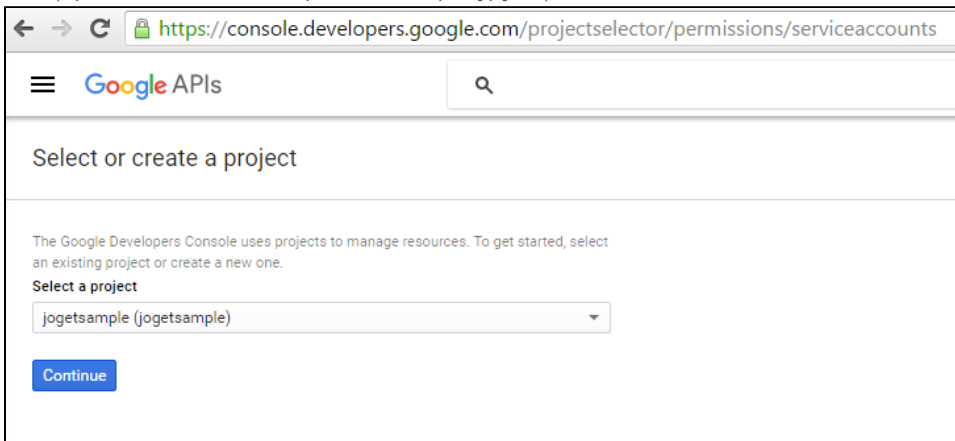


# Google API Configuration

- Create service account for the project
- Enable Google Drive API for the project
- Delegate Google Drive access to the project service account
- Configure Joget Workflow App
- Google Drive Form Element

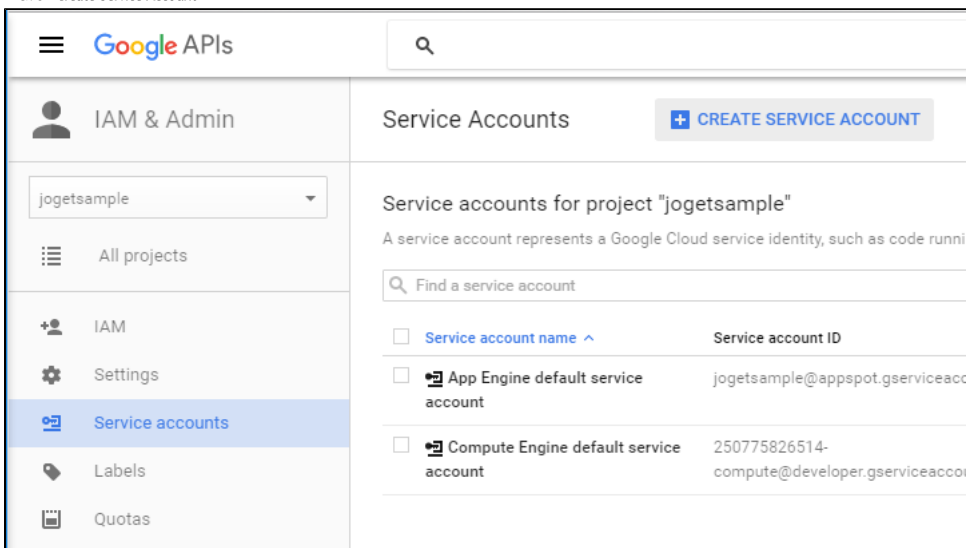
## Create service account for the project

1. Go to <https://console.developers.google.com/projectselector/permissions/serviceaccounts>
2. Select a project. You can create a new one too if you do not have one yet (e.g. jogetsample).



The screenshot shows the 'Select or create a project' page in the Google Developers Console. The URL in the address bar is <https://console.developers.google.com/projectselector/permissions/serviceaccounts>. The page has a header with the 'Google APIs' logo and a search bar. Below the header, the main heading is 'Select or create a project'. A subheading reads: 'The Google Developers Console uses projects to manage resources. To get started, select an existing project or create a new one.' There is a section titled 'Select a project' with a dropdown menu showing 'jogetsample (jogetsample)'. Below the dropdown is a blue 'Continue' button.

