Laravel Integration With Google Sheets API

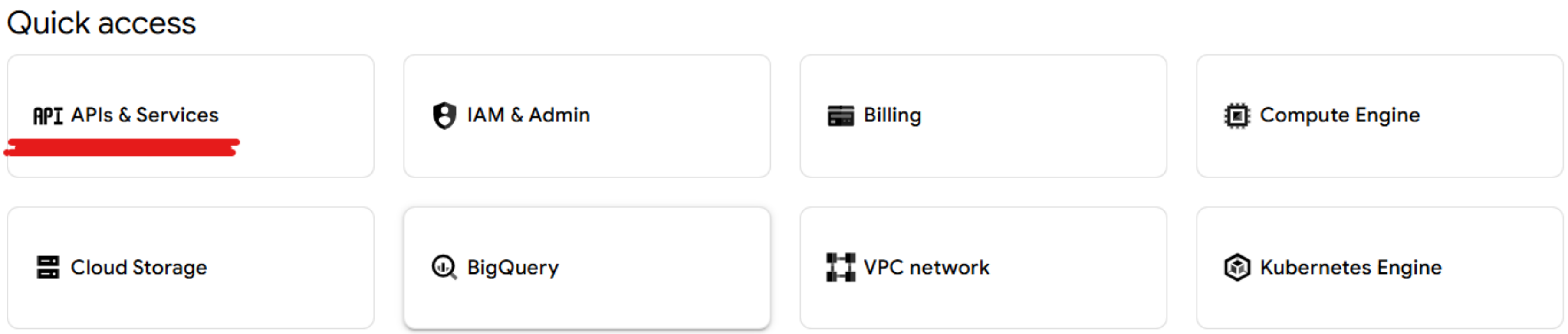
Setting Up Google Cloud

Step 1: Google Cloud Console

Visit the Google Cloud Console to start setting up your project. Click

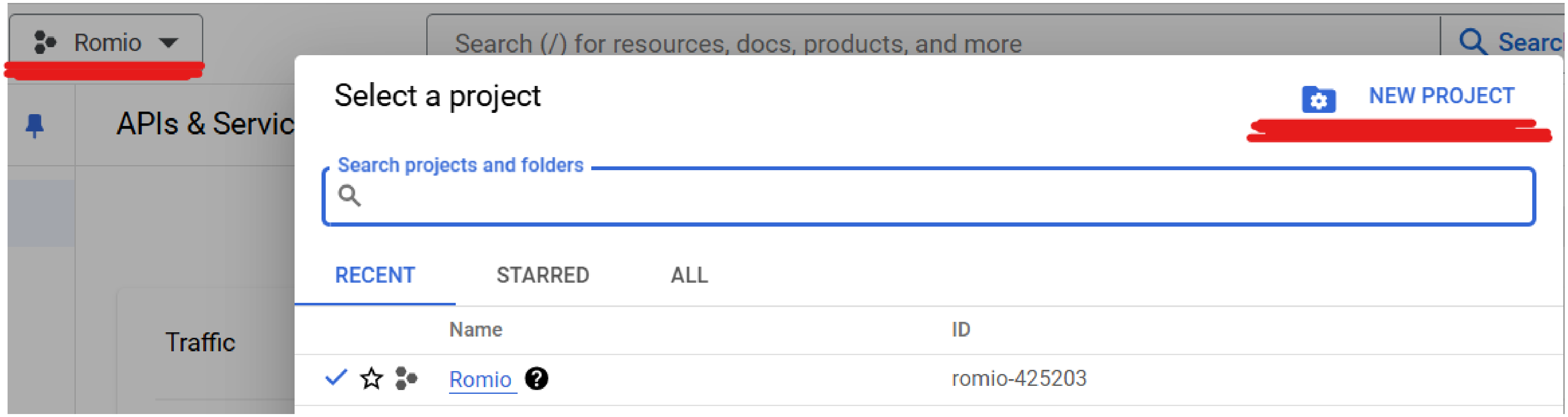
!open this

Create a new project in the Google Cloud Console.



