



Google BigQuery Adapter for SAP Integration Suite

Version 1.0.1 – September 2025

Contents

- 1 Introduction 4
 - 1.1 Objective 4
 - 1.2 Coding Samples 4
 - 1.3 Internet Hyperlinks 4
 - 1.4 Overview..... 4
 - 1.5 Features 5
- 2 Installation and Configuration..... 6
 - 2.1 Adapter Installation on Cloud Foundry..... 6
 - 2.1.1 Prerequisite 6
 - 2.1.2 Procedure 6
 - 2.1.2.1 Adapter Installation by Creating a New Integration Flow 6
 - 2.1.2.2 Adapter Installation without Creating a New Integration Flow.....7
 - 2.1.3 Monitor the Deployment Status 9
- 3 Getting Started: Google BigQuery Adapter10
 - 3.1 Architecture Overview10
 - 3.2 Application Configuration.....10
 - 3.3 Authentication..... 11
 - 3.3.1 Set up Workload Identity Federation (WIF)..... 11
 - 3.3.1.1 Creating OAuth2 Client Credentials 11
 - 3.3.2 Creating Keys for OAuth2 Service Account.....13
 - 3.3.2.1 Creating Keystore for JSON key14
 - 3.3.2.2 Creating Key-Pair for P12 key.....17
- 4 Google BigQuery Adapter Configuration19
 - 4.1 Receiver Adapter19
 - 4.1.1 Connection Tab19
 - 4.1.2 Processing Tab23
- 5 Google BigQuery Adapter Operations..... 27

5.1	Update Tables.....	27
5.2	Insert All TableData.....	29
5.3	Insert Job using Resumable Upload	31
5.4	List Datasets.....	34
5.5	List Row Access Policies	35
5.6	Insert Routines.....	36
5.7	Patch Models	37
6	References	39
6.1	Keystore Generation (JKS File Creation).....	39
7	Support.....	44
7.1	Troubleshooting	44

1 Introduction

1.1 Objective

This is the official guide for the Google BigQuery Adapter for SAP Integration Suite. This guide covers all relevant information for integration developers to start working with the Google BigQuery adapter. Read this guide carefully before using the Adapter.

1.2 Coding Samples

Any software coding and/or code lines/strings ("Code") included in this documentation are only examples and are not intended to be used in a productive system environment. The Code is only intended to better explain and visualize the syntax and phrasing rules of certain coding. The correctness and completeness of the Code given herein are not guaranteed.

1.3 Internet Hyperlinks

The documentation may contain hyperlinks to the Internet. These hyperlinks are intended to serve as a hint about where to find related information. The availability and the correctness of this related information or the ability of this information to serve a particular purpose are not warranted.

1.4 Overview

The Google BigQuery adapter is a Cloud Integration capability available on SAP Integration Suite that facilitates management and access to data residing on Google BigQuery. Maximise your productivity with data related integrations using Google BigQuery adapter.

1.5 Features

Google BigQuery Adapter has the following features:

- Supports **OAuth2 Service Account** and **Workload Identity Federation** authentication methods.
- Enables accessing and managing BigQuery resources like **datasets, jobs, tables, table data, routines, projects, row access policies, and models**.
- Supports convenient data retrieval using **List** operation which uses the **Page Token** field for efficient querying.
- Supports **Basic** configuration for convenient processing capability whereas **Advanced** enables proficient users to perform calls with greater control while connecting to any API endpoint.
- Supports **DML** operations like insert, update, patch for Google BigQuery resources.

2 Installation and Configuration

This section details the prerequisites to install and configure the Google BigQuery adapter.



The Google BigQuery adapter is available as part of your SAP Integration Suite license.

2.1 Adapter Installation on Cloud Foundry

Before the Google BigQuery adapter can be used in the Cloud Foundry environment, it must be deployed to the SAP Integration Suite tenant.

2.1.1 Prerequisite

To deploy the Google BigQuery adapter, you must have access to “*GoogleBigQuery Adapter for SAP Integration Suite*” as part of your SAP Integration Suite license.

2.1.2 Procedure



The below installation procedure is compatible with Apache Camel 2, Apache Camel 3, and Edge Integration Cell (EIC) platform.

You can deploy the adapter using the following methods:

2.1.2.1 Adapter Installation by Creating a New Integration Flow

The Google BigQuery adapter is available for selection in the Sender and Receiver adapter list and can be deployed in the **Design** tab directly as you use it in an Integration flow.




Purpose

To install an adapter for use in your Integration flow.

Procedure

Go to **Design** workspace and select the integration package where you want to create a new Integration flow.


1. Click **Edit** to make the package editable.

2. Go to the **Artifacts** tab. Click **Add** and select **Integration Flow**.
3. Enter the **Name** and **ID** for your flow. Additionally, select **Runtime Profile** from the drop-down and choose **Sender** and **Receiver** systems from the list . Finally, click **Add** to create the integration flow.
4. Go to the newly created integration flow and click **Edit** to make it editable.
 - i) For the Sender, in the integration flow add a **Connector**  between the **Sender box** and the **Start**.
 - ii) For the Receiver, in the integration flow, click **End** to add a **Connector**  between the **End** and the **Receiver Box**.
5. A drop-down with the available adapters appears. The **GoogleBigQuery** adapter should show up in the list.
6. Select the **GoogleBigQuery** adapter from the list.

The adapter is now imported which *triggers* an adapter deployment. Once the Google BigQuery Adapter is deployed, a success message is *displayed*.

After the above steps are done, the Google BigQuery Adapter is successfully deployed in your **Design** workspace of the SAP Integration Suite tenant.

2.1.2.2 Adapter Installation without Creating a New Integration Flow

 The following procedure describes how the Google BigQuery adapter is migrated from the Discover workspace to the Design workspace of the SAP Integration tenant.

This method is useful for scenarios where integration flow packages are migrated from development to a higher environment such as Production. The Google BigQuery adapter can be imported into the Design workspace without creating an integration flow.

Use the Transport Management Service (TMS) to import/transport the Google BigQuery adapter to a higher environment. Alternatively, if the TMS is not available in the landscape, the adapter package can be imported into the Design workspace by copying it from the Discover workspace.

Purpose

To import the **GoogleBigQuery** adapter to **Design** workspace by copying the integration package from **Discover** workspace.

Procedure

1. Go to **Discover** workspace.
2. In the search box, search for **GoogleBigQuery Adapter for SAP Integration Suite** package.
3. Select the package and click **Copy**.
This copies the package from the Discover workspace to Design workspace.
4. Go to Design workspace and select the copied **GoogleBigQuery Adapter for SAP Integration Suite** package.
5. In the **Actions** tab of the selected package, click **Deploy**.

This completes the adapter deployment to **Design** workspace.

2.1.3 Monitor the Deployment Status

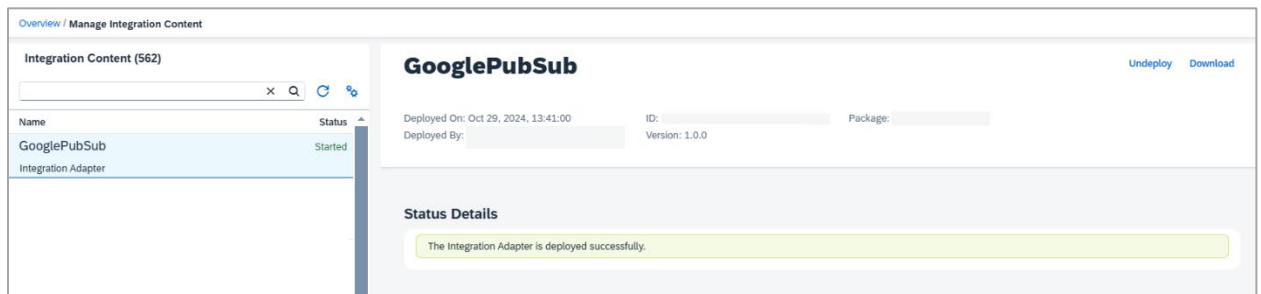
After the adapter deployment is complete, you can check the status in the **Monitor** section.

