Informatica® Cloud Application Integration
July 2024

GeminiAI Chat with File

Informatica Cloud Application Integration GeminiAI Chat with File
July 2024

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/hsqlLicense.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/licence.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: 2024-12-06

Table of Contents

Preface

Use *GeminiAI Chat with File* to learn how to ask questions to the Gemini Large Language Model (LLM) based on a specified file's contents. The recipe is based on REST and SOAP APIs and you use an HTTP request to call the process.

CHAPTER 1

Introduction to GeminiAI Chat with File recipe

The GeminiAI Chat with File recipe is based on REST and SOAP APIs. You can ask questions to the Gemini Large Language Model (LLM) based on a specified file's contents.

The process allows you to provide a file path and a user prompt. The process reads the text from the file and answers the user's questions based on the contents of the file using the Gemini Large Language Model (LLM).

You can use files of the following formats:

`.txt, .xml, .json, .html, .csv,` and `.md`

CHAPTER 2

GeminiAI Chat with File recipe contents

The GeminiAI Chat with File recipe contains a process object, an app connection, and two processes.

The following image shows the assets that the GeminiAI Chat with File recipe package contains:
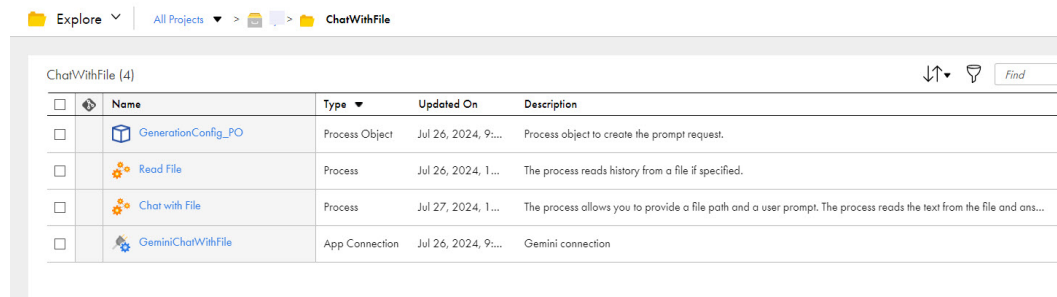
GeminiAI Chat with File recipe assets

The following table lists the assets that the GeminiAI Chat with File recipe package contains:

Asset Name	Asset Type	Description
GenerationConfig_PO	Process object	Creates the prompt request.
GeminiChatWithFile	App connection	Gemini connection.
Read File	Process	Subprocess that reads text from a file if specified.
Chat with File	Process	Allows you to provide a file path and a user prompt. The process reads the text from the file and answers the user's questions based on the text using the Gemini Large Language Model (LLM).

CHAPTER 3

Using the GeminiAI Chat with File recipe

To use the GeminiAI Chat with File recipe, you must perform the following steps manually:

Step 1: Copy and access the recipe

Step 2: Configure and publish the GeminiChatWithFile connection

Step 3: Configure and publish the processes

Step 4: Invoke the process

Copying and accessing the recipe
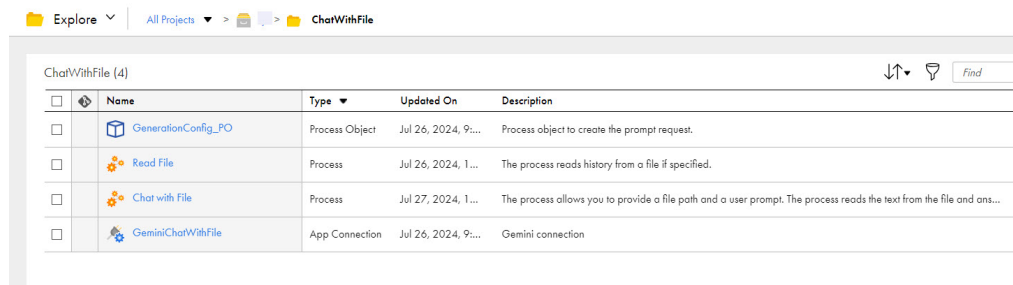
To copy and access the recipe content, perform the following steps:

1. Open the **GeminiAI Chat with File** 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:

Configuring and publishing the GeminiChatWithFile connection

To configure and publish the GeminiChatWithFile connection, perform the following steps:

1. Open the **GeminiChatWithFile** connection.

2. In the **Type** field, select **Gemini**.

3. In the **Run On** field, select **Cloud Server or any Secure Agent**.

4. In the **Connection Properties** section, enter the API key in the **API_Key** property. The **API_Key** property authenticates Gemini connection requests.