Step 2: Create a New Project

!Create New Project


Romio
Search (/) for resources, docs, products, and more
Searc
APIs & Servic
Traffic
NEW PROJECT
Search projects and folders
RECENT
STARRED
ALL

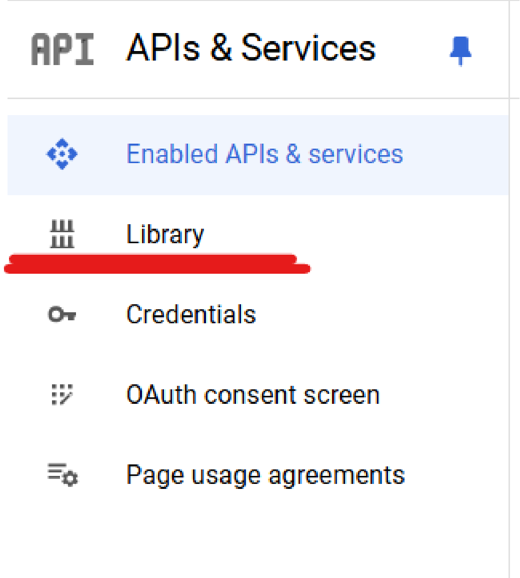
Select a project

Name	ID
Romio	romio-425203

Step 3: Go to Library

Navigate to the library section of your Google Cloud project.

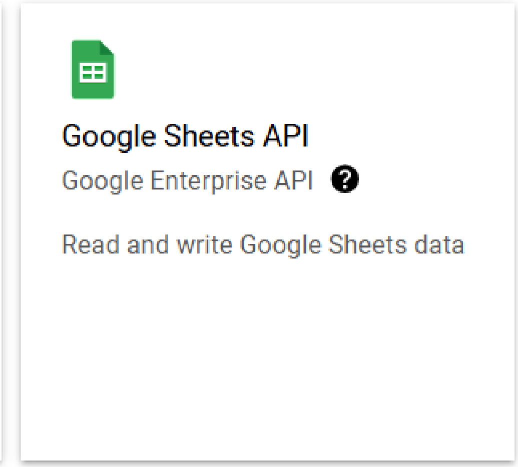
!Go to Library



Step 4: Enable APIs

Search for the required APIs and enable them for your project.