Purpose

To check the status of the deployed adapter.

Procedure

1. Under the **Monitor** tab, click **Integrations and APIs**. This opens the **Overview** page.
2. On the **Overview** page, go to **Manage Integration Content** section and click **All**. This opens **Integration Content** page with a list of all the deployed adapters.
3. Here, you can check and confirm the deployment status of your adapter.



3 Getting Started: Google BigQuery Adapter

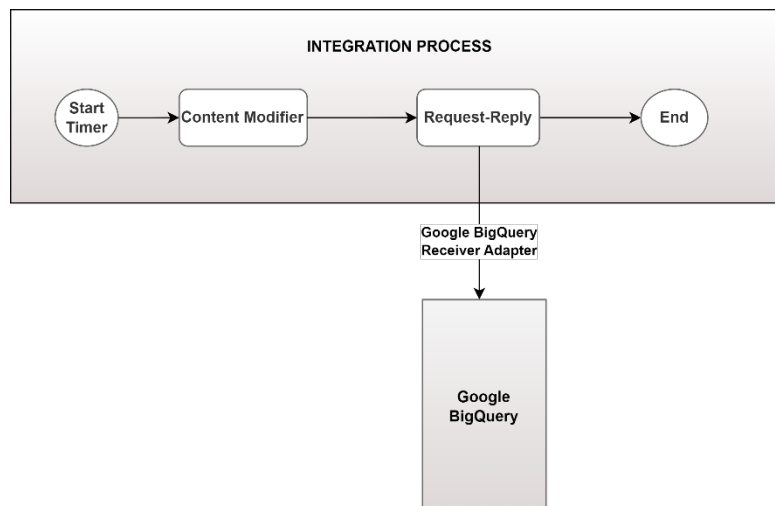
Before you start using the adapter, you can get to know underlying [Architecture](#) for the adapter and [Application Configuration](#) for Google BigQuery.

3.1 Architecture Overview

How Google BigQuery Receiver Adapter Works: The Google BigQuery Receiver adapter (as the name suggests) is designed to function as a receiver adapter.

In such a scenario where the adapter is used as a receiver adapter, SAP Integration Suite acts as the initiator of the calls.

SAP Integration Suite tenant sends the operation request to Google BigQuery (this is a receiver system), Google BigQuery Receiver Adapter works on the request and sends the data back to the SAP Integration Suite tenant.



3.2 Application Configuration

- To get an overview of BigQuery, see [Google BigQuery Overview](#).
- For more information about connecting to BigQuery, see [OAuth for Service Accounts](#).
- For more information about roles for your service account, see [Access Control](#).

3.3 Authentication

The Google Cloud Storage adapter supports authentication using OAuth Service Account and Workload Identity Federation (WIF).

Find more information about how to create service account, see [Create Service Account Keys](#).

3.3.1 Set up Workload Identity Federation (WIF)

- To learn more about WIF authentication, see [Workload Identity Federation](#).
- To set up WIF authentication, see [WIF with Azure](#).
- You must create a security artifact of type OAuth2 Client Credentials. For more information, see below **Creating OAuth2 Client Credentials**.

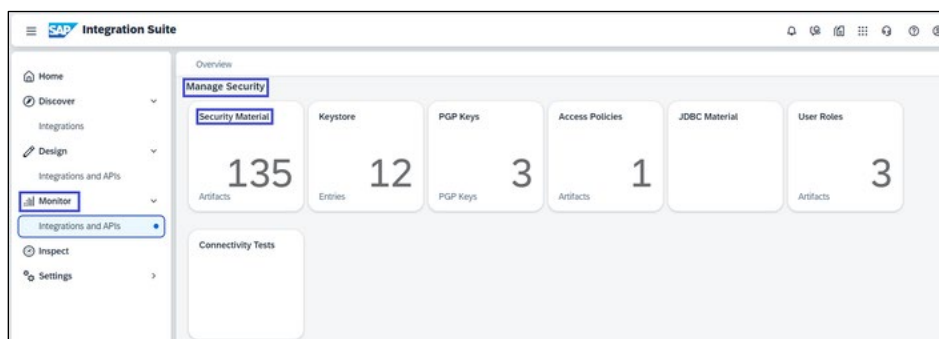
3.3.1.1 Creating OAuth2 Client Credentials

Purpose

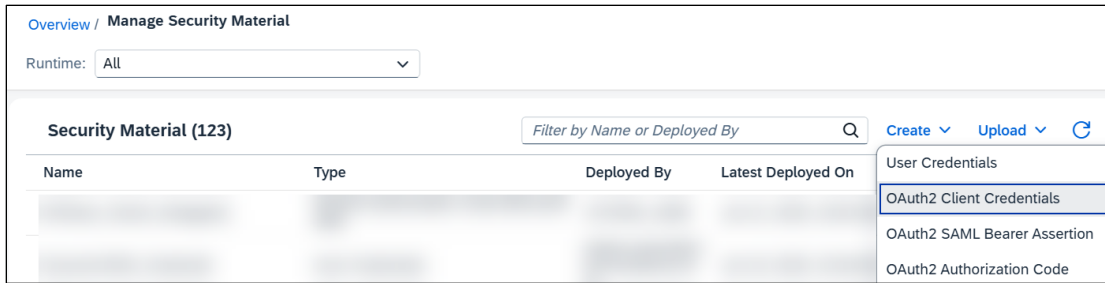
To create credentials in Security Material for **OAuth2 Client Credentials**.

Procedure

1. In SAP Integration Suite, navigate to **Monitor > Integrations and APIs**. This opens the **Overview** page.
2. On the **Overview** page, go to **Manage Security** section and click **Security Material**.




3. On Manage Security Material page, click Create to select **OAuth2 Client Credentials** from the dropdown.



4. In the Create **OAuth2 Client Credentials** popup, provide the below details.

Parameter	Description
Name	Specify the name for the credentials.
Description	Specify the description for the credentials.
Token Service URL	Specify the URL of the OAuth2 authorization server that issues the access token. Example: <code>https://login.microsoftonline.com/<tenant ID>/oauth2/token</code>
Client ID	Specify the ID of the client to which you are connecting.
Client Secret	Specify the Secret key of the client to which you are connecting. OAuth2 uses a multiple step authentication pattern: Client credentials (Client ID and Client Secret, as specified in the artifact) are used by the client application to initially request an access token. The access token is then used to authorize the client (for as long as the token is valid) to access the server's resources (for example, the resources that are used in the associated integration flow). In many OAuth2 scenarios, the access token is issued (or generated) by an authorization server.
Client Authentication	Select the Client Authentication which allows you to access an application using Client ID and Client Secret. By default, the Send as Body Parameter is selected, this option sends the Client ID and Client Secret as a JSON content to the authentication server in the request body. Example: Send as Body Parameter

Parameter	Description
Scope	Specify the OAuth2 scope information to be included in the request body. Example: <a href="https://iam.googleapis.com/projects/<projectid>/locations/global/workloadIdentityPools/<wif-pool>/providers/<wif-provider>/default">https://iam.googleapis.com/projects/<projectid>/locations/global/workloadIdentityPools/<wif-pool>/providers/<wif-provider>/default
Content Type	Select content type to indicate the media type. <div style="background-color: #e6f2ff; padding: 10px; border: 1px solid #cfe2f3;">  To use application permissions in the adapter, deploy an OAUTH2 Client Credential security artifact and ensure Content Type is set to 'application/x-www-form-urlencoded' </div>
Resource	Specify the identifier of the application or service that shares the same client secret. The identifier varies depending on the service that you want to connect with.
Audience	Specify the identifier of the application or service that shares the same client secret. The identifier varies depending on the service that you want to connect with.

