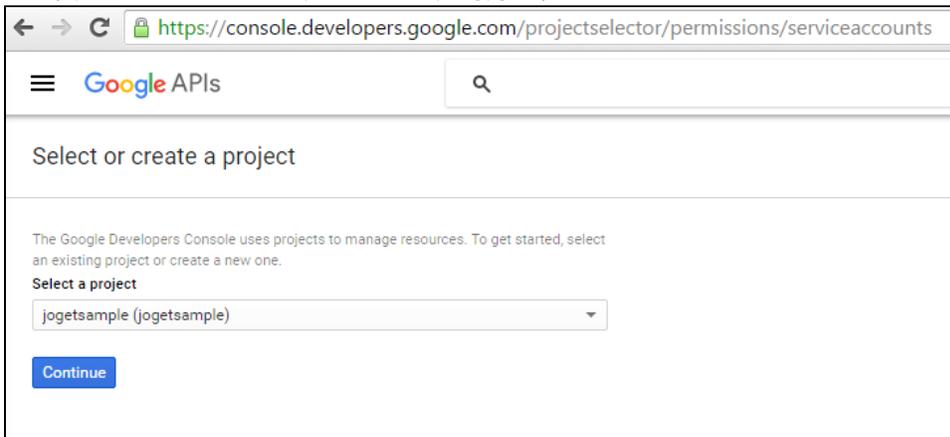


# Google API Configuration

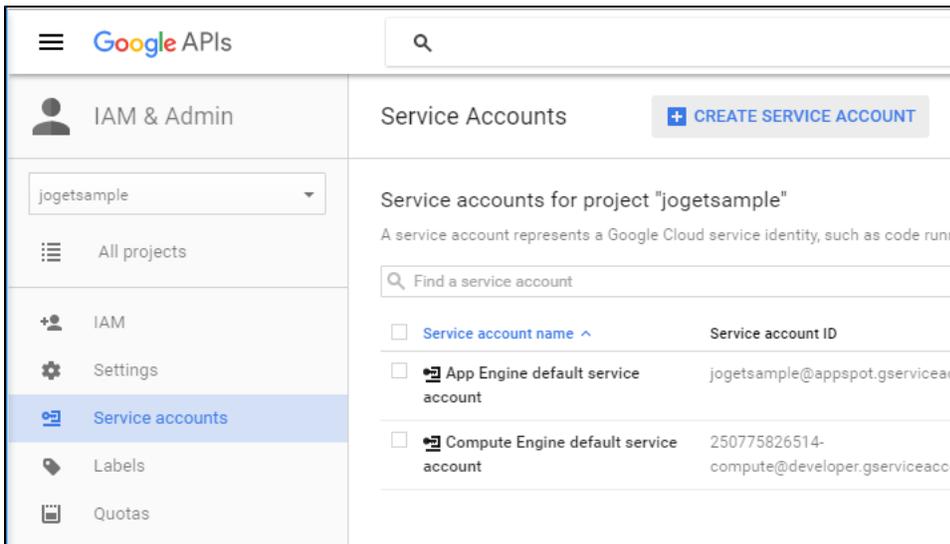
- Create service account for the project
- Enable Google Drive API for the project
- Delegate Google Drive access to the project service account
- Configure Joget Workflow App
- Google Drive Form Element

## 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. joetsample).



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



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**.

Service Accounts + CREATE SERVICE ACCOUNT 🗑 DELETE + PERMISSI

Service account

A service account re

Find a service ac

Service account

App Engine account

Compute En account

**Create service account**

Service account name ?

joetsample

Service account ID

joetsample @joetsample.iam.gserviceaccount.com ↻

**Furnish a new private key**  
Downloads a file that contains the private key. Store the file securely because this key can't be recovered if lost.

**Key type**

JSON  
Recommended

P12  
For backward compatibility with code using the P12 format

**Enable Google Apps Domain-wide Delegation**  
Grants a client access to all users' data on a Google Apps domain without manual authorization on their part. [Learn more](#)

