subject to terms available at <http://www.openssl.org> and <http://www.openssl.org/source/license.html>.

This product includes Curl software which is Copyright 1996-2013, Daniel Stenberg, <daniel@haxx.se>. All Rights Reserved. Permissions and limitations regarding this software are subject to terms available at <http://curl.haxx.se/docs/copyright.html>. Permission to use, copy, modify, and distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

The product includes software copyright 2001-2005 (©) MetaStuff, Ltd. All Rights Reserved. Permissions and limitations regarding this software are subject to terms available at <http://www.dom4j.org/license.html>.

The product includes software copyright © 2004-2007, The Dojo Foundation. All Rights Reserved. Permissions and limitations regarding this software are subject to terms available at <http://dojotoolkit.org/license>.

This product includes ICU software which is copyright International Business Machines Corporation and others. All rights reserved. Permissions and limitations regarding this software are subject to terms available at <http://source.icu-project.org/repos/icu/icu/trunk/license.html>.

This product includes software copyright © 1996-2006 Per Bothner. All rights reserved. Your right to use such materials is set forth in the license which may be found at <http://www.gnu.org/software/kawa/Software-License.html>.

This product includes OSSP UUID software which is Copyright © 2002 Ralf S. Engelschall, Copyright © 2002 The OSSP Project Copyright © 2002 Cable & Wireless Deutschland. Permissions and limitations regarding this software are subject to terms available at <http://www.opensource.org/licenses/mit-license.php>.

This product includes software developed by Boost (<http://www.boost.org/>) or under the Boost software license. Permissions and limitations regarding this software are subject to terms available at http://www.boost.org/LICENSE_1_0.txt.

This product includes software copyright © 1997-2007 University of Cambridge. Permissions and limitations regarding this software are subject to terms available at <http://www.pcre.org/license.txt>.

This product includes software copyright © 2007 The Eclipse Foundation. All Rights Reserved. Permissions and limitations regarding this software are subject to terms available at <http://www.eclipse.org/org/documents/epl-v10.php> and at <http://www.eclipse.org/org/documents/edl-v10.php>.

This product includes software licensed under the terms at <http://www.tcl.tk/software/tcltk/license.html>, <http://www.bosrup.com/web/overlib/?License>, <http://www.stlport.org/doc/license.html>, <http://asm.ow2.org/license.html>, <http://www.cryptix.org/LICENSE.TXT>, <http://hsqldb.org/web/hsqldbLicense.html>, <http://httpunit.sourceforge.net/doc/license.html>, <http://jung.sourceforge.net/license.txt>, http://www.gzip.org/zlib/zlib_license.html, <http://www.openldap.org/software/release/license.html>, <http://www.libssh2.org>, <http://slf4j.org/license.html>, <http://www.sente.ch/software/OpenSourceLicense.html>, <http://fusesource.com/downloads/license-agreements/fuse-message-broker-v-5-3-license-agreement>; <http://antlr.org/license.html>; <http://aopalliance.sourceforge.net/>; <http://www.bouncycastle.org/licence.html>; <http://www.jgraph.com/jgraphdownload.html>; <http://www.jcraft.com/jsch/LICENSE.txt>; http://jotm.objectweb.org/bsd_license.html; <http://www.w3.org/Consortium/Legal/2002/copyright-software-20021231>; <http://www.slf4j.org/license.html>; <http://nanoxml.sourceforge.net/orig/copyright.html>; <http://www.json.org/license.html>; <http://forge.ow2.org/projects/javaservice/>; <http://www.postgresql.org/about/license.html>, <http://www.sqlite.org/copyright.html>, <http://www.tcl.tk/software/tcltk/license.html>, <http://www.jaxen.org/faq.html>, <http://www.jdom.org/docs/faq.html>, <http://www.slf4j.org/license.html>; <http://www.iodbc.org/dataspace/iodbc/wiki/IODBC/License>; <http://www.keplerproject.org/md5/license.html>; <http://www.toedter.com/en/jcalendar/license.html>; <http://www.edankert.com/bounce/index.html>; <http://www.net-snmp.org/about/license.html>; <http://www.openmdx.org/#FAQ>; http://www.php.net/license/3_01.txt; <http://srp.stanford.edu/license.txt>; <http://www.schneier.com/blowfish.html>; <http://www.jmock.org/license.html>; <http://xsom.java.net>; <http://benalman.com/about/license/>; <https://github.com/CreateJS/EaselJS/blob/master/src/easeljs/display/Bitmap.js>; <http://www.h2database.com/html/license.html#summary>; <http://jsoncpp.sourceforge.net/LICENSE>; <http://jdbc.postgresql.org/license.html>; <http://protobuf.googlecode.com/svn/trunk/src/google/protobuf/descriptor.proto>; <https://github.com/rantav/hector/blob/master/LICENSE>; <http://web.mit.edu/Kerberos/krb5-current/doc/mitK5license.html>; <http://jibx.sourceforge.net/jibx-license.html>; <https://github.com/lyokato/libgeohash/blob/master/LICENSE>; <https://github.com/hjiang/jsonxx/blob/master/LICENSE>; <https://code.google.com/p/lz4/>; <https://github.com/jedisct1/libsodium/blob/master/LICENSE>; <http://one-jar.sourceforge.net/index.php?page=documents&file=license>; <https://github.com/EsotericSoftware/kryo/blob/master/license.txt>; <http://www.scala-lang.org/license.html>; <https://github.com/tinkerpop/blueprints/blob/master/LICENSE.txt>; <http://gee.cs.oswego.edu/dl/classes/EDU/oswego/cs/dl/util/concurrent/intro.html>; <https://aws.amazon.com/asl/>; <https://github.com/twbs/bootstrap/blob/master/LICENSE>; <https://sourceforge.net/p/xmlunit/code/HEAD/tree/trunk/LICENSE.txt>; <https://github.com/documentcloud/underscore-contrib/blob/master/LICENSE>, and <https://github.com/apache/hbase/blob/master/LICENSE.txt>.