!Choose this one

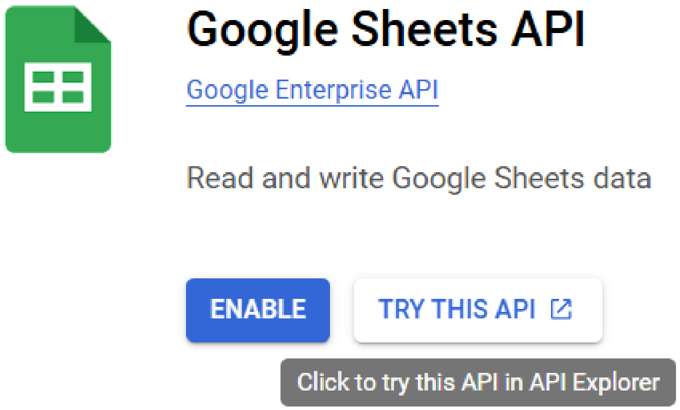


Google Sheets API

Google Enterprise API

Read and write Google Sheets data

!Enable APIs



Google Sheets API

Google Enterprise API

Read and write Google Sheets data

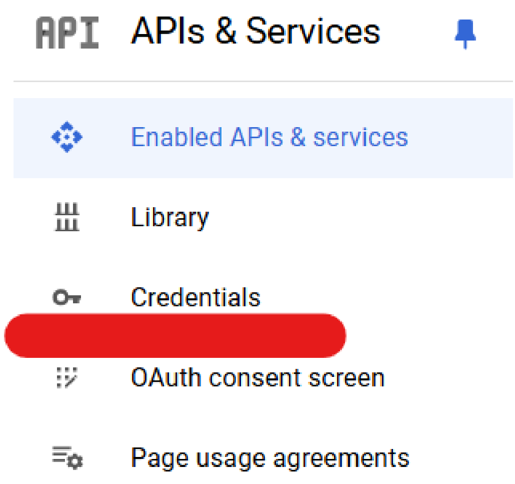
Click to try this API in API Explorer

Configuring Credentials

Step 1: Access Credentials

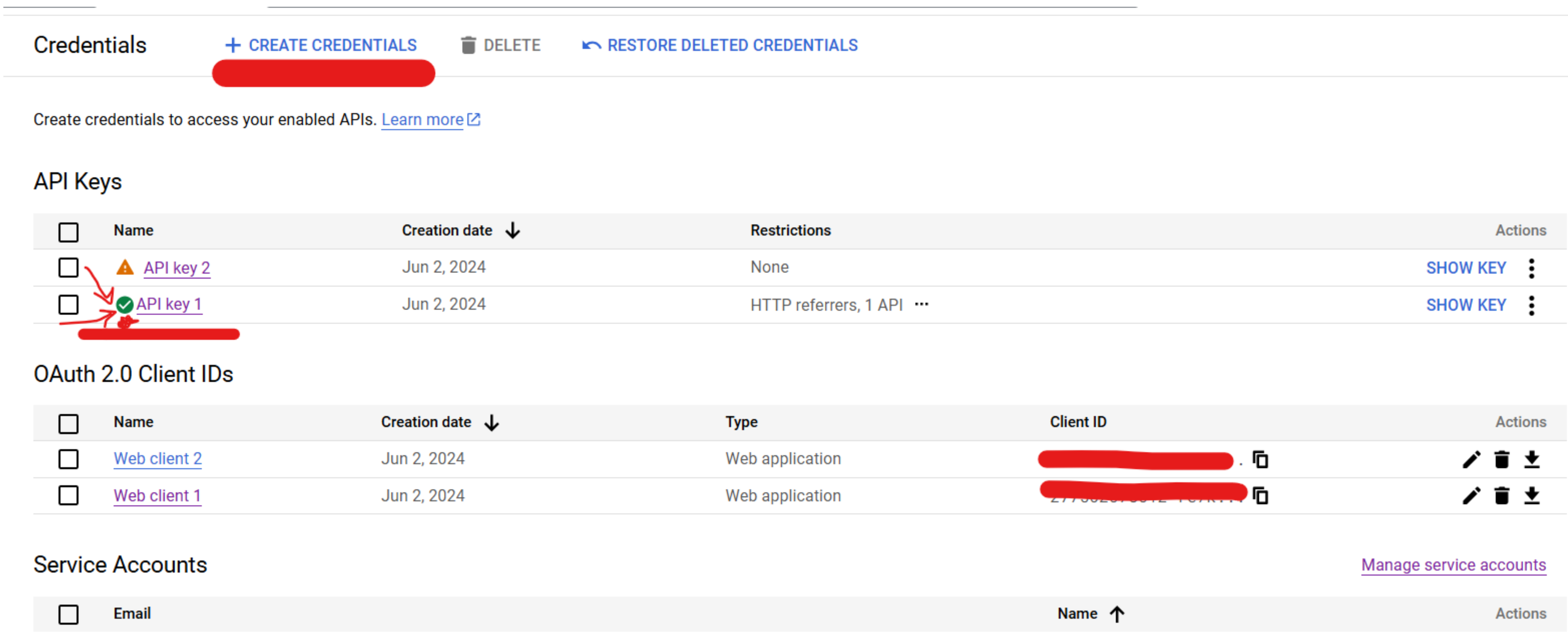
Go to the 'Credentials' section after enabling the necessary APIs.

!Credentials Section



Step 2: API Key

Ensure your API key is enabled for editing.