3. Click on **Create Service Account**.



The screenshot shows the 'Service Accounts' page in the Google Developers Console for the project 'jogetsample'. The URL in the address bar is <https://console.developers.google.com/projectselector/permissions/serviceaccounts>. The page has a header with the 'Google APIs' logo and a search bar. On the left, there is a sidebar with the 'IAM & Admin' section expanded, showing 'Service accounts' as the selected option. The main content area is titled 'Service Accounts' and has a blue '+ CREATE SERVICE ACCOUNT' button. Below this, the heading is 'Service accounts for project "jogetsample"'. A subheading reads: 'A service account represents a Google Cloud service identity, such as code running on Google Cloud.' There is a search bar labeled 'Find a service account'. Below the search bar is a table with two columns: 'Service account name' and 'Service account ID'. The table lists two service accounts: 'App Engine default service account' with ID 'jogetsample@appspot.gserviceaccount.com' and 'Compute Engine default service account' with ID '250775826514-compute@developer.gserviceaccount.com'.

Service account name	Service account ID
App Engine default service account	jogetsample@appspot.gserviceaccount.com
Compute Engine default service account	250775826514-compute@developer.gserviceaccount.com

4. Name the service account (e.g. joetsample).

Remember to check **Furnish a new private key** and also **Enable Google Apps Domain-wide Delegation**. Press **Create**.

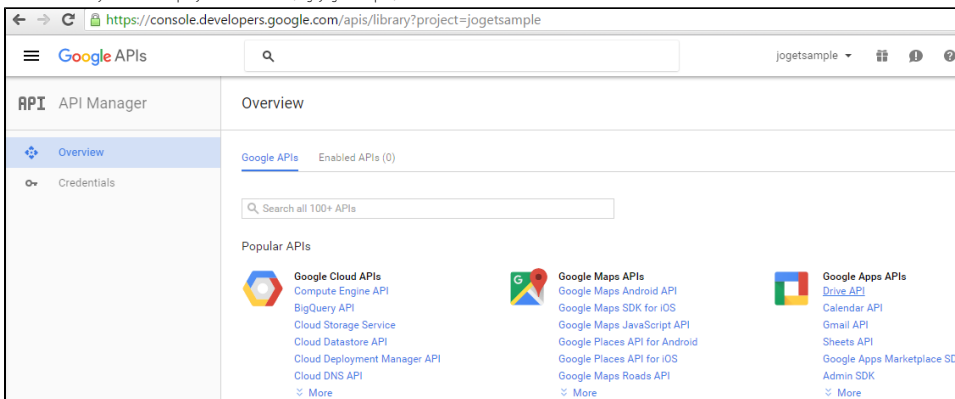
The screenshot shows the 'Create service account' dialog in the Google Cloud console. The dialog is titled 'Create service account' and has a sidebar on the left with options: 'Service account', 'App Engine account', and 'Compute Engine account'. The main form contains the following fields and options:

- Service account name:** A text input field containing 'joetsample'.
- Service account ID:** A text input field containing 'joetsample', followed by '@joetsample.iam.gserviceaccount.com' and a refresh icon.
- Furnish a new private key:** A checked checkbox. Below it, text says: 'Downloads a file that contains the private key. Store the file securely because this key can't be recovered if lost.'
- Key type:** Two radio button options: 'JSON' (selected, with 'Recommended' text below it) and 'P12' (with 'For backward compatibility with code using the P12 format' text below it).
- Enable Google Apps Domain-wide Delegation:** A checked checkbox. Below it, text says: 'Grants a client access to all users' data on a Google Apps domain without manual authorization on their part. [Learn more](#)'.
- Information box:** A box with an 'i' icon containing the text: 'To change settings for Google Apps domain, product name for the OAuth consent screen must be configured. Assign the product name below or configure the OAuth consent screen.'
- Product name for the consent screen:** A text input field containing 'Joget Workflow'.
- Buttons:** At the bottom, there are three buttons: 'Create' (blue), 'Configure consent screen' (grey), and 'Cancel' (grey).

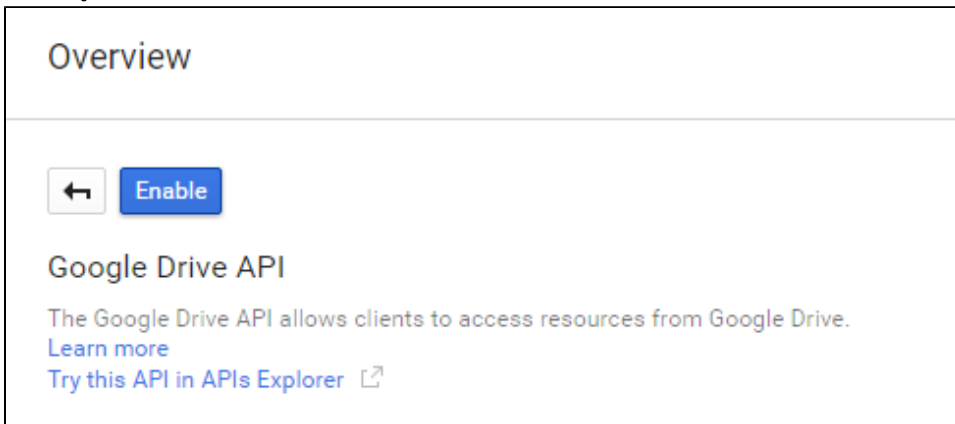
5. A .json file will be downloaded. Open up the file in your favorite text editor.