This product includes software licensed under the Academic Free License (<http://www.opensource.org/licenses/afl-3.0.php>), the Common Development and Distribution License (<http://www.opensource.org/licenses/cddl1.php>), the Common Public License (<http://www.opensource.org/licenses/cpl1.0.php>), the Sun Binary Code License Agreement Supplemental License Terms, the BSD License (<http://www.opensource.org/licenses/bsd-license.php>), the new BSD License (<http://opensource.org/licenses/BSD-3-Clause>), the MIT License (<http://www.opensource.org/licenses/mit-license.php>), the Artistic License (<http://www.opensource.org/licenses/artistic-license-1.0>) and the Initial Developer's Public License Version 1.0 (<http://www.firebirdsql.org/en/initial-developer-s-public-license-version-1-0/>).

This product includes software copyright © 2003-2006 Joe Walnes, 2006-2007 XStream Committers. All rights reserved. Permissions and limitations regarding this software are subject to terms available at <http://xstream.codehaus.org/license.html>. This product includes software developed by the Indiana University Extreme! Lab. For further information please visit <http://www.extreme.indiana.edu/>.

This product includes software Copyright (c) 2013 Frank Balluffi and Markus Moeller. All rights reserved. Permissions and limitations regarding this software are subject to terms of the MIT license.

See patents at <https://www.informatica.com/legal/patents.html>.

DISCLAIMER: Informatica LLC provides this documentation "as is" without warranty of any kind, either express or implied, including, but not limited to, the implied warranties of noninfringement, merchantability, or use for a particular purpose. Informatica LLC does not warrant that this software or documentation is error free. The information provided in this software or documentation may include technical inaccuracies or typographical errors. The information in this software and documentation is subject to change at any time without notice.

NOTICES

This Informatica product (the "Software") includes certain drivers (the "DataDirect Drivers") from DataDirect Technologies, an operating company of Progress Software Corporation ("DataDirect") which are subject to the following terms and conditions:

1. THE DATADIRECT DRIVERS ARE PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NON-INFRINGEMENT.
2. IN NO EVENT WILL DATADIRECT OR ITS THIRD PARTY SUPPLIERS BE LIABLE TO THE END-USER CUSTOMER FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, CONSEQUENTIAL OR OTHER DAMAGES ARISING OUT OF THE USE OF THE ODBC DRIVERS, WHETHER OR NOT INFORMED OF THE POSSIBILITIES OF DAMAGES IN ADVANCE. THESE LIMITATIONS APPLY TO ALL CAUSES OF ACTION, INCLUDING, WITHOUT LIMITATION, BREACH OF CONTRACT, BREACH OF WARRANTY, NEGLIGENCE, STRICT LIABILITY, MISREPRESENTATION AND OTHER TORTS.