- Click Deploy to complete the process.

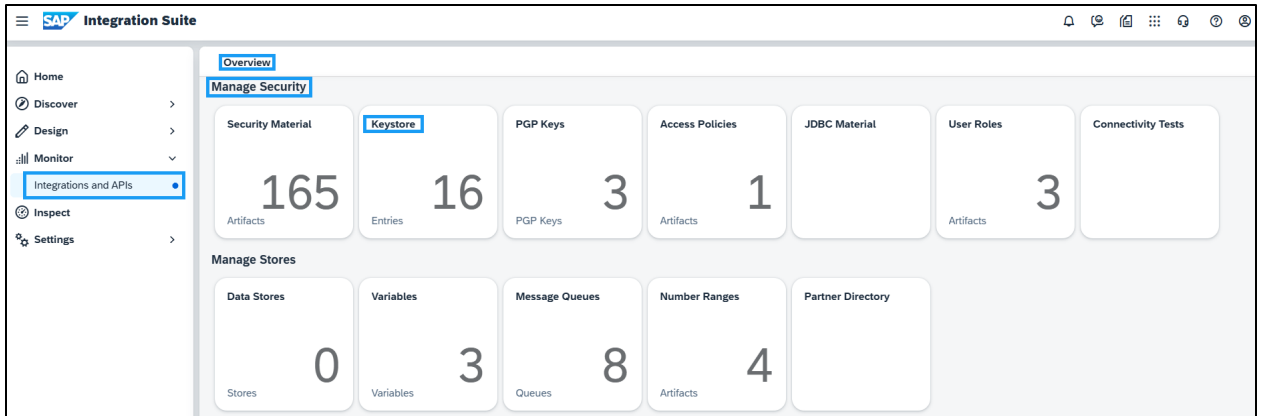
When you refresh the Manage Security Material page, the new artifact is displayed (with Type **OAuth2 Client Credentials**) in the artifact table.

3.3.2 Creating Keys for OAuth2 Service Account

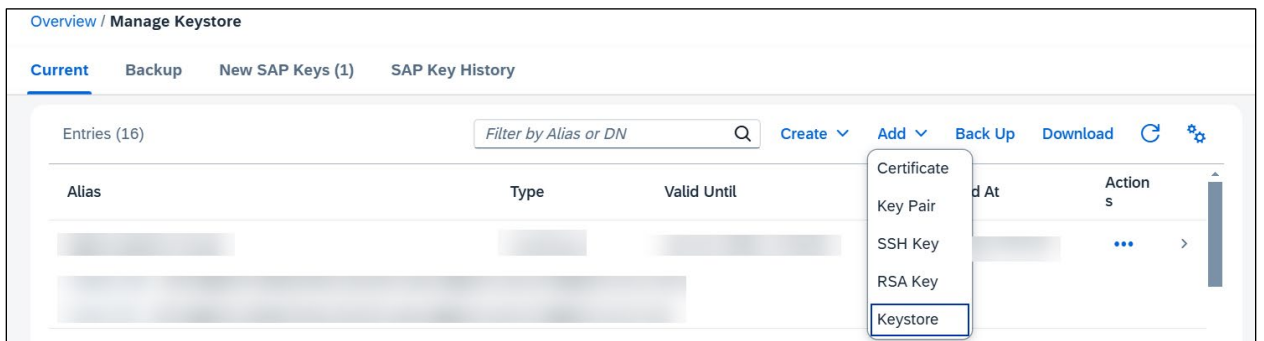
The Security artifact created below is used to connect to the Google BigQuery Application by configuring the **Connection tab** of the Adapter.

Log in to Google BigQuery and generate a key. For more information, see [Create Keys](#).

The **Key type** for private key can be either **JSON** or **P12**. Refer the following sections for each Key type usage.



- On **Manage Keystore** page, click **Add** to select **Keystore** from the dropdown.



- In the **Add Keystore** popup, provide the below details.

Add Keystore

Keystore: * Browse...

Passphrase: *

Action: Add ▼

Overwrite existing entries

Deploy
Cancel

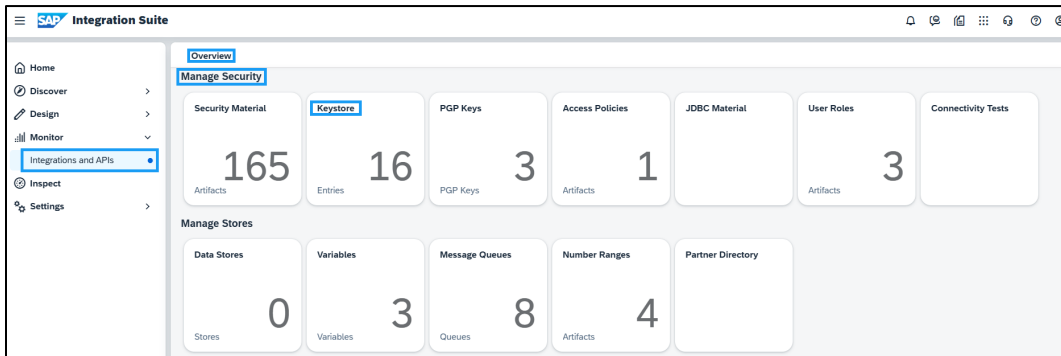
Parameter	Description
Keystore	Choose a keystore for upload. Select the .jks file. For detailed information, see Keystore Generation (JKS file creation) .
Passphrase	Specify the password used while creating the .jks file.
Action	Specify the action to be performed as Add .

Overwrite Existing Entries	Enable or Disable depending on your need to overwrite existing entries.
-----------------------------------	---

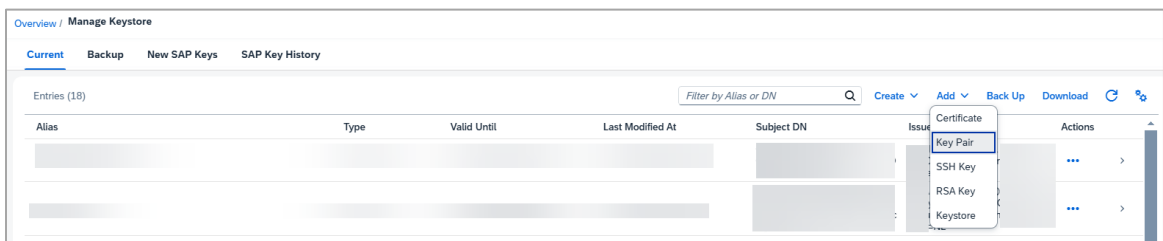
7. Click **Deploy** to complete the process.

3.3.2.2 Creating Key-Pair for P12 key

1. In SAP Integration Suite, navigate to **Monitor > Integrations and APIs**. This opens the **Overview** page.
2. On the **Overview** page, go to **Manage Security** section and click **Keystore**.



