

# Access Google Sheets Data Using a Service Account

Accessing Google Sheets data programmatically can be essential for server-side applications that need to read or manipulate spreadsheet data without human interaction. Whether you're automating reports, synchronizing data, or managing inventory systems, accessing Google Sheets through a Service Account provides a secure and efficient solution for backend environments.

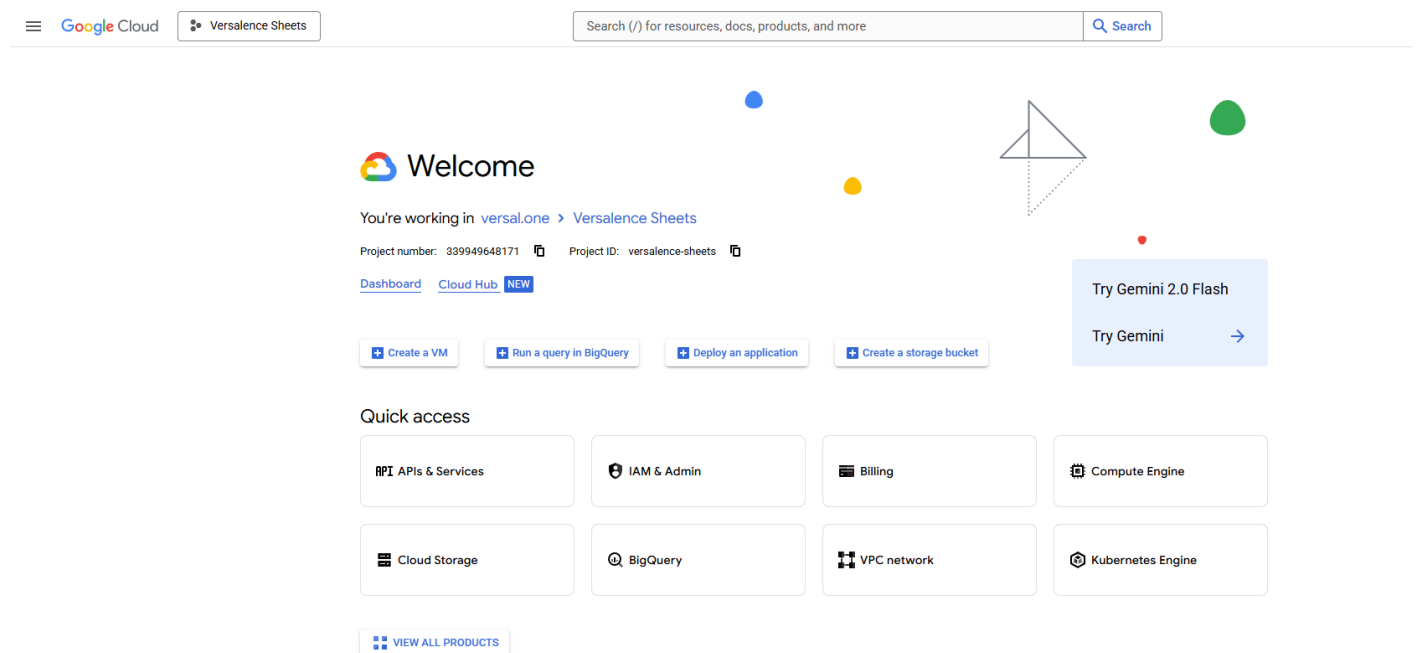
This tutorial will guide you through accessing a Google Sheet's data using a Service Account, which is ideal for server environments.

## Prerequisites

Before you start, ensure you have the following:

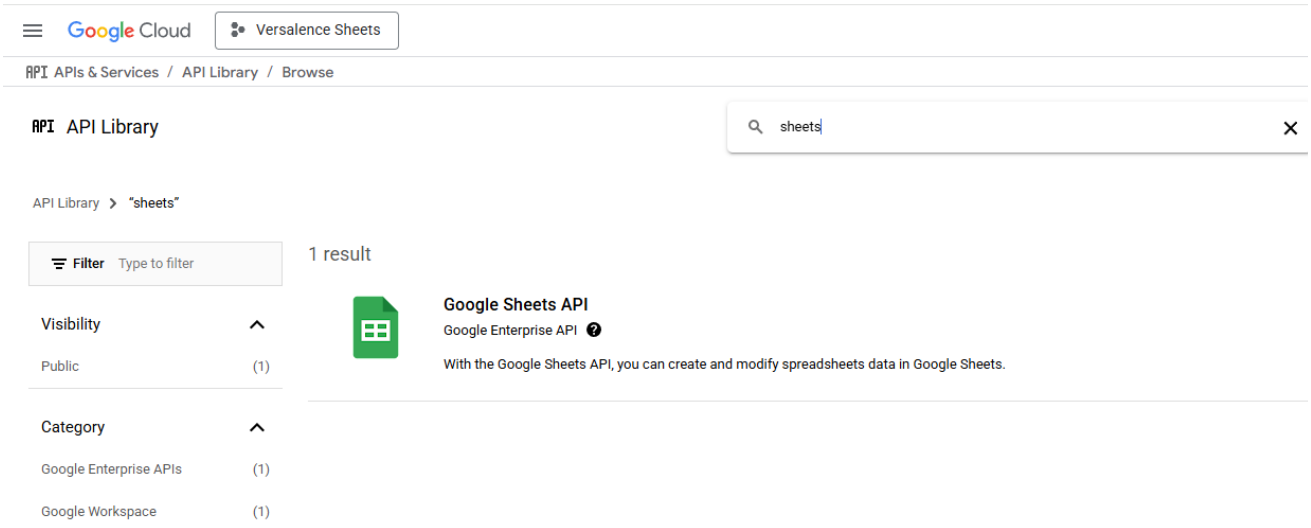
- A **Google Account**: Access to the Google Cloud Console.
- An **Existing Google Spreadsheet**: A Google Sheet that you want to access.

## Step 1: Enable the Google Sheets API



To access Google Sheets programmatically, you first need to enable the Google Sheets API in your Google Cloud project.

1. **Go to the [Google Cloud Console](#).**
2. **Create a New Project** (if you don't have one):
  - Click the project dropdown at the top of the page.
  - Click **New Project**.
  - Enter a project name (e.g., "Google Sheets Node.js Access").
  - Click **Create**.
3. **Select Your Project:**
  - Ensure your new project is selected in the project dropdown.



4. **Enable the Google Sheets API:**
  - Navigate to **APIs & Services > Dashboard**.
  - Click **Enable APIs and Services**.
  - Search for **Google Sheets API**.
  - Click on **Google Sheets API** and then click **Enable**.

With the API enabled, your project can now interact with Google Sheets data programmatically, preparing for the next steps.

## Step 2: Create a Service Account and Generate a Key File

A Service Account allows server-side applications to authenticate without user interaction.

Google Cloud | Versalence Sheets | Search (/) for resources, docs, products, and more

APIs & Services | Credentials | + Create credentials | Delete | Restore deleted credentials

Remember to configure the OAuth consent screen with information about your application. [Configure consent screen](#)

Create credentials to access your enabled APIs. [Learn more](#)

**API Keys**

<input type="checkbox"/>	Name	Creation date ↓	Restrictions	Actions
<input type="checkbox"/>	API key 1	Apr 12, 2025	—	<a href="#">Show key</a> ⋮

**OAuth 2.0 Client IDs**

<input type="checkbox"/>	Name	Creation date ↓	Type	Client ID	Actions
No OAuth clients to display					

**Service Accounts** [Manage service accounts](#)

<input type="checkbox"/>	Email	Name ↑	Actions
<input type="checkbox"/>	versalence-sheets-service-acc@versalence-sheets.iam.gserviceaccount.com	Versalence Sheets Service Acc	<a href="#">Edit</a> <a href="#">Delete</a>

### 1. Navigate to Service Accounts:

- Go to **APIs & Services > Credentials**.
- Click on **Create Credentials** and select **Service Account**.

### 2. Create a Service Account:

- **Service Account Name:** Enter a name (e.g., “sheets-access-account”).
- **Service Account ID:** This will auto-populate based on the name.
- **Description:** Optional.
- Click **Create and Continue**.

### 3. Set Service Account Permissions:

- For this tutorial, you don’t need to grant any specific roles.
- Click **Continue**.

### 4. Grant Users Access to This Service Account:

- Skip this step by clicking **Done**.