Publication Date: 2025-01-08

Table of Contents

- Preface 5**

- Chapter 1: Introduction to Function Calling using Google Vertex AI recipe..... 6**
 - Function Calling using Google Vertex AI recipe contents. 6
 - Function Calling using Google Vertex AI recipe assets. 7

- Chapter 2: Using the Function Calling using Google Vertex AI recipe..... 8**
 - Step 1. Copy and access the recipe. 8
 - Step 2. Publish the Generic Connector service connector. 9
 - Step 3. Publish the GenericConnection connection. 9
 - Step 4. Configure and publish the Google Vertex AI connection. 9
 - Step 5. Configure and publish the processes. 9
 - Step 6. Publish and run the guide. 12
 - Test Function Calling using Google Vertex AI. 13

Preface

Use *Function Calling using Google Vertex AI* to learn how to create the payload for Application Integration processes and execute them dynamically based on a user query. This guide assumes that you have an understanding of the Google Vertex AI Connector concepts.

CHAPTER 1

Introduction to Function Calling using Google Vertex AI recipe

The Function Calling using Google Vertex AI recipe is based on REST and SOAP APIs. The recipe allows you to create the payload for Application Integration processes and execute them dynamically based on a user query.

The recipe allows you to specify the user prompt and enter the function declaration in the JSON format to query other processes based on a function call response using a guide.

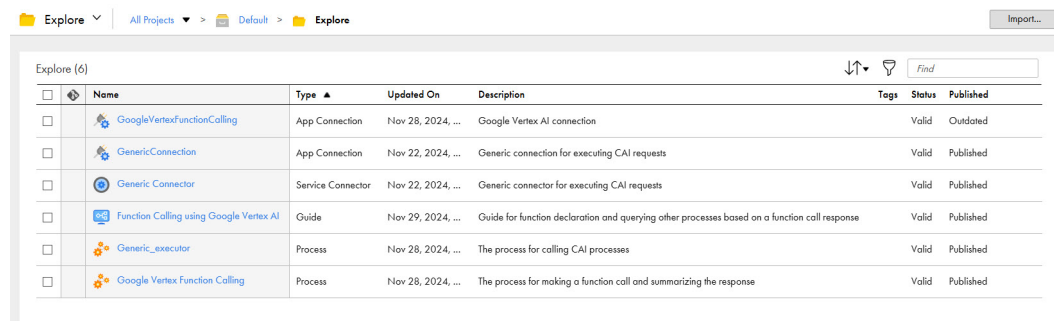
The process makes a request to Google Vertex AI with the specified function declaration and user prompt. Google Vertex AI recognizes the query, extracts key variables and values, and returns them in the appropriate format. Based on the received response, a function call is generated with function arguments and a link to the process execution.

The user confirms the accuracy of the function call response to make the call to Application Integration process or refuses to make function call again. The process requests the Application Integration recipe based on the function-calling response. The user then receives a message about the success of the recipe call and a response from the Application Integration recipe.

Function Calling using Google Vertex AI recipe contents

The Function Calling using Google Vertex AI recipe contains a service connector, app connections, a guide, and processes.

The following image shows the assets that the Function Calling using Google Vertex AI recipe package contains:



The screenshot shows the Google Cloud console interface with the 'Explore' view. The table below lists the assets contained within the 'Function Calling using Google Vertex AI' recipe package.