3. On **Manage Keystore** page, click **Add** to select **Key Pair** from the dropdown.



4. In the **Key Pair** popup, provide the below details.

Add Key Pair

Alias: *

File: * Browse...

Password: *

Add
Cancel

Parameter	Description
Alias	Specify the alias for the key pair.
File	Select the P12 key for upload.

Password	Specify the password used while creating the .jks file.
-----------------	---

5. Click **Deploy** to complete the process.

4 Google BigQuery Adapter Configuration

This section describes the parameters to be configured for your Google BigQuery adapter. You need to configure the **Connection** and **Processing** tabs. A description and example usage for each field has been added.

4.1 Receiver Adapter

4.1.1 Connection Tab




The **Connection** tab contains connection and authentication parameters for Google BigQuery. Before you set the connection details, see [Creating a Keystore](#).



The screenshot shows the configuration interface for the GoogleBigQuery adapter, specifically the Connection tab. The interface is titled "GoogleBigQuery" and has three tabs: "General", "Connection", and "Processing". The "Connection" tab is selected. Below the tabs, the section is titled "CONNECTION DETAILS". The configuration fields are as follows:


Address: *	https://bigquery.googleapis.com
Authentication Type:	OAuth2 Service Account
OAuth2 Token URL: *	https://www.googleapis.com/oauth2/v4/token
Client Email: *	
Private Key Alias: *	
Scope: *	https://www.googleapis.com/auth/bigquery
Proxy Type:	On-Premise
Location ID:	
Reuse Connection:	<input type="checkbox"/>
Connection Timeout (in ms):	60000
Response Timeout (in ms):	60000

The above screenshot portrays a sample configuration using **OAuth2 Service Account** authentication.

The connection tab contains the following fields:

Parameter	Description
Address	<p>Specify the name for the Google BigQuery service.</p> <p>Default: <code>https://bigquery.googleapis.com</code></p> <div style="background-color: #e1f5fe; padding: 10px; border: 1px solid #ccc;"> <p> When Proxy Type is set to On-Premise, you must use virtual host and port from cloud connector.</p> </div>
Authentication Type	<p>Select your preferred method for authentication:</p> <ul style="list-style-type: none"> • OAuth2 Service Account • Workload Identity Federation
<p> The below fields are applicable when Authentication Type is set to OAuth2 Service Account.</p>	
OAuth2 Token URL	<p>Specify the OAuth2 Token URL which identifies as the authorization server for producing a JWT token internally.</p> <p>Default: <code>https://www.googleapis.com/oauth2/v4/token</code></p>
Client Email	<p>Specify the Google Service Account Client Email.</p> <p>Example: <code>limited-svc-account@192843.iam.gserviceaccount.com</code></p>
Private Key Alias	<p>Specify the Keystore artifact for storing the Google Service Account Private Key.</p> <p>For more information, see Creating Keys for OAuth2 Service Account.</p>
Scope	<p>Specify the scope of the connection to Google BigQuery service.</p> <p>Default: <code>https://www.googleapis.com/auth/bigquery</code></p>
<p> The below fields are applicable when Authentication Type is set to Workload Identity Federation.</p>	

Parameter	Description
Microsoft Entra ID Credential Name	Specify the OAuth2 Client Credentials security artifact created for Microsoft Entra ID. For more information, see Set up Workload Identity Federation (WIF) .
STS Token URL	Specify the Security Token Service URL. Example: https://sts.googleapis.com/v1/token
STS Audience	Specify the audience i.e. the full resource name of the identity provider. Example: <code>//iam.googleapis.com/projects/<project-number>/locations/global/workloadIdentityPools/<pool-id>/providers/<provider-id></code>
STS Scopes	Specify the OAuth 2.0 scopes to include on the resulting access token, formatted as a list of space-delimited, case-sensitive strings. Example: https://www.googleapis.com/auth/cloud-platform
STS Options	Specify the additional options to use in the STS token call.
 The below fields are applicable when Use Service Account Impersonation is enabled.	
Access Token URL	Specify the URL for generating access token with Service Account Impersonation. Example: <code>https://iamcredentials.googleapis.com/v1/projects/-/serviceAccounts/{ACCOUNT_EMAIL_OR_UNIQUEID}:generateAccessToken</code>  The - wildcard character is required; replacing it with a project ID is invalid.
Service Account Impersonation Scopes	Specify the comma-separated scopes to be included in the resulting OAuth 2.0 access token. Example: https://www.googleapis.com/auth/pubsub , https://www.googleapis.com/auth/cloud-platform

Parameter	Description
Delegates	Specify the comma-separated sequence of service accounts in a delegation chain. Example: <code>projects/-/serviceAccounts/{ACCOUNT_EMAIL_OR_UNIQUEID}</code>
Use Service Account Impersonation	Enable to use Service Account Impersonation.
 The below fields are applicable are available for all authentication methods.	
Proxy Type	Specify the proxy type: <ul style="list-style-type: none"> • Internet • On-Premise
Location ID (Only available when Proxy Type is set to On-Premise).	Specify the Location ID from Cloud Connector.
Reuse Connection	Enable this property to reuse the connection.
Connection Timeout (in ms)	Specify the maximum waiting time (in milliseconds) for the connection to be established with Google BigQuery service.
Response Timeout (in ms)	Specify the maximum waiting time (in milliseconds) for a response message to be received from Google BigQuery service.


4.1.2 Processing Tab


The **Processing** tab contains the operational configurations for the Google BigQuery receiver adapter.

The screenshot shows the 'Processing' tab of the GoogleBigQuery configuration interface. It is divided into two sections: 'PROCESSING DETAILS' and 'HEADER DETAILS'. In the 'PROCESSING DETAILS' section, there are several configuration fields: 'Configuration Type' is a dropdown menu set to 'Basic'; 'Entity' is a dropdown menu set to 'Routines'; 'Operation' is a dropdown menu set to 'Insert'; 'Project ID' is a text field containing 'gbq-docs'; 'Dataset ID' is a text field containing 'org_primary_dataset'; 'Query Parameters' is an empty text field. In the 'HEADER DETAILS' section, there are two text fields: 'Request Headers' is empty, and 'Response Headers' contains an asterisk (*).

The above image shows a sample processing configuration for Insert Routines in Google BigQuery.

Parameter	Description
Configuration Type	Select the required configuration type: <ul style="list-style-type: none">• Basic to use the dropdowns and parameter text fields.• Advanced to specify the relative URL.
Entity	Select the entity for operation: <ul style="list-style-type: none">• Datasets• Jobs• Models• Projects• Routines• Row Access Policies• Table Data• Tables

Parameter	Description
Operation	Select the operation to be performed.
Project ID	Specify the Google Project ID.
Dataset ID	Specify the Dataset ID to be used.
Job ID	Specify the Job ID.
Model ID	Specify the Model ID.
Policy ID	Specify the Policy ID.
Table ID	Specify the Table ID.
Max Results	Specify the maximum results to be returned. Example: 100
Page Token	Specify the nextPageToken returned in the response, to retrieve the next page of results. <div data-bbox="483 999 1373 1108" style="background-color: #e6f2ff; padding: 5px; border: 1px solid #ccc;">  A value of null or an empty string retrieves the first page. </div>
State Filter	Select the state filter for the jobs: <ul style="list-style-type: none"> • Done • Pending • Running
Object Metadata	Specify the object metadata.
Content Type	Specify the content type of the object data. Example: <ul style="list-style-type: none"> ▪ text/html ▪ media-type

Parameter	Description
Filter	Select the filter for routines: <ul style="list-style-type: none"> • Procedure • Routine Type Unspecified • Scalar Function • Table Valued Function
HTTP Method	Select the required HTTP method from the available dropdown: <ul style="list-style-type: none"> • DELETE • GET • HEAD • PATCH • POST • PUT
Relative URL	Specify the endpoint path, excluding the Host. Example: <code>/bigquery/v2/projects/my_project/datasets</code>
Query Parameters	Specify the query that should be transferred with the HTTP request. Example: <code>"param1=value1&param2=value2"</code> . <div style="background-color: #e6f2ff; padding: 10px; margin-top: 10px;"> <p>  Ensure that the query parameter value is properly encoded before being passed. <code>filter=labels.department:finance%20labels.owner:data-team</code> </p> </div>
Request Headers	Enter a list of custom headers, separated by a pipe (), to be sent to the target system. Use an asterisk (*) to send all custom headers to the target system. All Camel-specific headers and HTTP protocol headers except "date" are excluded by default even if you specify them. Note that this value can also be read dynamically using an exchange header or property.

