Google Sheets Connector for ABBYY Vantage

"Seamlessly Integrate ABBYY Vantage with Google Sheets for Automated Data Processing and Enhanced Workflow Efficiency.".

By: Deepak Goyal (Partner Innovation and Enablement Specialist At ABBYY)

Table of Contents

About Google Sheets Connector for ABBYY Vantage

The Google Sheets Connector for ABBYY Vantage is designed to streamline your data processing and enhance your workflow efficiency. ABBYY Vantage, a leading platform for intelligent document processing, seamlessly integrates with Google Sheets, allowing you to automatically extract, process, and manage data directly within your spreadsheets. Whether you're handling invoices, receipts, forms, or other documents, this connector enables you to harness the power of ABBYY Vantage's advanced AI and machine learning capabilities, bringing automation and accuracy to your data management tasks.

Security is a top priority with this integration, utilizing the OAuth 2.0 protocol for secure authentication with the Google Sheets API. This ensures that your data remains protected and accessible only to authorized users.

Ideal for businesses of all sizes, this integration simplifies complex processes, reduces manual effort, and ensures data integrity across your operations. Explore the full potential of ABBYY Vantage with the convenience of Google Sheets, and transform the way you handle your document workflows.

System Requirements and Limitations

To ensure optimal performance and compatibility, please ensure that your system meets the following requirements:

Operating System:

- Windows 10 or later
- macOS 10.15 (Catalina) or later
- Linux distributions supporting Google Chrome or Mozilla Firefox

Web Browser:

- Google Chrome (latest version recommended)
- Mozilla Firefox (latest version recommended)
- Microsoft Edge (latest version recommended)

- Safari (latest version recommended)

Hardware:

- Processor: Intel Core i3 or equivalent
- Memory: 4 GB RAM (8 GB recommended)
- Storage: 200 MB of available disk space for installation
- Internet Connection: Broadband or higher

Software:

- Google Account with access to Google Sheets
- ABBYY Vantage account

Additional Requirements:

- OAuth 2.0 authentication for secure access to Google Sheets API
- Up-to-date web browser to ensure compatibility with the Google Sheets and ABBYY Vantage interfaces.

By meeting these system requirements, you can ensure a smooth and efficient integration experience, maximizing the benefits of the Google Sheets Connector for ABBYY Vantage.

Installing the Connector

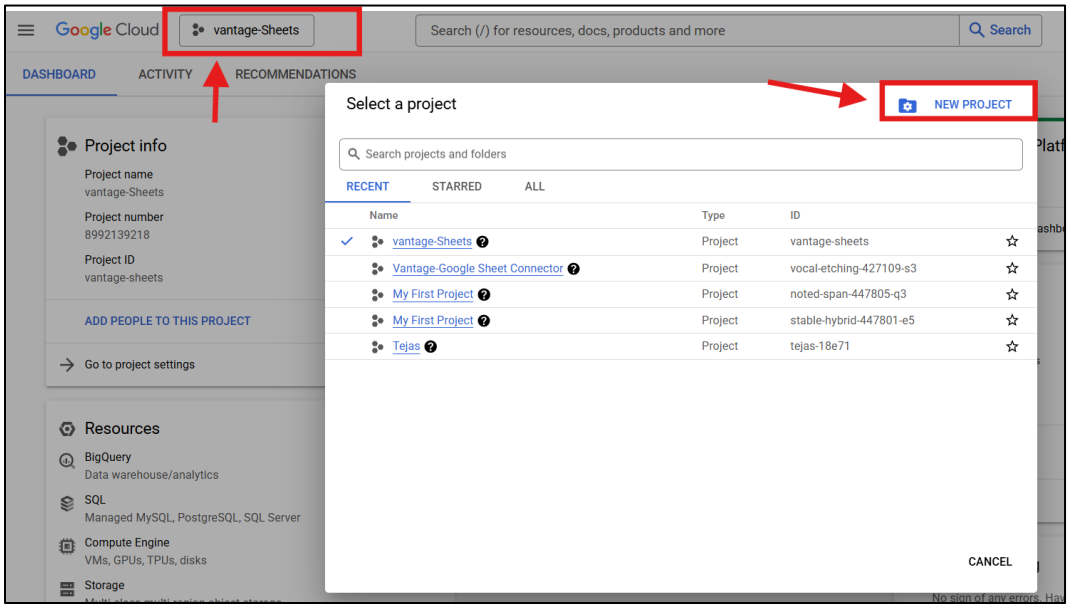
ABBYY Vantage Integration with Google Sheets is a script that runs in an Output or Custom Activity of a Process Skill. The current version of ABBYY Vantage Integration with Google Sheets is configured by modifying the script (see Configuring the Connector below).

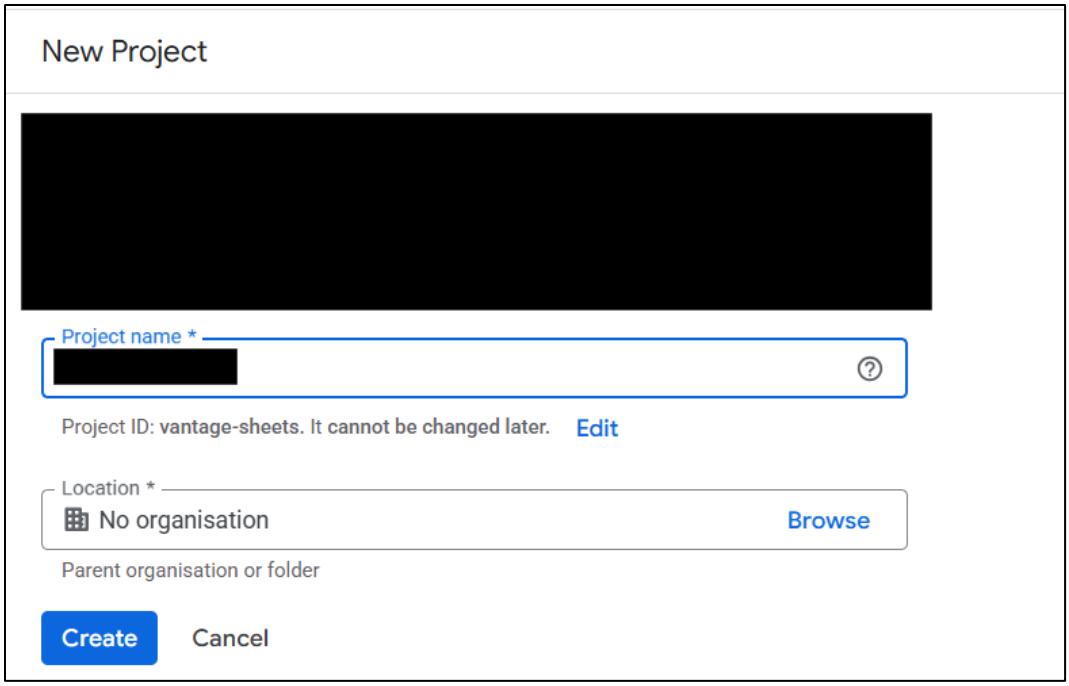
Setting Up Google Project

Before setting up the connector, you need to set up a Google Project to enable the Google Sheets API. Follow these steps:

1. **Create a Google Cloud Project:**

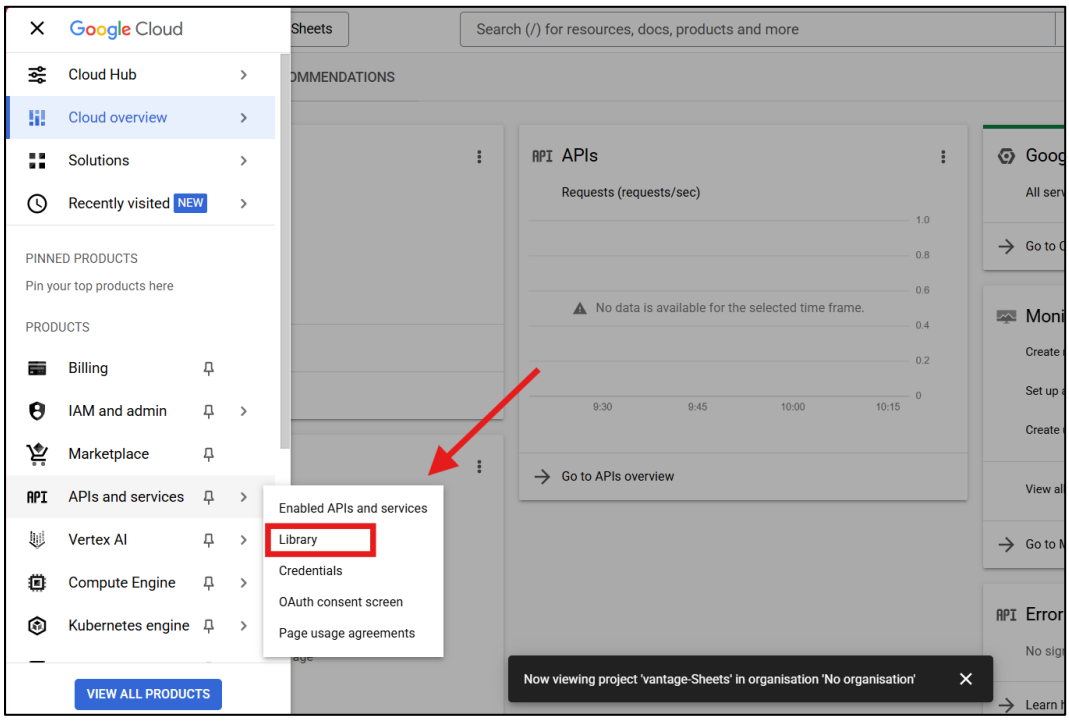
   - Go to the Google Cloud Console.
   - Click on the project drop-down menu at the top of the page and select "**New Project**".
   - Enter a name for your project and click "**Create**".



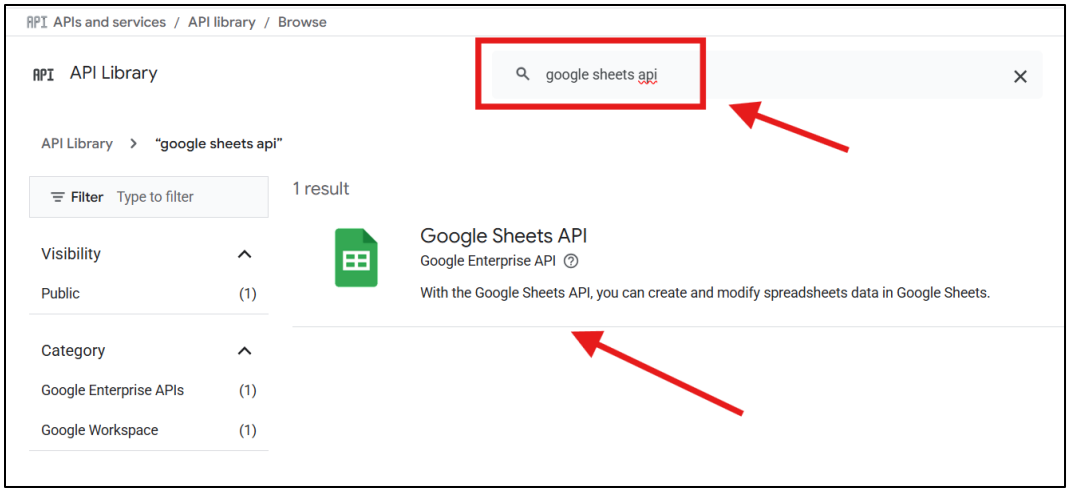


2. **Enable the Google Sheets API:**

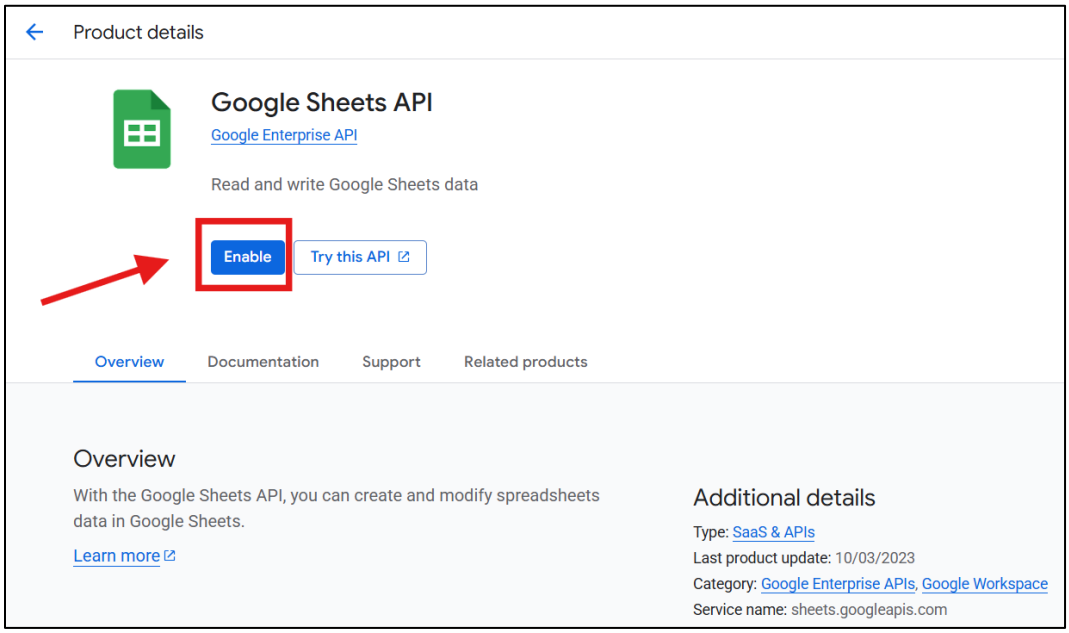
- In the Google Cloud Console, navigate to the "**APIs & Services" > "Library**".
- Search for **"Google Sheets API"** and click on it.
- Click the **"Enable"** button.