Name	Type	Updated On	Description	Tags	Status	Published
GoogleVertexFunctionCalling	App Connection	Nov 28, 2024, ...	Google Vertex AI connection		Valid	Outdated
GenericConnection	App Connection	Nov 22, 2024, ...	Generic connection for executing CAI requests		Valid	Published
Generic Connector	Service Connector	Nov 22, 2024, ...	Generic connector for executing CAI requests		Valid	Published
Function Calling using Google Vertex AI	Guide	Nov 29, 2024, ...	Guide for function declaration and querying other processes based on a function call response		Valid	Published
Generic_executor	Process	Nov 28, 2024, ...	The process for calling CAI processes		Valid	Published
Google Vertex Function Calling	Process	Nov 28, 2024, ...	The process for making a function call and summarizing the response		Valid	Published

Function Calling using Google Vertex AI recipe assets

The following table lists the assets that the Function Calling using Google Vertex AI recipe package contains:

Asset Name	Asset Type	Description
Generic Connector	Service connector	Generic connector for executing an Application Integration request.
GoogleVertexFunctionCalling	App connection	Google Vertex AI connection.
GenericConnection	App connection	Generic connection for executing an Application Integration request.
Generic_executor	Process	Process to call another Application Integration processes.
Google Vertex Function Calling	Process	Process for making a function call and summarizing the response.
Function calling using Google Vertex AI	Guide	Guide for declaring a function and querying other processes based on a function call response.

CHAPTER 2

Using the Function Calling using Google Vertex AI recipe

To use the Function Calling using Google Vertex AI recipe, you must perform the following steps manually:

1. Copy and access the recipe.
2. Publish the Generic Connector connector.
3. Publish the GenericConnection connection.
4. Configure and publish the Google Vertex AI connection.
5. Configure and publish the processes.
6. Publish and run the guide.

Step 1. Copy and access the recipe

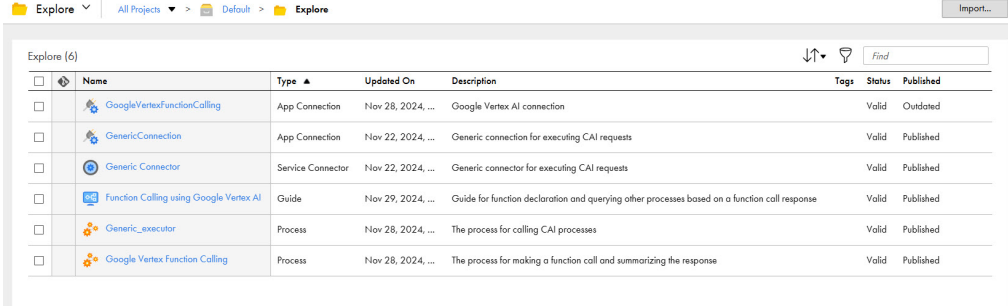
Copy the pre-configured assets in the recipe to a separate project or folder.

1. Open the **Function Calling using Google Vertex AI** recipe and click **Use**.
2. Select the location where you want to copy the recipe, and then click **Continue**.
3. In the **Copying the recipe** dialog box, click **OK**.

It might take some time for the recipe to get copied. You will receive a notification when the recipe is ready for use.

4. After the recipe is copied, click **Explore** to access the recipe content.
5. Navigate to the project or folder where you copied the recipe or enter the recipe name in the **Find** box.

All the assets in the recipe are displayed as shown in the following image:



The screenshot shows the Google Cloud console interface with the 'Explore' view. The breadcrumb navigation is 'Explore > All Projects > Default > Explore'. The main content area displays a table of assets for the recipe 'Function Calling using Google Vertex AI'. The table has columns for Name, Type, Updated On, Description, Tags, Status, and Published. The assets listed are:

Name	Type	Updated On	Description	Tags	Status	Published
GoogleVertexFunctionCalling	App Connection	Nov 28, 2024, ...	Google Vertex AI connection		Valid	Outdated
GenericConnection	App Connection	Nov 22, 2024, ...	Generic connection for executing CAI requests		Valid	Published
Generic Connector	Service Connector	Nov 22, 2024, ...	Generic connector for executing CAI requests		Valid	Published
Function Calling using Google Vertex AI	Guide	Nov 29, 2024, ...	Guide for function declaration and querying other processes based on a function call response		Valid	Published
Generic_executor	Process	Nov 28, 2024, ...	The process for calling CAI processes		Valid	Published
Google Vertex Function Calling	Process	Nov 28, 2024, ...	The process for making a function call and summarizing the response		Valid	Published

Step 2. Publish the Generic Connector service connector

To publish the **Generic Connector** service connector, open the **Generic Connector** service connector and click **Publish**.

Step 3. Publish the GenericConnection connection

To publish the **GenericConnection** connection, open the **GenericConnection** connection and click **Publish**.