Parameter	Description
Response Headers	<p>Enter a list of headers, separated by a pipe (), coming from the target system's response to be received in the message. Use an asterisk (*) to receive all the headers from the target system, which is also the default value.</p> <p>All Camel-specific headers and HTTP protocol headers except "date" are excluded by default even if you specify them. Note that this value can also be read dynamically using an exchange header or property.</p>

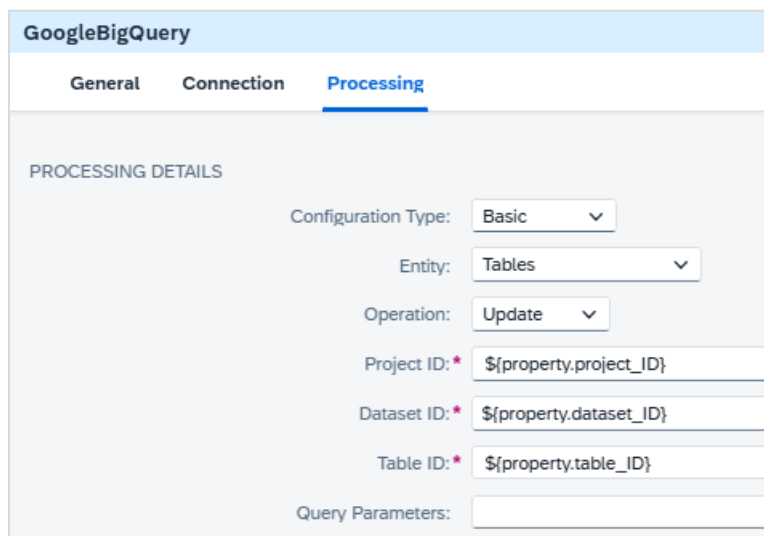
5 Google BigQuery Adapter Operations

This section describes some of the operations supported by the Google BigQuery adapter.

Note: Each operation requires a specific set of permissions depending on the entity and operation.

5.1 Update Tables

You can update the details for a specific table by providing a JSON payload.



The screenshot shows the 'Processing' tab of the GoogleBigQuery configuration interface. Under the 'PROCESSING DETAILS' section, the following settings are visible:

- Configuration Type: Basic
- Entity: Tables
- Operation: Update
- Project ID: * \${property.project_ID}
- Dataset ID: * \${property.dataset_ID}
- Table ID: * \${property.table_ID}
- Query Parameters: (empty field)

Parameter	Values
Configuration Type	Basic
Entity	Tables
Operation	Update
Project ID	Set as <code>\${property.project_ID}</code>
Dataset ID	Set as <code>\${property.dataset_ID}</code>
Table ID	Set as <code>\${property.table_ID}</code>

Sample Payload:

```
{
  "tableReference": {
    "projectId": "gbq-docs",
    "datasetId": "org_primary_dataset",
    "tableId": "employee"
  },
  "friendlyName": "Employee Data Table",
  "description": "This table stores employee details and creation
  timestamps."
}
```

You can achieve the same using **Advanced** configuration type as well.

The screenshot shows the 'GoogleBigQuery' configuration interface. The 'Processing' tab is selected. Under 'PROCESSING DETAILS', the 'Configuration Type' is set to 'Advanced', the 'HTTP Method' is 'PUT', and the 'Relative URL' is '/bigquery/v2/projects/org/datasets/department/tables/employee'.

Parameter	Values
Configuration Type	Advanced
HTTP Method	PUT
Relative URL	bigquery/v2/projects/{projectId}/datasets/{datasetId}/tables/{tableId}

5.2 Insert All TableData

Allows you to insert data into a table.

The screenshot shows the 'Processing' tab in the Google BigQuery configuration interface. Under 'PROCESSING DETAILS', the following settings are visible:

- Configuration Type: Basic
- Entity: Table Data
- Operation: Insert All
- Project ID: CXF_DE
- Dataset ID: `${property.table_ID}`
- Table ID: `${property.dataset_ID}`
- Query Parameters: (empty field)

Parameter	Values
Configuration Type	Basic
Entity	Table Data
Operation	Insert All
Project ID	CXF_DE
Dataset ID	Set as <code>\${property.dataset_ID}</code>
Table ID	Set as <code>\${property.table_ID}</code>

The sample payload has been outlined in the next page.

```
{
  "rows": [
    {
      "json": {
        "Name": "John Doe",
        "ID": 123,
        "Variable": 45.67,
        "Secret_Code": "U29tZSBieXRlcyBkYXRh",
        "Is_Permanent": true,
        "joining_date": "2024-03-11",
        "latest_timestamp": "2024-03-11T14:30:00",
        "timestamp": "1709802600",
        "time": "14:30:00",
        "location": "POINT(-73.935242 40.730610)",
        "Variable1": "12345.678901234",
        "Variable2": "98765432101234567890.123456789012345678901234567890",
        "additional": {
          "sub_field1": "Blood",
          "sub_field2": "A+"
        },
        "misc": {
          "sub_field1": "Code",
          "sub_field2": "ACX"
        }
      }
    },
    {
      "json": {
        "Name": "Barry Allen",
        "ID": 456,
        "Variable": 98.76,
        "Secret_Code": "QW5vdGhlciBieXRlcyBkYXRh",
        "Is_Permanent": false,
        "joining_date": "2024-03-12",
        "latest_timestamp": "2024-03-12T16:45:00",
        "timestamp": "1709889000",
        "time": "16:45:00",
        "location": "POLYGON((-73.935242 40.730610, -73.935242 40.731610, -73.934242 40.731610, -73.934242 40.730610, -73.935242 40.730610))",
        "Variable1": "98765.432109876",
        "Variable2": "12345678901234567890.987654321098765432109876543210",
        "additional": {
          "sub_field1": "Blood",
          "sub_field2": "A+"
        },
        "misc": {
          "sub_field1": "Code",
          "sub_field2": "ACX"
        }
      }
    }
  ]
}
```

5.3 Insert Job using Resumable Upload

You can create an Insert job with resumable uploads to insert data in chunks.

Select the **Upload Type** as Resumable, you must also specify a metadata file that specifies value for **Object Metadata** and **Chunk Size**.