**i** 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**

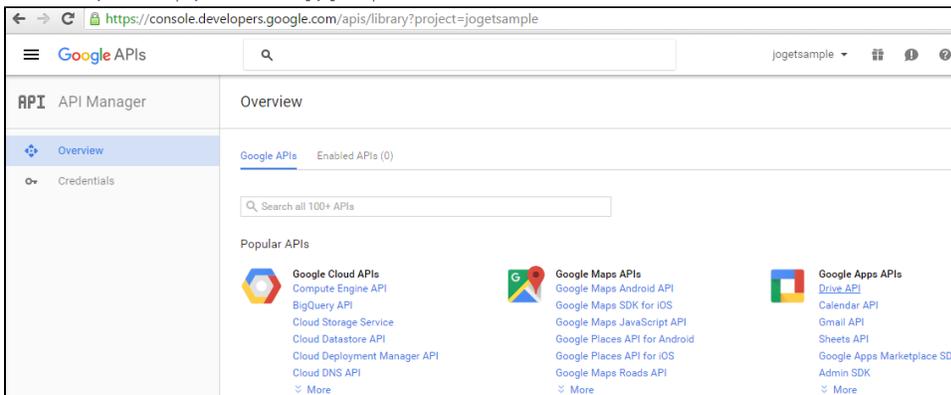
Joget Workflow

**Create** **Configure consent screen** **Cancel**

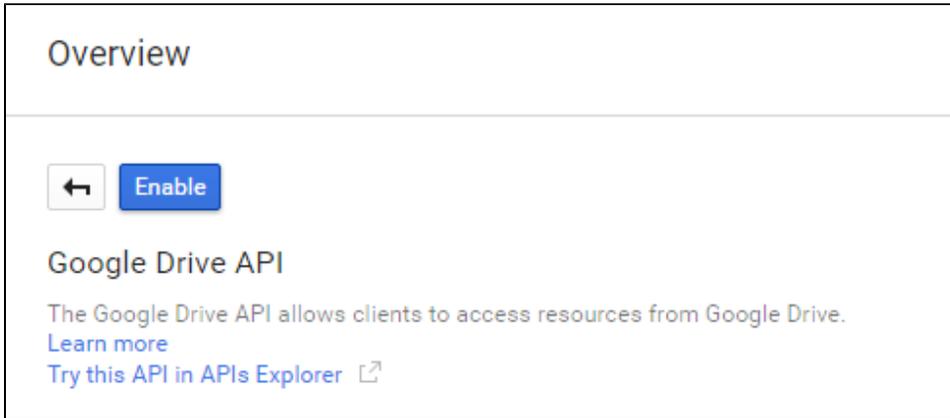
5. A .json file will be downloaded. Open up the file in your favorite text editor.

## 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).



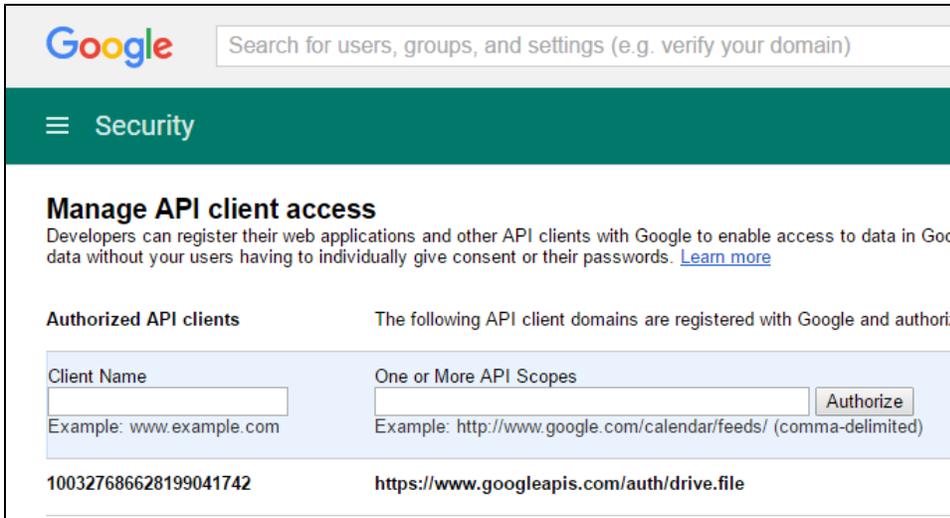
3. Click on **Google Drive API**.



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.

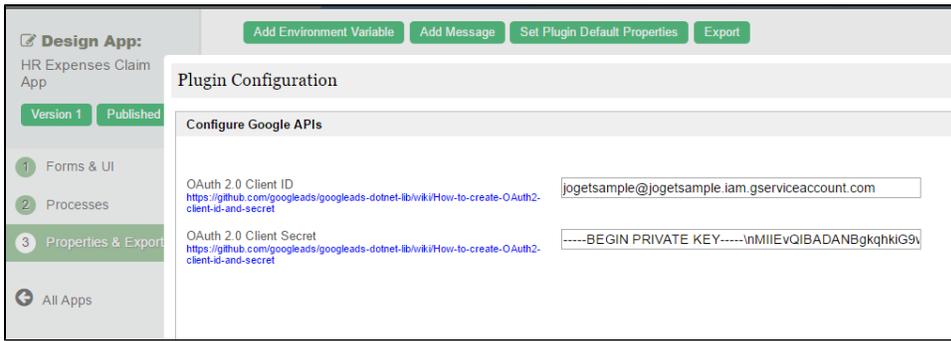


3. In the field **One or More API** Scopes, key in the following value.

4. Click **Authorize**.

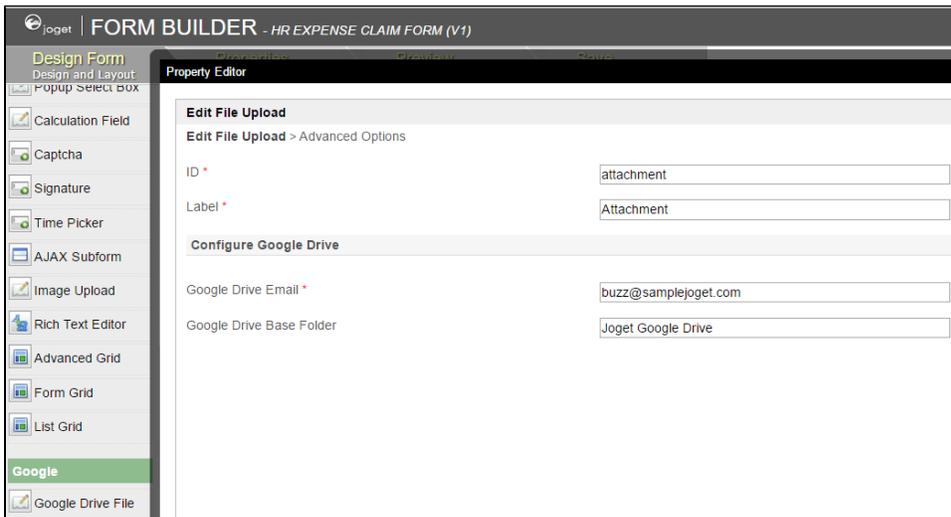
### Configure Joget Workflow App

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**.



## 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.



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