Step 1



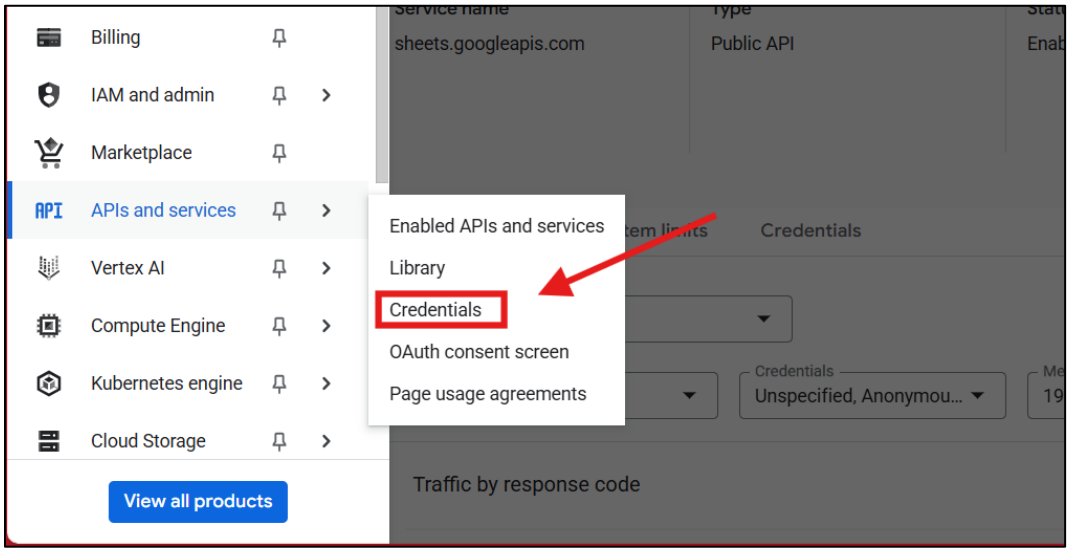
Step 2



Step 3

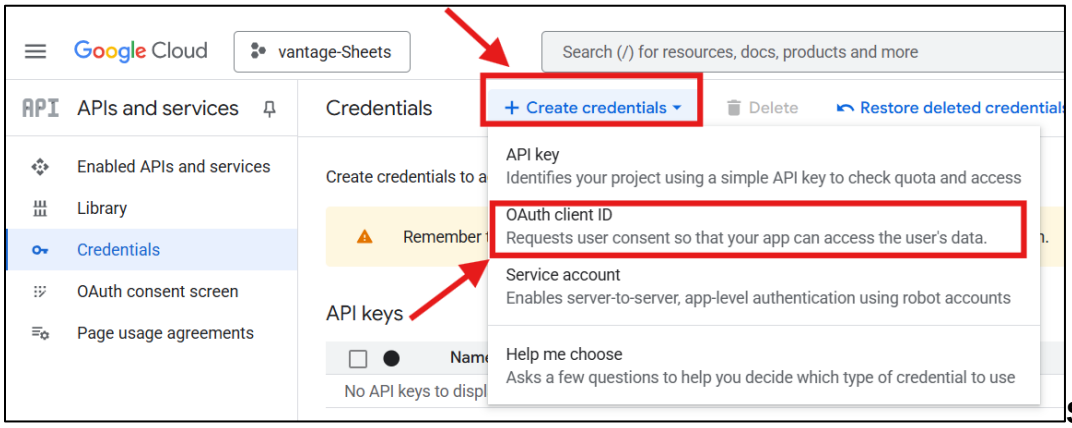
3. **Create OAuth 2.0 Credentials and Choose Scopes:**

- Navigate to "**APIs & Services**" > "**Credentials**".



Step 1

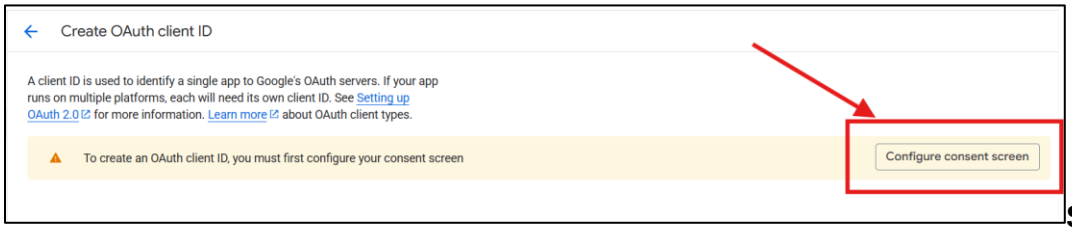
- Click on "**Create Credentials**" and select "**OAuth 2.0 Client ID**".



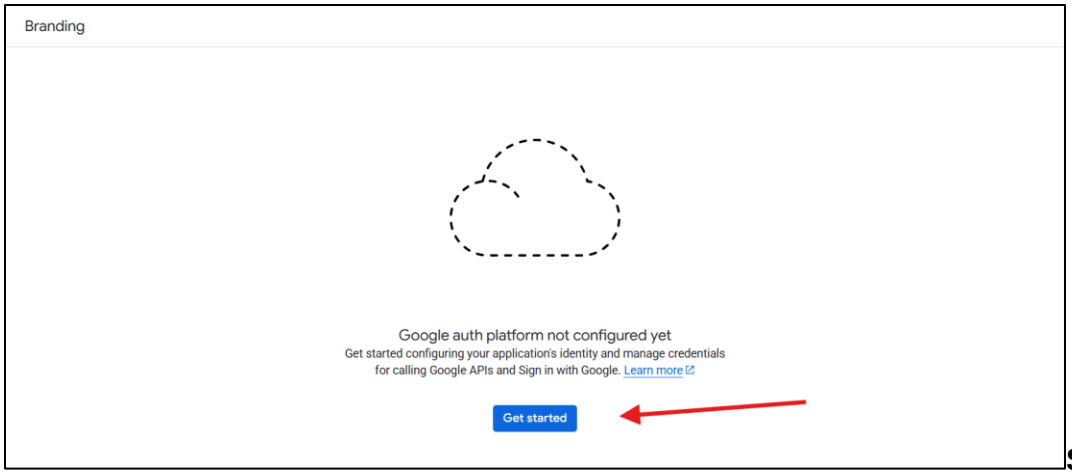
Step 2

- Configure the consent screen with the necessary information.