The screenshot shows the 'Processing' tab of the Google BigQuery configuration interface. The 'PROCESSING DETAILS' section includes the following fields:

- Configuration Type: Basic (dropdown)
- Entity: Jobs (dropdown)
- Operation: Insert (dropdown)
- Project ID: * \${header.Project_ID}
- Upload Type: Resumable (dropdown)
- Object Metadata: * \${header.metadata}
- Content Type: * */*
- Upload in Multiple Chunks:
- Chunk Size: * \${header.chunk_size}
- Query Parameters: \${header.Query_Parameters}

The 'HEADER DETAILS' section includes:

- Request Headers: (empty text field)
- Response Headers: *

Parameter	Values
Configuration Type	Basic
Entity	Jobs
Operation	Insert

Parameter	Values
Project ID	<code>\${header.Project_ID}</code>
Upload Type	Resumable
Object Metadata	<code>\${header.metadata}</code>
Content Type	<code>*/*</code>
Upload in Multiple Chunks	Enable this checkbox
Chunks Size	<code>\${header.chunk_size}</code>
Query Parameters	<code>\${header.Query_Parameters}</code>

You need to specify the metadata and a valid payload.

A sample metadata is provided below

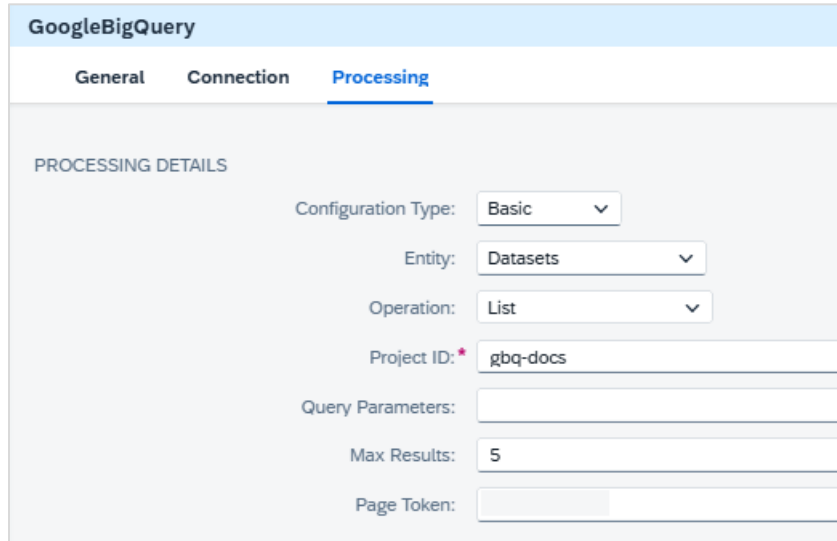
```
{
  "jobReference": {
    "projectId": "org_prj",
    "jobId": "{{randomJobId}}"
  },
  "configuration": {
    "load": {
      "destinationTable": {
        "projectId": "org_prj",
        "datasetId": "dataset_get_jobs_resumable",
        "tableId": "table_get_jobs_resumable"
      },
      "schema": {
        "fields": [
          {
            "name": "id",
            "type": "INTEGER",
            "mode": "NULLABLE"
          },
          {
            "name": "name",
            "type": "STRING",
            "mode": "NULLABLE"
          },
          {
            "name": "email",
            "type": "STRING",
            "mode": "NULLABLE"
          },
          {
            "name": "created_at",
            "type": "TIMESTAMP",
            "mode": "NULLABLE"
          }
        ]
      },
      "sourceFormat": "NEWLINE_DELIMITED_JSON",
      "writeDisposition": "WRITE_APPEND"
    }
  }
}
```

The payload for two rows corresponding to the above metadata is given below:

```
{"id":33,"name":"Sam_Evans","email":"sam@hotmail.com","created_at":"2024-03-06T12:00:00Z"}
{"id":44,"name":"Bob_Mark","email":"bob@hotmail.com","created_at":"2024-03-06T13:00:00Z"}
```

5.4 List Datasets

This operation allows you to list the datasets under a project. You can use **Max Results** to limit the number of datasets returned.

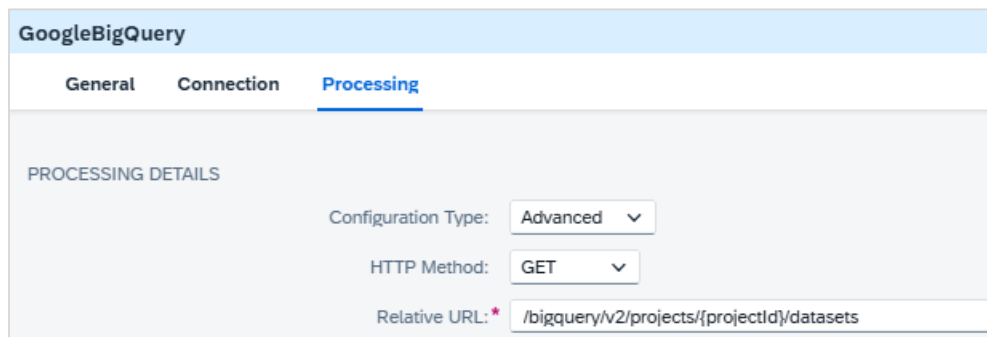


The screenshot shows the 'Processing' tab in the Google BigQuery interface. Under 'PROCESSING DETAILS', the following configuration is shown:

- Configuration Type: Basic
- Entity: Datasets
- Operation: List
- Project ID: gbq-docs
- Query Parameters: (empty)
- Max Results: 5
- Page Token: (empty)

Parameter	Values
Configuration Type	Basic
Entity	Datasets
Operation	List
Project ID	gbq_docs
MaxResults	5

You can achieve the same using **Advanced** configuration type as well.



The screenshot shows the 'Processing' tab in the Google BigQuery interface. Under 'PROCESSING DETAILS', the following configuration is shown:

- Configuration Type: Advanced
- HTTP Method: GET
- Relative URL: /bigquery/v2/projects/{projectId}/datasets

Parameter	Values
Configuration Type	Advanced
HTTP Method	GET
Relative URL	/bigquery/v2/projects/{projectId}/datasets

5.5 List Row Access Policies

This operation allows you to view all existing row access policies on a table. You can use **Max Results** to limit the number of results returned.

The screenshot shows the 'Processing' tab in the Google BigQuery interface. Under 'PROCESSING DETAILS', the following settings are visible:

- Configuration Type: Basic
- Entity: Row Access Policies
- Operation: List
- Project ID: gbq-docs
- Dataset ID: org_primary_dataset
- Table ID: employee_copy
- Query Parameters: (empty)
- Page Token: (empty)
- Page Size: 5

Parameter	Values
Configuration Type	Basic
Entity	Row Access Policies
Operation	List
Project ID	Set as gbq-docs
Dataset ID	Set as org_primary_dataset
Table ID	Set as employee_copy

Parameter	Values
Page Token	Specify the page token.
Page Size	5

5.6 Insert Routines

You can provide a json payload to insert a routine in Google BigQuery.