Step 4. Configure and publish the Google Vertex AI connection

Configure the project ID, private key ID, private key, client email, and location to connect to Google Vertex AI, and then publish the GoogleVertexFunctionCalling connection.

1. Open the **GoogleVertexFunctionCalling** connection.
2. In the **Connection Properties** section, enter values for the following properties:

Property	Description
Project_ID	The project ID to generate a valid access token.
Location	The region to process the request.
Private_Key_ID	The private key ID associated with the Google Vertex AI service account.
Client_Email	The email address associated with the Google Vertex AI service account.
Private_Key	The Google Vertex AI private key associated with the service account. Enter the PKCS1 certificate as a Base64-encoded string in the following format: -----BEGIN PRIVATE KEY----- n-----END PRIVATE KEY-----

3. Save and publish the connection.

Step 5. Configure and publish the processes

Configure the Google Vertex LLM model version and service URL, and publish the processes.

1. To publish the **Generic_executor** process, click **Actions** in the row that contains the process and select **publish**.
2. Open the **Google Vertex Function Calling** process.

- On the **Assignments** tab of the **Set Flow Configuration** step, enter values in the following fields:
 - In the **LLM_Model** field, enter the model ID of the model that you want to use. The model ID is set to gemini-1.5-pro-002, by default.
 - In the **Generation_Config** field, enter the instructions using the Expression Editor, as shown in the following sample code:

```
<generationConfig>
  <temperature>0.2</temperature>
  <maxOutputTokens>1024</maxOutputTokens>
  <topP>0.8</topP>
</generationConfig>
```

For the **Generation_Config** field, enter values for the following properties:

Property	Description
temperature	Controls the randomness of the model's output. A lower value makes the output more deterministic, while a higher value increases randomness and creativity. For example, a temperature of 0.5 balances between deterministic and creative outputs.
maxOutputTokens	Defines the maximum number of tokens the model can generate in its response. Setting a limit ensures that the response is concise and fits within the desired length constraints.
topP	Determines the cumulative probability threshold for token selection. The model considers the smallest set of tokens whose cumulative probability meets or exceeds topP. For example, if topP is set to 0.1, the model considers only the top 10% most probable tokens at each step.

To add additional generation config parameters, see the [Generative AI on Vertex AI](#) documentation.

- On the **Assignments** tab of the **Prepare Default Function Declaration** step, enter the process URL of the recipe in the **Function_Declaration** field using the Expression Editor, as shown in the following sample code:

```
<tools>
  <functionDeclarations>
    <name>Update_Amazon_Bedrock_Knowledge_Base_executor</name>
    <description>This function processes a user-provided prompt to identify and extract key information based on predefined parameters. For instance, it can detect specific identifiers such as Knowledge_Base_ID or Email_Address by scanning the text for patterns or matches associated with these keywords. The function returns the extracted data for further processing or validation, facilitating automated workflows where structured information is derived from unstructured input.</description>
    <parameters>
      <type>object</type>
      <properties>
        <Process_URL>
          <type>string</type>
          <description>Fixed URL to execute process</description>
          <enum><Service_URL_of_the_Application_Integration_Process></enum>
        </Process_URL>
        <Payload>
          <type>array</type>
          <description>Structured payload for processes</description>
          <items>
            <description>Fields in payload for processes</description>
            <type>object</type>
            <properties>
              <Bucket_Name>
                <type>string</type>
                <description>The bucket name in AWS S3 is a unique
```

```

identifier for an S3 bucket.</description>
    </Bucket_Name>
    <Knowledge_Base_ID>
      <type>string</type>
      <description>The unique identifier of the knowledge
base. Sometimes, the knowledge base is abbreviated as KB.</description>
    </Knowledge_Base_ID>
    <Email_Address>
      <type>string</type>
      <description>Email address where the process result
will be sent.</description>
    </Email_Address>
    <Data_Source_ID>
      <type>string</type>
      <description>The unique identifier of the data
source. Sometimes, the data source is abbreviated as DS.</description>
    </Data_Source_ID>
  </properties>
  <required>Knowledge_Base_ID</required>
  <required>Email_Address</required>
</items>
</Payload>
</properties>
<required>Process_URL</required>
<required>Payload</required>
</parameters>
</functionDeclarations>
<functionDeclarations>
  <name>Synchronize_ServiceNow_Incidents_with_Jira_Issues_executor</name>
  <description>This function processes a user-provided prompt to identify and
extract key information based on predefined parameters. For instance, it can detect
specific identifiers such as Jira_Issue_Type_ID or Jira_Project_ID by scanning the
text for patterns or matches associated with these keywords. The function returns
the extracted data for further processing or validation, facilitating automated
workflows where structured information is derived from unstructured input.</
description>
  <parameters>
    <type>object</type>
    <properties>
      <Process_URL>
        <type>string</type>
        <description>Fixed URL to execute process</description>
        <enum><b>Service_URL_of_the_Application_Integration_Process</b></enum>
      </Process_URL>
      <Payload>
        <type>array</type>
        <description>Structured payload for processes</description>
        <items>
          <description>Fields in payload for processes</description>
          <type>object</type>
          <properties>
            <Date_For_Search>
              <type>string</type>
              <description>Search date in the YYYY-MM-DD format.</
description>
            </Date_For_Search>
            <Filter>
              <type>string</type>
              <description>Name of the category to be filtered</
description>
            </Filter>
            <Email>
              <type>string</type>
              <description>Email address where the process result
will be sent.</description>
            </Email>
            <Jira_Issue_Type_ID>
              <type>string</type>
              <description>The unique identifier of Jira issue