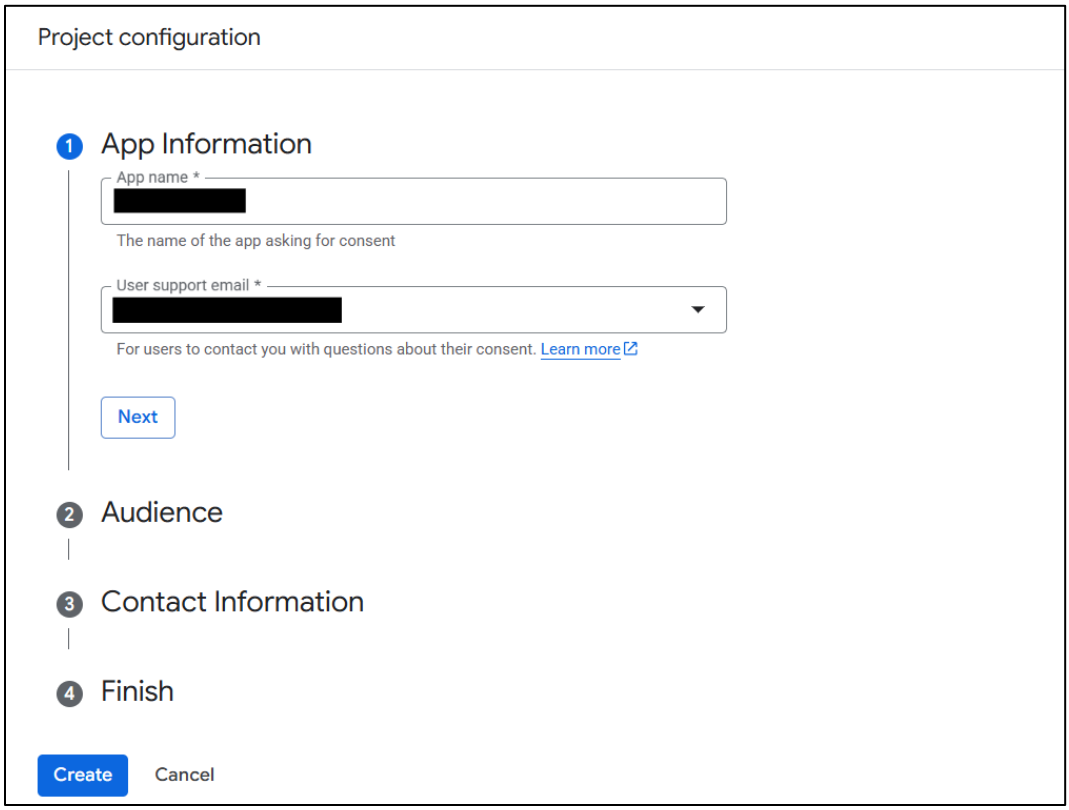
  # Refer to latest Google documentation to Configure the OAuth consent screen and choose scopes- [Configure the OAuth consent screen and choose scopes | Google Workspace | Google for Developers]