The screenshot shows the 'Processing' tab in the Google BigQuery interface. Under the 'PROCESSING DETAILS' section, the following configuration is visible:

- Configuration Type: Basic
- Entity: Routines
- Operation: Insert
- Project ID: CXF_DE
- Dataset ID: \${property.dataset_ID}

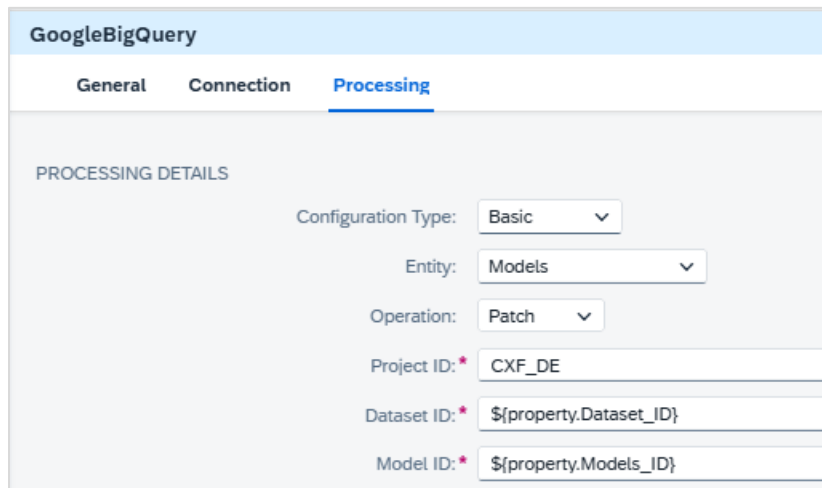
Parameter	Values
Configuration Type	Basic
Entity	Routines
Operation	Insert
Project ID	Set as CXF_DE
Dataset ID	Set as \${property.dataset_ID}

Sample Payload:

```
{
  "routineReference": {
    "projectId": "gbq-docs",
    "datasetId": "org_primary_dataset",
    "routineId": "routine_id_scalar_1"
  },
  "routineType": "SCALAR_FUNCTION",
  "language": "JAVASCRIPT",
  "arguments": [
    {
      "name": "input_value",
      "dataType": {
        "typeKind": "STRING"
      }
    }
  ],
  "returnType": {
    "typeKind": "STRING"
  },
  "definitionBody": "function myFunction(input_value) { return
input_value.toUpperCase(); }"
}
```

5.7 Patch Models

You can modify the model's metadata or configuration (like updating labels, description, or expiration time) using this operation.



The screenshot shows the Google BigQuery console interface. At the top, there are three tabs: 'General', 'Connection', and 'Processing', with 'Processing' selected. Below the tabs, the section is titled 'PROCESSING DETAILS'. It contains several configuration fields:

- Configuration Type: Basic (dropdown menu)
- Entity: Models (dropdown menu)
- Operation: Patch (dropdown menu)
- Project ID: CXF_DE (text input)
- Dataset ID: \${property.Dataset_ID} (text input)
- Model ID: \${property.Models_ID} (text input)

Parameter	Values
Configuration Type	Basic
Entity	Models
Operation	Insert
Project ID	CXF_DE
Dataset ID	Set as \${property.Dataset_ID}
Model ID	Set as \${property.Models_ID}

Sample Payload:

```
{
  "description": "Updated model to test patch operation of adapter.",
  "labels": {
    "team": "big query adapter-patch",
    "version": "v3"
  },
  "friendlyName": "Patch model"
}
```

6 References

6.1 Keystore Generation (JKS File Creation)

1. Generate a certificate signing request by using the `gbq_rsa_key.p8` file and store it in a file called `gbq_server.csr`.

OpenSSL Command:

```
openssl req -new -key <source_path>/gbq_rsa_key.p8 -out  
<target_path>/gbq_server.csr
```

You are about to be asked to enter information that will be incorporated into your certificate request.

What you are about to enter is what is called a Distinguished Name or a DN.

There are quite a few fields but you can leave some blank

For some fields there will be a default value,

If you enter '.', the field will be left blank.

Country Name (2 letter code) [AU]:<enter_country_code>

State or Province Name (full name) [Some-State]:<enter_city_name>

Locality Name (eg, city) []:<enter_locality_name>

Organization Name (eg, company) [Internet Widgits Pty Ltd]:<enter_org_name>

Organizational Unit Name (eg, section) []:<enter_org_unit_name>

Common Name (e.g. server FQDN or YOUR name) []:<enter_common_name>

Email Address []:<enter_email_address>

Please enter the following 'extra' attributes

to be sent with your certificate request

A challenge password []:<password>

An optional company name []:<company_name>

2. It is possible to sign the certificate using a self-signing mechanism or via CA.



Please refer to your organisation policy to choose a suitable signing mechanism for the certificate.

You can generate a self-signed digital certificate using our private key

(`gbq_rsa_key.p8`) and certificate signing request (`gbq_server.csr`) file and store the certificate in a file called (`gbq_server.crt`).

This `gbq_server.crt` is uploaded when creating the jks file.

OpenSSL

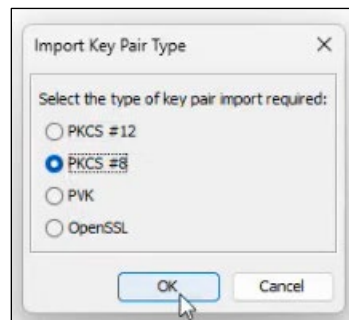
Command:

```
openssl x509 -req -sha256 -days 365 -in  
<source_path>/gbq_server.csr -signkey  
<source_path>/gbq_rsa_key.p8 -out <target/path>/gbq_server.crt
```

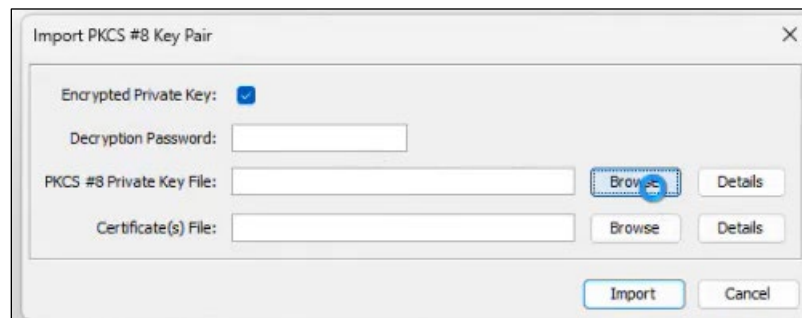
3. To create a JKS file using Keystore in Windows follow the below steps:
 - i. Go to KeyStore and **Import key Pair**.

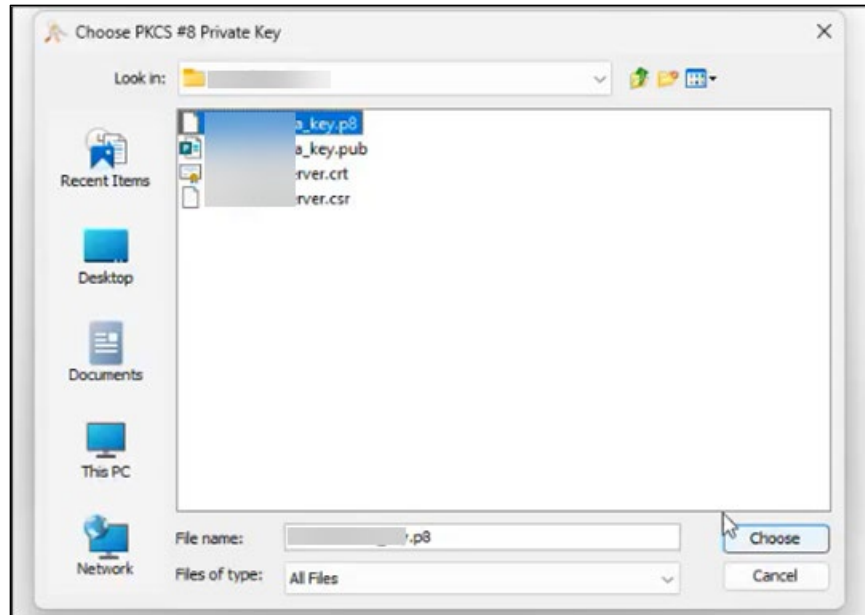


- ii. Select the type of key pair as **PKCS #8**.