## Enable Google Drive API for the project

1. Go to <https://console.developers.google.com/apis/library>
2. Make sure that your current project is selected (e.g. joetsample).



3. Click on **Google Drive API**.



4. **Enable** it.

#### Delegate Google Drive access to the project service account

1. Go to <https://admin.google.com/AdminHome?chromeless=1#OGX:ManageOAuthClients>.
2. From the downloaded JSON file earlier, copy the "client\_id" and put into the **Client Name** field.

The screenshot shows the 'Manage API client access' page in the Google Admin console. At the top, there's a Google logo and a search bar with the placeholder text 'Search for users, groups, and settings (e.g. verify your domain)'. Below this is a green header with a menu icon and the word 'Security'. The main section is titled 'Manage API client access' with a subtext: 'Developers can register their web applications and other API clients with Google to enable access to data in Google data without your users having to individually give consent or their passwords.' There's a 'Learn more' link. Below this, there's a section 'Authorized API clients' with a subtext: 'The following API client domains are registered with Google and authorized'. A form is present with two main fields: 'Client Name' and 'One or More API Scopes'. The 'Client Name' field has an example: 'www.example.com'. The 'One or More API Scopes' field has an example: 'http://www.google.com/calendar/feeds/' and a note '(comma-delimited)'. There's an 'Authorize' button. Below the form, the client ID '100327686628199041742' and the scope 'https://www.googleapis.com/auth/drive.file' are displayed.

3. In the field **One or More API Scopes**, key in the following value.

The screenshot shows a text input field with the value 'https://www.googleapis.com/auth/drive.file' entered.

4. Click **Authorize**.

#### Configure Joget Workflow App

1. From the downloaded JSON file earlier, copy the "client\_email" and "private\_key" values.
2. In your Joget app, select Google API Configuration in **Properties & Export -> Set Default Plugin Properties**
3. Key in the value of "client\_email" value into OAuth 2.0 Client ID, and value of "private\_key" into OAuth 2.0 Client Secret, then **Submit**.

Design App:

HR Expenses Claim App

Version 1 Published

1 Forms & UI

2 Processes

3 Properties & Export

All Apps

Add Environment Variable

Add Message

Set Plugin Default Properties

Export

Plugin Configuration

Configure Google APIs

OAuth 2.0 Client ID

<https://github.com/googleads/googleads-dotnet-lib/wiki/How-to-create-OAuth2-client-id-and-secret>

jogetsample@jogetsample.iam.gserviceaccount.com

OAuth 2.0 Client Secret

<https://github.com/googleads/googleads-dotnet-lib/wiki/How-to-create-OAuth2-client-id-and-secret>

-----BEGIN PRIVATE KEY-----\nMIIEvQIBADANBgkqhkiG9w

Google Drive Form Element

1. With the steps above done, you are now ready to use the **Google Drive File** form element.
2. Just configure on which user's Google Drive you want to store to and name the folder accordingly.

joget

FORM BUILDER - HR EXPENSE CLAIM FORM (V1)

Design Form

Design and Layout

Popup Select box

Calculation Field

Captcha

Signature

Time Picker

AJAX Subform

Image Upload

Rich Text Editor

Advanced Grid

Form Grid

List Grid

Google

Google Drive File

Property Editor

Edit File Upload

Edit File Upload > Advanced Options

ID \*

attachment

Label \*

Attachment

Configure Google Drive

Google Drive Email \*

buzz@samplejoget.com

Google Drive Base Folder

Joget Google Drive

3. Files uploaded will follow the same folder structure as the native **File Upload** element. Here's the sample screenshot in Google Drive.

Drive

My Drive > Joget Google Drive > hr\_expense\_claim > 171\_hr\_expense\_process1

NEW

My Drive

Shared with me

Google Photos

Recent

Starred

Name	Owner	Last modified	File size
script.txt	me	5:00 PM	2 KB
tts.m4a	me	5:00 PM	2 MB