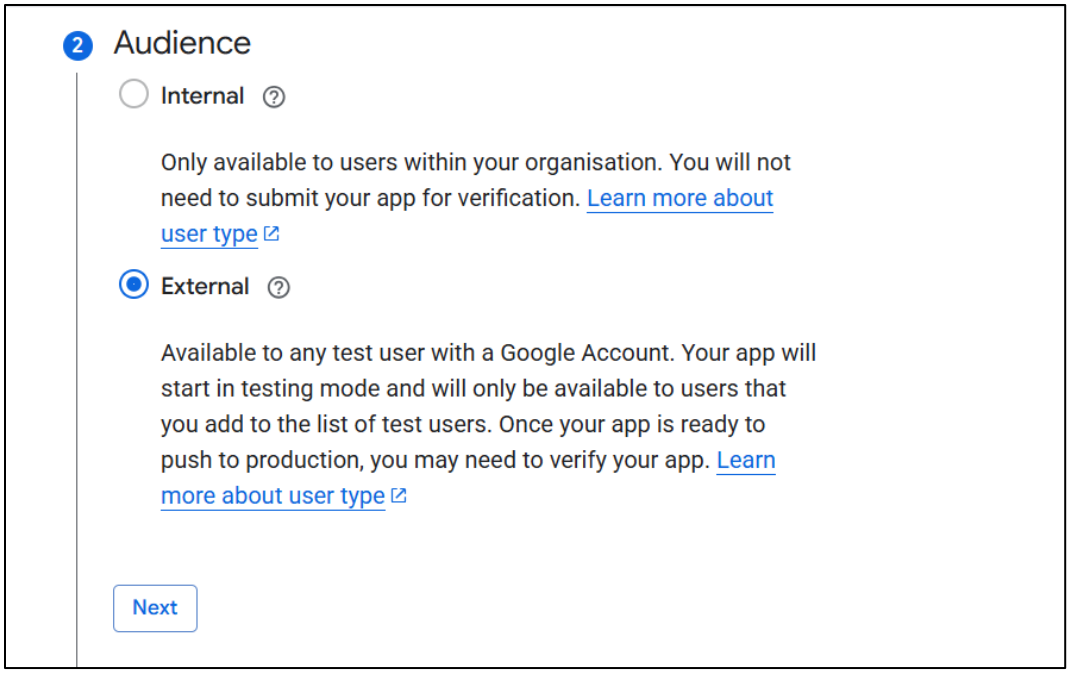
Step 3



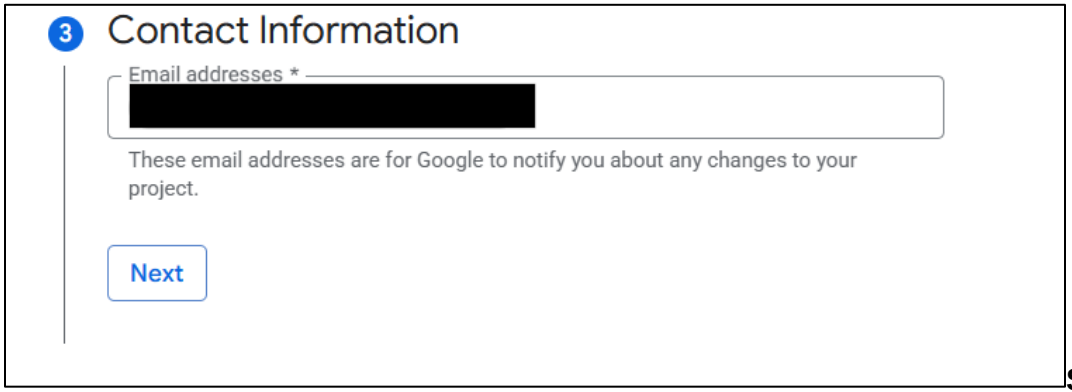
Step 4



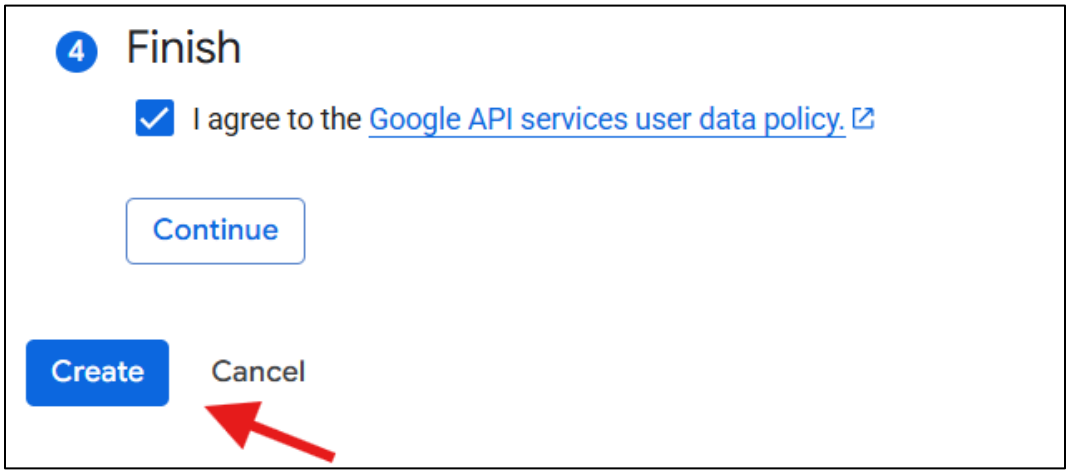
Step 5



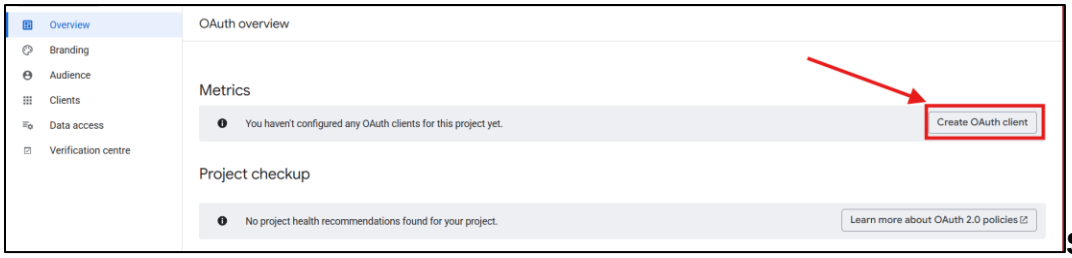
Step 6



Step 7

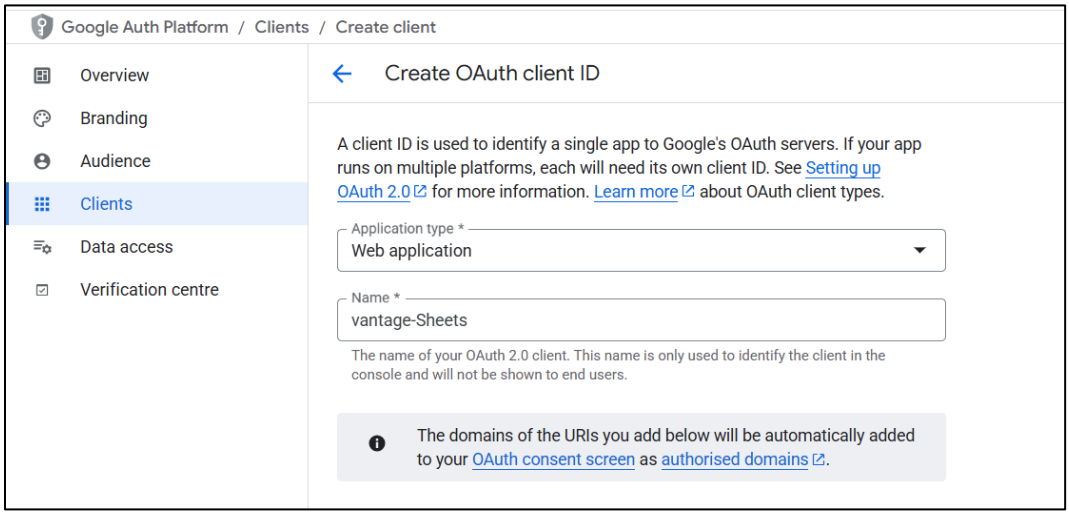


Step 8



Step 9

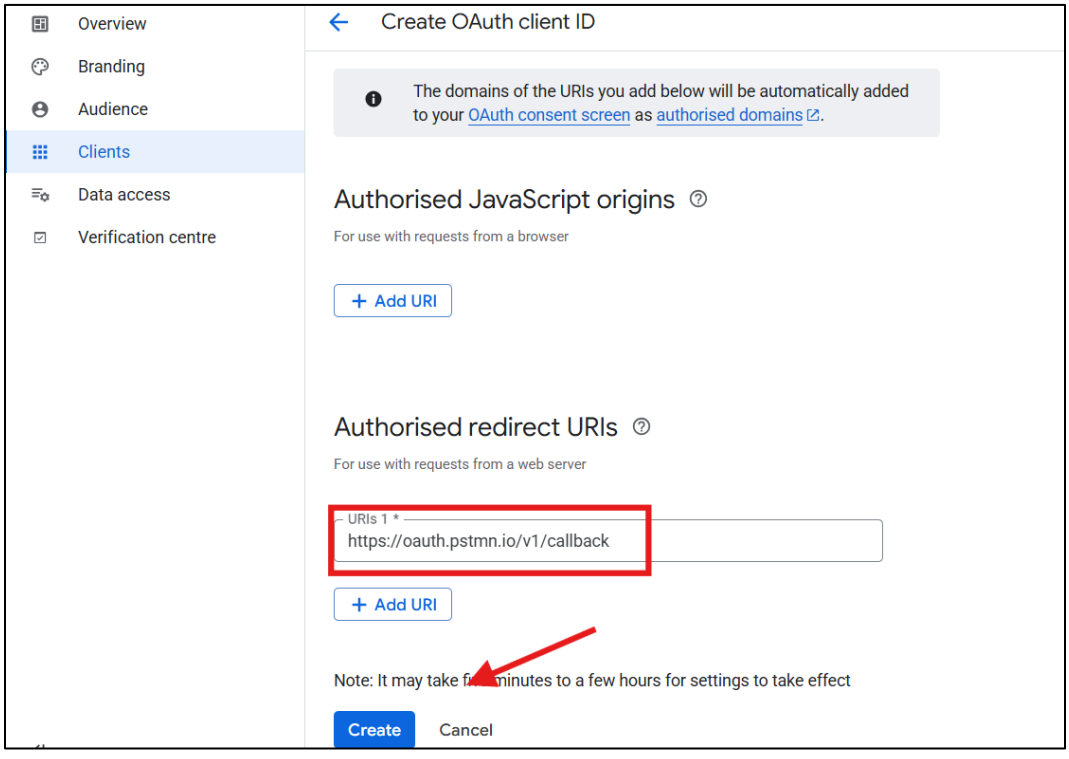
- Select "**Application type**" as "**Web application**".