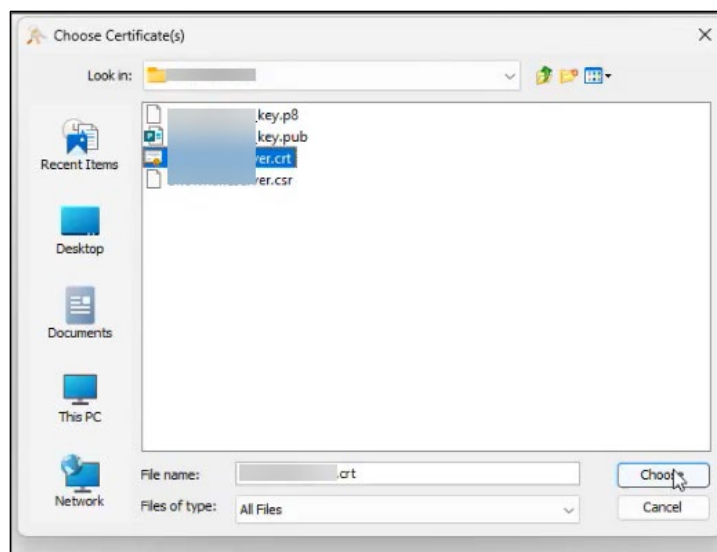
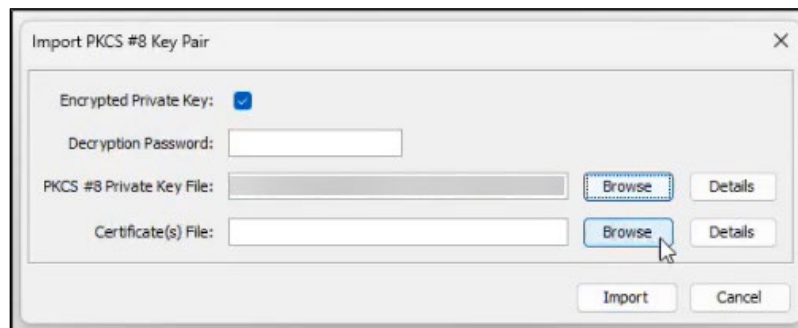


- iii. Add the private key file by uploading the PKCS **Private Key** file.

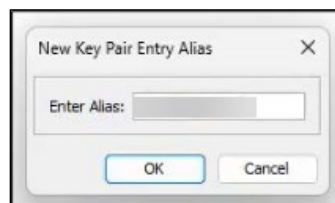
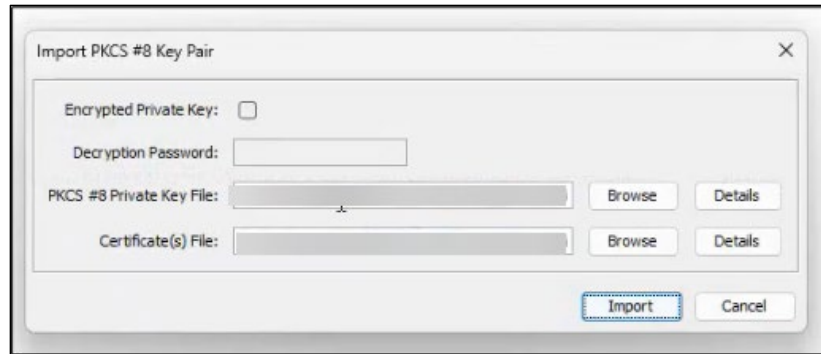




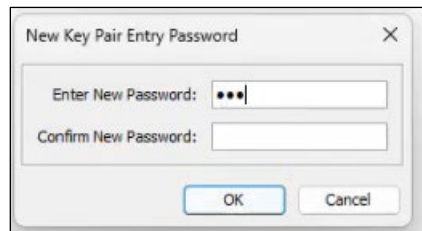
iv. Add the Google BigQuery **Certificate(s) File** created above.



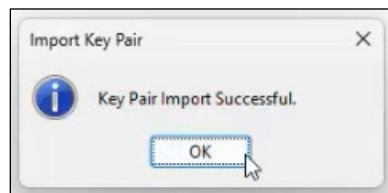
- v. Click **Import** and provide the **Alias**.



- vi. Provide a **New Key Pair Entry Password**.



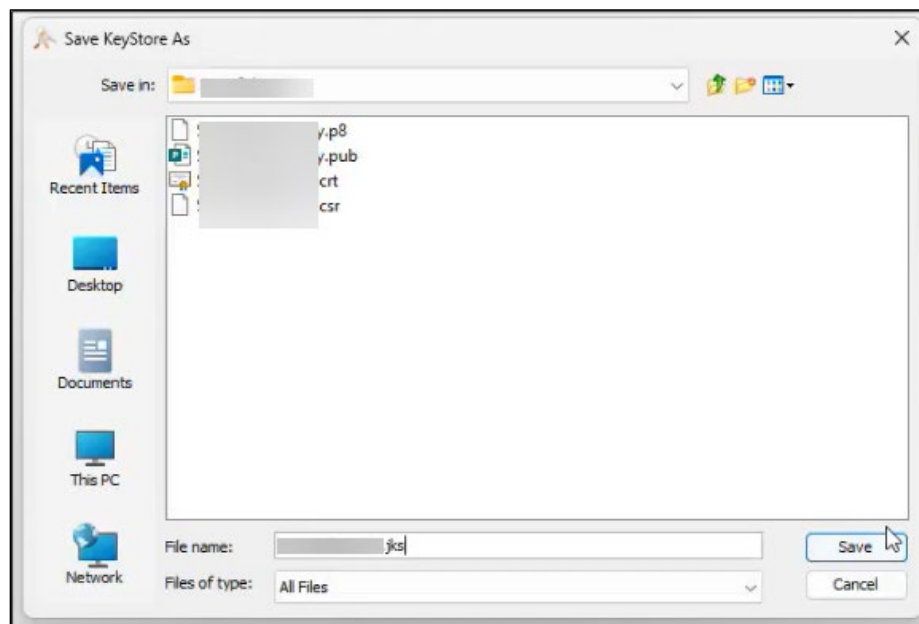
- vii. You will get a message that indicates key pair import was successful.



viii. Provide the **KeyStore Password**.



ix. Finally, save the jks file.



7 Support

7.1 Troubleshooting

A few scenarios along with their error messages and possible reasons for occurrence have been documented below.

Scenario	Error Message	Error Reasoning
After an InsertAll operation, executing an UPDATE or DELETE operation results in a failure.	<code>UPDATE or DELETE statement over table <tablename> would affect rows in the streaming buffer, which is not supported.</code>	An UPDATE or DELETE operation following InsertAll operation, can fail because it is not possible to modify (UPDATE or DELETE) a table with an active streaming buffer. For more information about this behaviour, see Life of a BigQuery streaming insert Google Cloud Blog .
Table not found during insertAll operation	<code>Table <tablename> not found</code>	This is a behaviour where the table isn't fully prepared or found yet. Make sure you create the table with a schema , wait until it is active, and double-check your project/setup.
InsertAll throws an error for input files greater than 10 MB	<code>com.sap.it.rt.adapter.http.api.exception.HttpResponseException: An internal server error occurred: Error message: Error occurred,</code>	As per Quotas & Limits documentation on BigQuery, the HTTP request size limit has been restricted to 10 MB, hence the error.

Scenario	Error Message	Error Reasoning
	Check the Request and try again., Reason: error 32 - Broken pipe (Write failed) (local port <port_number> to address <address> , remote host unknown). The MPL ID for the failed message is :<mpl_id>	For more than 10 MB, use insert job via resumable upload.