```

```

type</description>
                                </Jira_Issue_Type_ID>
                                <Jira_Project_ID>
                                  <type>string</type>
                                  <description>The unique identifier of Jira project</
description>
                                </Jira_Project_ID>
                                <Sync_Assignee>
                                  <type>boolean</type>
                                  <description>Boolean variable whether to synchronize
the Assignee</description>
                                </Sync_Assignee>
                                </properties>
                                <required>Email</required>
                                <required>Jira_Issue_Type_ID</required>
                                <required>Jira_Project_ID</required>
                                </items>
                                </Payload>
                                </properties>
                                <required>Process_URL</required>
                                <required>Payload</required>
                                </parameters>
                                </functionDeclarations>
                                </tools>

```

This example applies to and can execute the **Synchronize_ServiceNow_Incidents_with_Jira_Issue** and **Update_Amazon_Bedrock_Knowledge_Base** recipes.

5. Save and publish the process.

Step 6. Publish and run the guide

You can declare functions and query other processes based on a function call response.

1. Open the **Function Calling using Google Vertex AI** guide.
2. On the **Start** tab of the Start step, ensure that the **Run As** field is set to **Current User**.
3. Save and publish the guide.
4. On the **Actions** menu, click **Run**. Alternatively, you can copy the execution URL from the **Properties Details** dialog box to run the guide.
5. On the **Instructions** page, enter the following values:
 - In the **Enter a prompt below** field, enter your question based on the function declaration.
 - Optionally, in the **Function declaration** field, enter the custom function declaration. By default, the function declaration value is set for the Synchronize ServiceNow Incidents with Jira Issues recipe and Update Amazon Bedrock Knowledge Base recipe in the **Prepare Default Function Declaration** step of the **Google Vertex Function Calling** process.
 - Select the function calling confirmation as **No** if you don't want to receive a summary of function call response and omit confirmaton step before the function is executed. By default, the function calling confirmation is selected as **Yes**.
6. Click **Continue**.
You will receive a summary of the function call request in response if the function call is generated. Otherwise, an error message appears.
7. Click **Confirm** to confirm the summary of the function call response and make a call to the CAI process or click **Try again**.
You will receive a response that contains the process call status, the process execution status, and the process response.

Note: If the process you are calling returns any response, you will see the response on the Answer screen. However, if the process runs for more than 59 seconds, you will see a message about the successful process call, but the result can be viewed in Application Integration Console.

You can also use the embed code to embed the guide into an HTML document of a third-party application.

Test Function Calling using Google Vertex AI

You can test the Function Calling using Google Vertex AI recipe with the function declaration value that is set as default in the **Prepare Default Function Declaration** step of the **Google Vertex Function Calling** process.

The default value applies to and can execute the Synchronize ServiceNow Incidents with Jira Issues recipe and the Update Amazon Bedrock Knowledge Base recipe.