5. Save, test, and publish the connection.

Configuring and publishing the processes

1. Open the **Read File** process.

2. On the **Start** tab of the Start step, select the Secure Agent from the **Run On** list.

3. Optionally, you can change the tracing level from **Verbose** to **None** on the **Advanced** tab.

4. Save and publish the process.

5. Open the **Chat with File** process.

6. On the **Start** tab of the Start step, select **Cloud Server** from the **Run On** list.

7. On the **Temp Fields** tab of the Start step, the **Model_LLM** field is set to **gemini-1.5 pro** by default. You can optionally edit the model version. For information about changing the model version, see the Gemini documentation.

8. Optionally, you can change the tracing level from **Verbose** to **None** on the **Advanced** tab.

9. Optionally, in the **Prepare Request** step, enter the prompt instructions in the **Assignments** field by updating the **Prompt_Configuration** and **Request** fields using the Expression Editor, as shown in the following sample code:

```
For Prompt_Configuration:
 <generationConfig>
      <stopSequences>.</stopSequences>
      <candidateCount>1</candidateCount>
      <maxOutputTokens>500</maxOutputTokens>
      <temperature>0.1</temperature>
      <topP>0.1</topP>
      <topK>2</topK>
   </generationConfig>

For Request:
<Generate_Content_Request>
   <contents>
      <parts>
         <text>This is the text from the file: {$temp.Content_From_File}
      </parts>
      <role>user</role>
   </contents>
      <parts>
         <text>{$input.User_Prompt}</text>
      </parts>
      <role>user</role>
```

```
            </contents>
    <generationConfig>
        <stopSequences>{$temp.Prompt_Configuration[1]/stopSequences}</stopSequences>
        <candidateCount>{$temp.Prompt_Configuration[1]/candidateCount }</
candidateCount>
        <maxOutputTokens>{$temp.Prompt_Configuration[1]/maxOutputTokens }</
maxOutputTokens>
        <temperature>{$temp.Prompt_Configuration[1]/temperature }</temperature>
        <topP>{$temp.Prompt_Configuration[1]/topP }</topP>
        <topK>{$temp.Prompt_Configuration[1]/topK }</topK>
    </generationConfig>
</Generate_Content_Request>
```

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

Property	Description
stopSequences	Contains sequences of characters or strings that stop the model's output. This property controls where the model must end its response.
candidateCount	Specifies the number of response candidates that the model must generate. For example, if the value is set to 1, the model generates one response. If set to a higher number, the model generates that many alternative responses for the same input.
maxOutputTokens	Defines the maximum number of tokens that the model can generate in its response. Setting a limit ensures that the response is concise and fits within the desired length constraints.
temperature	Controls the randomness of the model's output. A lower value close to 0 makes the output more deterministic, while a higher value close to 1 increases randomness and creativity. For example, if `temperature` is set to 0.5, the model balances between deterministic and creative outputs.
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.
topK	Limits the number of the highest-probability tokens to consider during response generation. For example, if `topK` is set to 2, the model considers only the top 2 tokens at each step, controlling output diversity and quality.

After configuring the prompt instructions, the process reads the file using the XQuery function `fn:unparsed-text`. It reads the file stored in the database as either a text or binary file and returns its contents as a string.

Note: You can use files of the following formats: `.txt`, `.xml`, `.json`, `.html`, `.csv`, and `.md`

10. Save and publish the process.

Invoking the process

When you invoke the Chat with File process, the user sees an answer from the LLM based on the context of the file.

To invoke the process, perform the following steps:

1. Open the **Chat with File** process and click **Actions** > **Properties Detail** > **Copy Service URL**.

2. Open a text editor and add the input fields and values to the service URL as shown in the following format:

```
<Cloud Application Integration POD URL>/active-bpel/public/rt/<API_name>?
User_Prompt=<User_Prompt>&Path_To_File=<C:/folder>&Filename=<Filename>
```

3. Open a browser and paste the service URL.

You can also invoke the process using the Run Using option, REST or SOAP API endpoints in any API clients such as cURL, Postman, SOAP UI, or through any programming language.

The process prepares a request with the file and sends it to the LLM and the user sees an answer from the LLM based on the context of the file. If the file is missing or empty, the process is terminated with an appropriate message.