Step 10

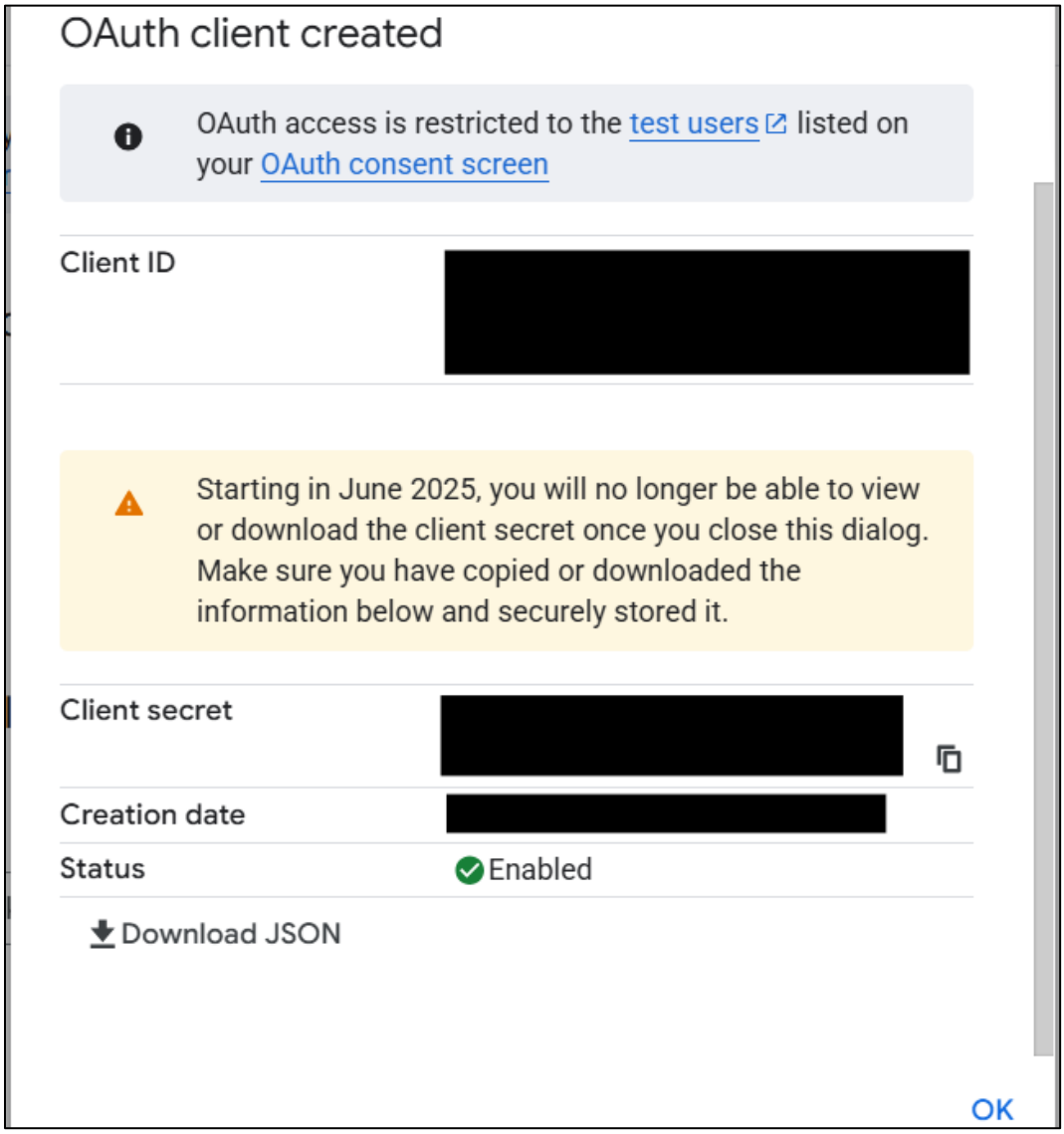
 # App name should be the same as the Project name else you will get an error

- Enter a name for your credentials and add your authorized redirect URIs.



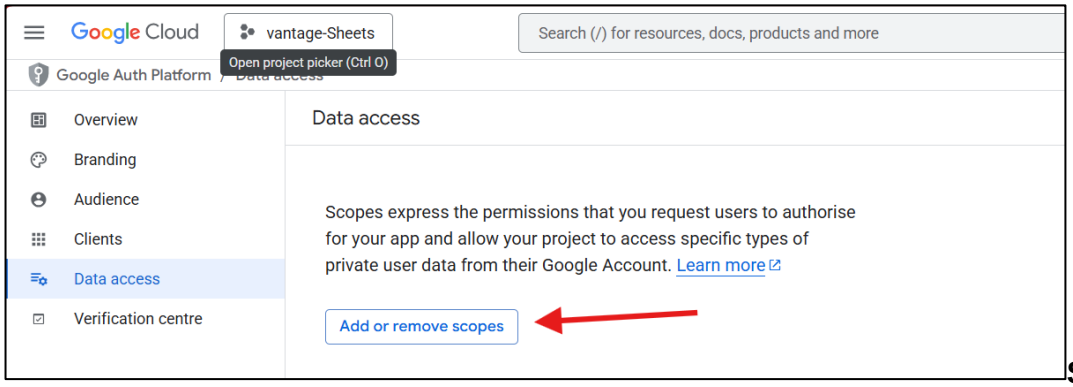
Step 11

- Click "**Create**" and note down the client ID and client secret.