### 5. Generate a JSON Key File:

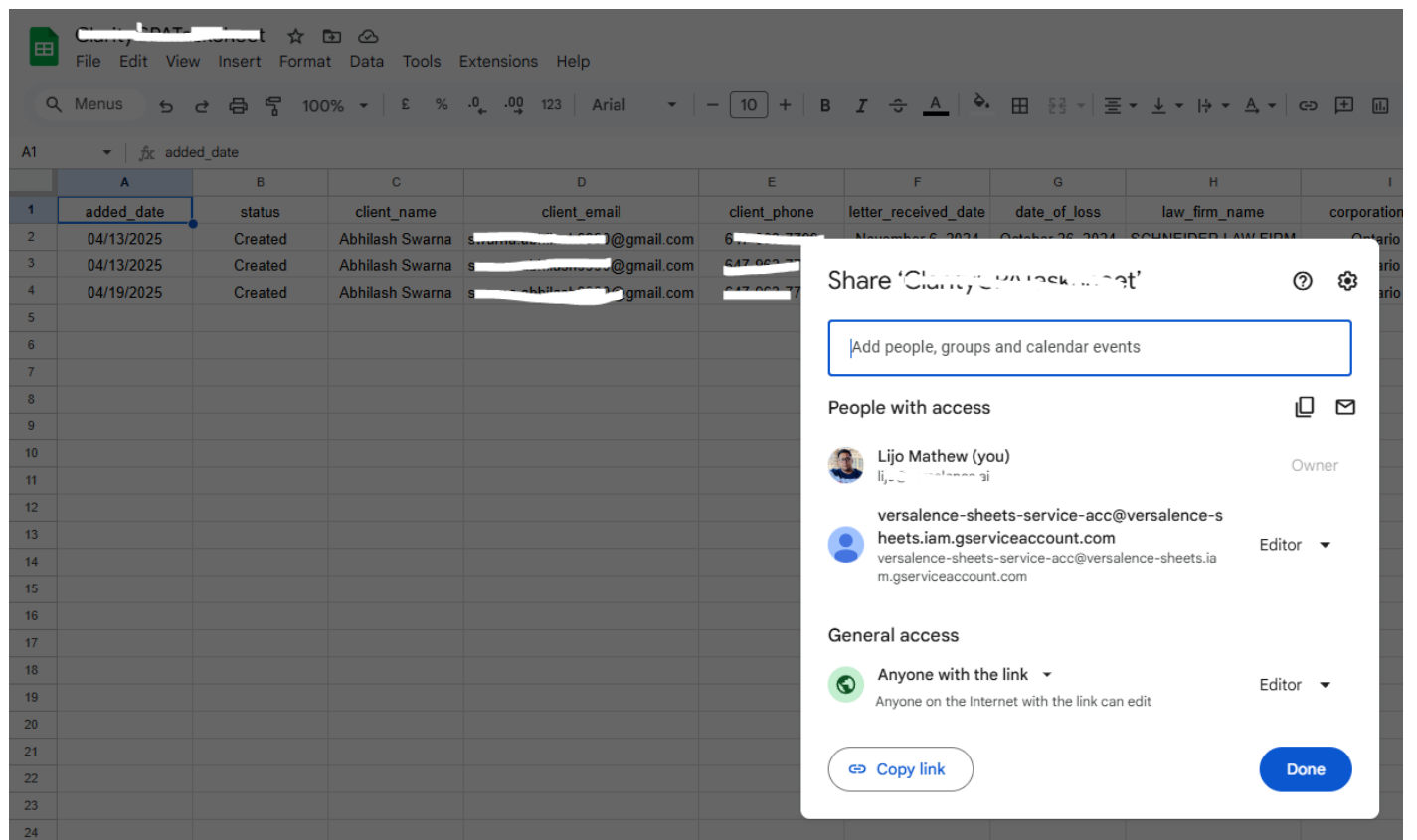
- You should now see your service account listed.
- Click on the **Service Account** you just created.
- Navigate to the **Keys** tab.
- Click **Add Key > Create New Key**.
- Choose **JSON** as the key type.
- Click **Create**.
- A JSON key file will be downloaded to your computer (e.g., `service-account-key.json`).

### 6. Secure the Key File:

**Important:** Do not commit this file to version control. It’s sensitive information.

## Step 3: Share the Spreadsheet with the Service Account Email

The Service Account acts like a virtual user. You need to share the spreadsheet with the Service Account's email address.



### 1. Get the Service Account Email:

- In the **Details** tab of your Service Account, locate the **Email** field.
- It will look like: `your-service-account-name@your-project-id.iam.gserviceaccount.com`.

### 2. Share the Spreadsheet:

- Open your Google Sheet in a browser.
- Click the **Share** button.
- Enter the Service Account email.
- Set the permission to **Editor** (or **Viewer** if you only need read access).
- Click **Send**.

Revision #2

Created 23 April 2025 15:37:47 by Admin

Updated 23 April 2025 15:47:26 by Admin