For example, in the **Function Calling using Google Vertex AI** guide, you can enter one of the following prompts for the correct function call for these recipes:

- *Please make an ingestion job for KB ON1FS86KFX with DS REQJVGUX8S and send the result to example@gmail.com*
- *Synchronize my incidents in the Software category for November 19, 2024, project id 1001, Jira issue type id 1001, you also need to synchronize the Assignee and send the result to example@gmail.com*

The function name must start with a letter or an underscore, and can contain alphanumeric characters, underscores (_), periods (.), and hyphens (-). For more information, see [Google Vertex AI](#) documentation.

The structure of the **Function_Call_Answer** is important for further execution of the request to the Application Integration process. To set your **Function_Declaration** in the guide, you must specify it in the JSON format as shown in the following example:

```
{
  "tools": [{
    "functionDeclarations": [{
      "name": "Update_Amazon_Bedrock_Knowledge_Base_executor",
      "description": "This function processes a user-provided prompt to identify and extract key information based on predefined parameters. For instance, it can detect specific identifiers such as Knowledge_Base_ID or Email_Address by scanning the text for patterns or matches associated with these keywords. The function returns the extracted data for further processing or validation, facilitating automated workflows where structured information is derived from unstructured input.",
      "parameters": {
        "type": "object",
        "properties": {
          "Process_URL": {
            "type": "string",
            "description": "Fixed URL to execution process",
            "enum": [
              "<Service_URL_of_the_Application_Integration_Process>"
            ]
          },
          "Payload": {
            "type": "array",
            "description": "Structured payload for processes",
            "items": {
              "description": "Fields in payload for processes",
              "type": "object",
              "properties": {
                "Bucket_Name": {
                  "type": "string",
                  "description": "The bucket name in AWS S3 is a unique identifier for an S3 bucket."
                }
              }
            }
          }
        }
      }
    ]
  }
}
```

```

        "Knowledge_Base_ID": {
            "type": "string",
            "description": "The unique identifier of the
knowledge base. Sometimes, the knowledge base is abbreviated as KB."
        },
        "Email_Address": {
            "type": "string",
            "description": "Email address where the
process result will be sent."
        },
        "Data_Source_ID": {
            "type": "string",
            "description": "The unique identifier of the
data source. Sometimes, the data source is abbreviated as DS."
        }
    },
    "required": ["Knowledge_Base_ID", "Email_Address"]
}
},
"required": ["Process_URL", "Payload"]
}, {
    "name": "Synchronize_ServiceNow_Incidents_with_Jira_Issues_executor",
    "description": "This function processes a user-provided prompt to
identify and extract key information based on predefined parameters. For instance, it
can detect specific identifiers such as Jira_Issue_Type_ID or Jira_Project_ID by
scanning the text for patterns or matches associated with these keywords. The function
returns the extracted data for further processing or validation, facilitating automated
workflows where structured information is derived from unstructured input.",
    "parameters": {
        "type": "object",
        "properties": {
            "Process_URL": {
                "type": "string",
                "description": "Fixed URL to execution process",
                "enum":
["<Service_URL_of_the_Application_Integration_Process>"]
            },
            "Payload": {
                "type": "array",
                "description": "Structured payload for processes",
                "items": {
                    "description": "Fields in payload for processes",
                    "type": "object",
                    "properties": {
                        "Date_For_Search": {
                            "type": "string",
                            "description": "Search date in the YYYY-MM-
DD format."
                        },
                        "Filter": {
                            "type": "string",
                            "description": "Name of the category to be
filtered"
                        },
                        "Email": {
                            "type": "string",
                            "description": "Email address where the
process result will be sent."
                        },
                        "Jira_Issue_Type_ID": {
                            "type": "string",
                            "description": "The unique identifier of
Jira issue type"
                        },
                        "Jira_Project_ID": {
                            "type": "string",
                            "description": "The unique identifier of
Jira project"
                        }
                    }
                }
            }
        }
    }
},

```

```

        "Sync_Assignee": {
            "type": "boolean",
            "description": "Boolean variable whether to
synchronize the Assignee"
        },
        "required": ["Email", "Jira_Issue_Type_ID",
"Jira_Project_ID"]
    },
    "required": ["Process_URL", "Payload"]
}
]
}
]
}

```

For testing, you can construct another structure of a Function Declaration as required.