Step 12

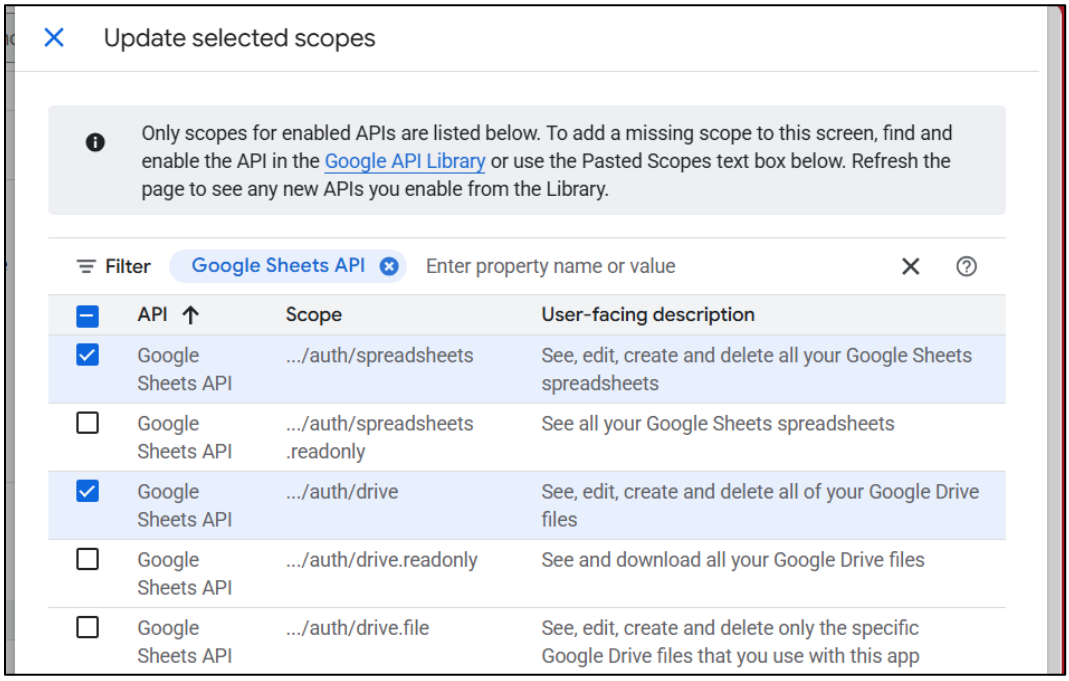
- If you're creating an app for use outside of your Google Workspace organization, click **Add or Remove Scopes.**



Step 13

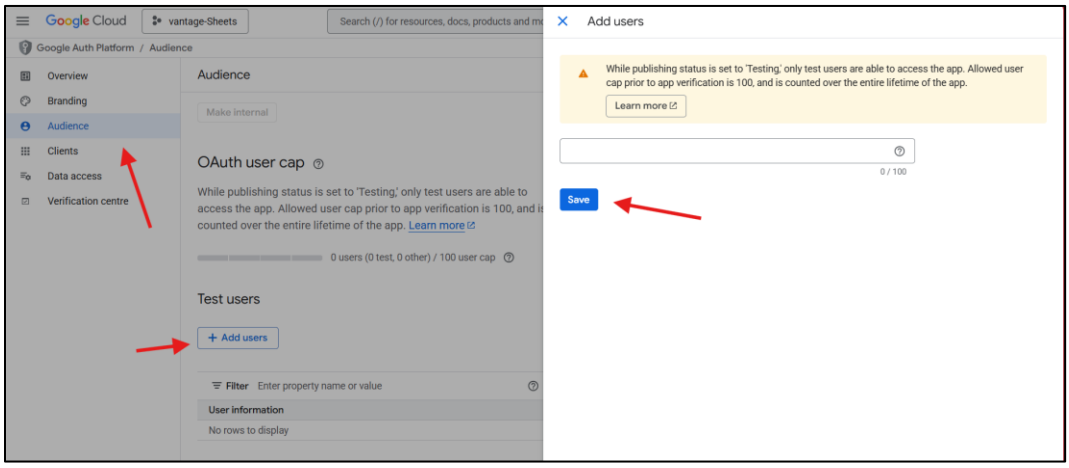
- After selecting the scopes required by your app, click **Save and Continue**.

Select '**./auth/spreadsheets'** and select '**./auth/drive'** - this will give all the permissions required.



Step 14

- If you selected **External** for user type, add test users:
- Under Test users, click **Add users**.



Step 15

Setting Up OAuth 2.0 for Script

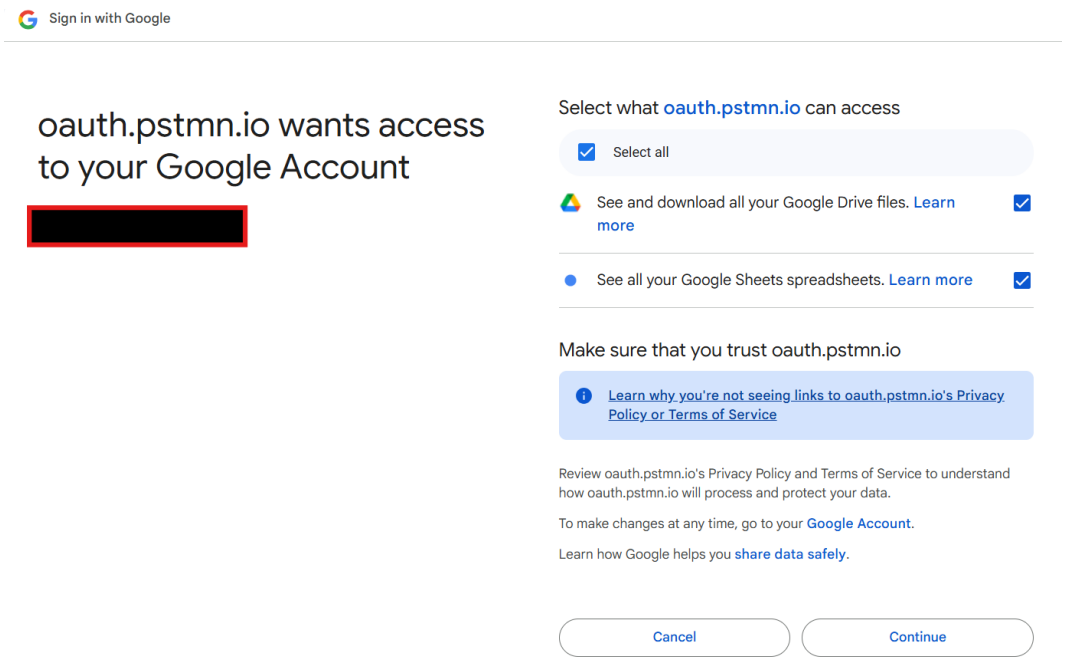
To set up OAuth 2.0, follow these steps using Postman:

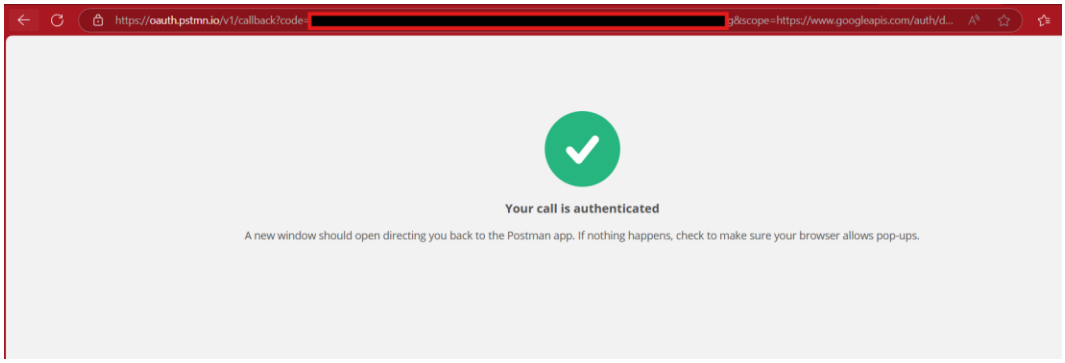
1. **Generate Authorization Code:**

   - Edit Below URL replacing **YOUR_CLIENT_ID** and **YOUR_REDIRECT_URI** with your actual client ID and redirect URI.

     ```
     https://accounts.google.com/o/oauth2/auth?client_id=YOUR_CLIENT_ID&redirect_uri=YOUR_REDIRECT_URI&response_type=code&scope=https://www.googleapis.com/auth/spreadsheets.readonly https://www.googleapis.com/auth/drive.readonly
     ```

   - Open the URL in a web browser, authorize the application, and copy the authorization code from the URL.





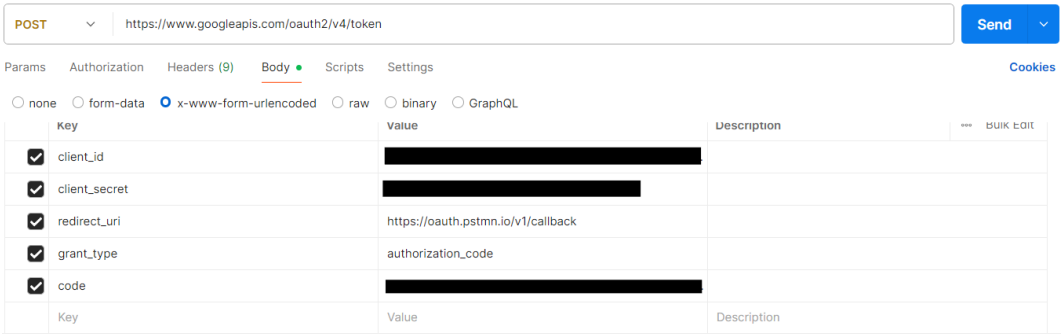
   - copy the authorization code from the URL.

2. **Generate Refresh Token:**

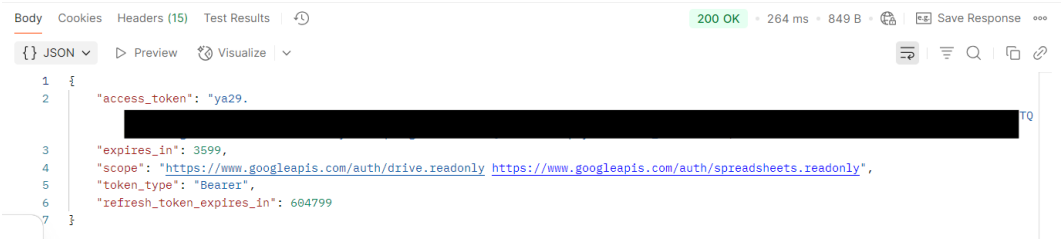
   - In Postman, create a new request.
   - Set the request type to "POST" and the URL to:

```
curl \
  -d "client_id={your client ID}" \
  -d "client_secret={your client secret}" \
  -d "redirect_uri={your redirect uri}" \
  -d "grant_type=authorization_code" \
  -d "code={retrieved your authorization code}" \
```

   - In the body, select "x-www-form-urlencoded" and add the following parameters:



   - Send the request and note down the **refresh_token** from the response.



3. **Configure the Refresh Token in Your Script:**

   - Modify the ABBYY Vantage Integration script to include your refresh token. This token will be used to generate access tokens when needed.

By following these steps, you ensure that your script can securely authenticate and interact with Google Sheets using OAuth 2.0.

After completing these steps, you can proceed with configuring the connector as described in the next sections.

Configuring The Connector in ABBYY Vantage

This script retrieves table data from a Vantage document and updates a Google Sheet with the data, including handling header rows.

Configuration

Environment Variables:

- **GoogleSheet_CLIENT_ID**: Client ID for Google Sheets API.
- **GoogleSheet_CLIENT_SECRET**: Client Secret for Google Sheets API.
- **GoogleSheet_REFRESH_TOKEN**: Refresh Token for Google Sheets API.
- **GoogleSheet_SHEET_ID**: The ID of the Google Sheet to update.

Step-by-Step Explanation

1. **Set Up Configuration Parameters -** Define configuration parameters that fetch secrets from the environment and the Google Sheet ID.

2. **Function to Get Access Token -** getAccessToken() **-** This function retrieves an OAuth 2.0 access token using the refresh token.

3. **Function to Read the First Row of the Google Sheet -** readGoogleSheet(accessToken) **-** This function retrieves the first row of the Google Sheet to check for headers.

4. **Function to Retrieve Table Data and Create a DataFrame -** getTableData() - This function retrieves table data from the Vantage document and formats it for Google Sheets.

5. **Function to Update Google Sheet -** updateGoogleSheet(accessToken, tableData) **-** This function updates the Google Sheet with the table data, including headers if necessary.

6. **Main Function to Orchestrate the Process -** main() **-** The main function orchestrates the entire process by fetching the access token, checking the Google Sheet for headers, retrieving table data, and updating the Google Sheet.

Summary

This script effectively retrieves table data from a Vantage document, including handling invoice numbers and table headers, and updates a specified Google Sheet with this data. The main function ensures headers are added if they are missing from the Google Sheet. The process involves fetching an access token, reading the Google Sheet to check for headers, retrieving table data, and updating the Google Sheet with the formatted data.