Credentials

Create credentials to access your enabled APIs. Learn more

API Keys

□	Name	Creation date ↓	Restrictions	Actions
□	⚠ API key 2	Jun 2, 2024	None	SHOW KEY ⋮
□	✓ API key 1	Jun 2, 2024	HTTP referrers, 1 API ⋯	SHOW KEY ⋮

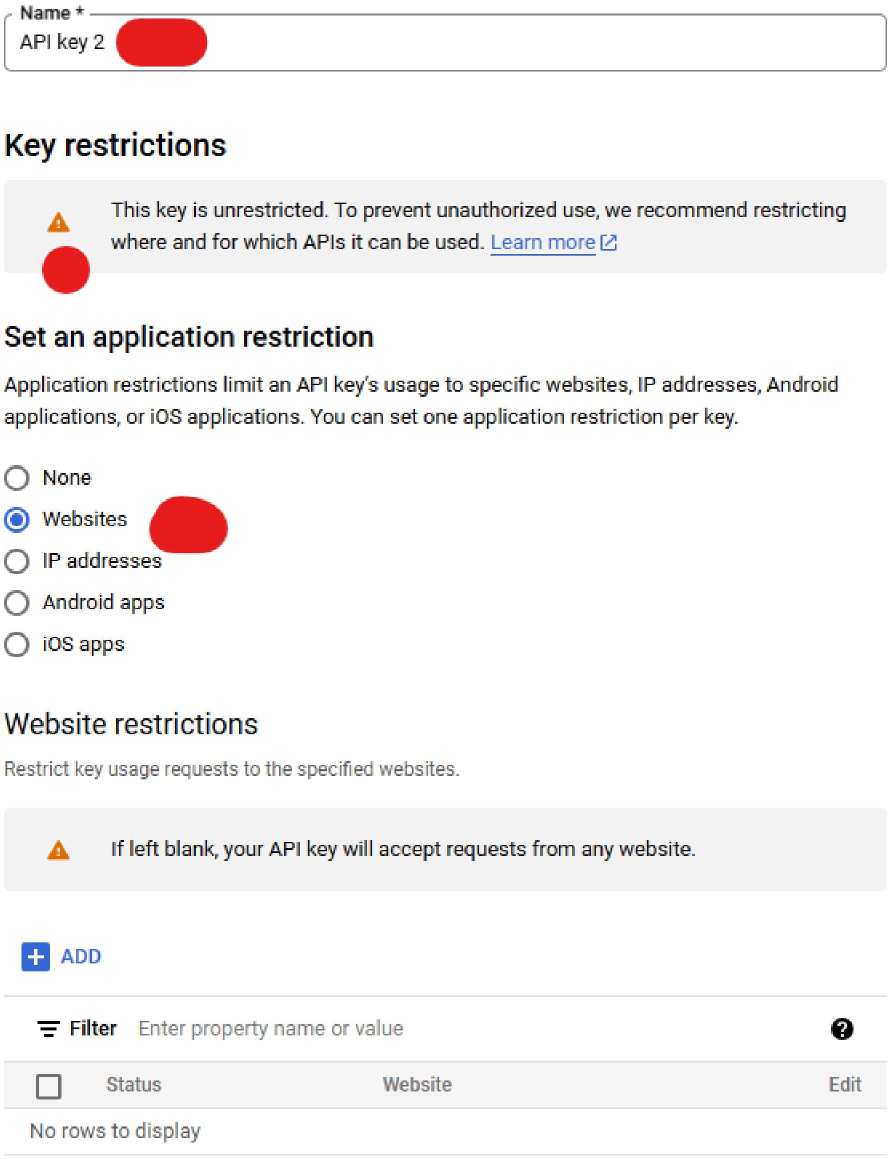
OAuth 2.0 Client IDs

□	Name	Creation date ↓	Type	Client ID	Actions
□	Web client 2	Jun 2, 2024	Web application	[redacted] ⧉	✎ 🗑 ⬇
□	Web client 1	Jun 2, 2024	Web application	[redacted] ⧉	✎ 🗑 ⬇

Service Accounts

Manage service accounts

□	Email	Name ↑	Actions



Name *

API key 2

Key restrictions

This key is unrestricted. To prevent unauthorized use, we recommend restricting where and for which APIs it can be used. Learn more

Set an application restriction

Application restrictions limit an API key's usage to specific websites, IP addresses, Android applications, or iOS applications. You can set one application restriction per key.

○ None

● Websites

○ IP addresses

○ Android apps

○ iOS apps

Website restrictions

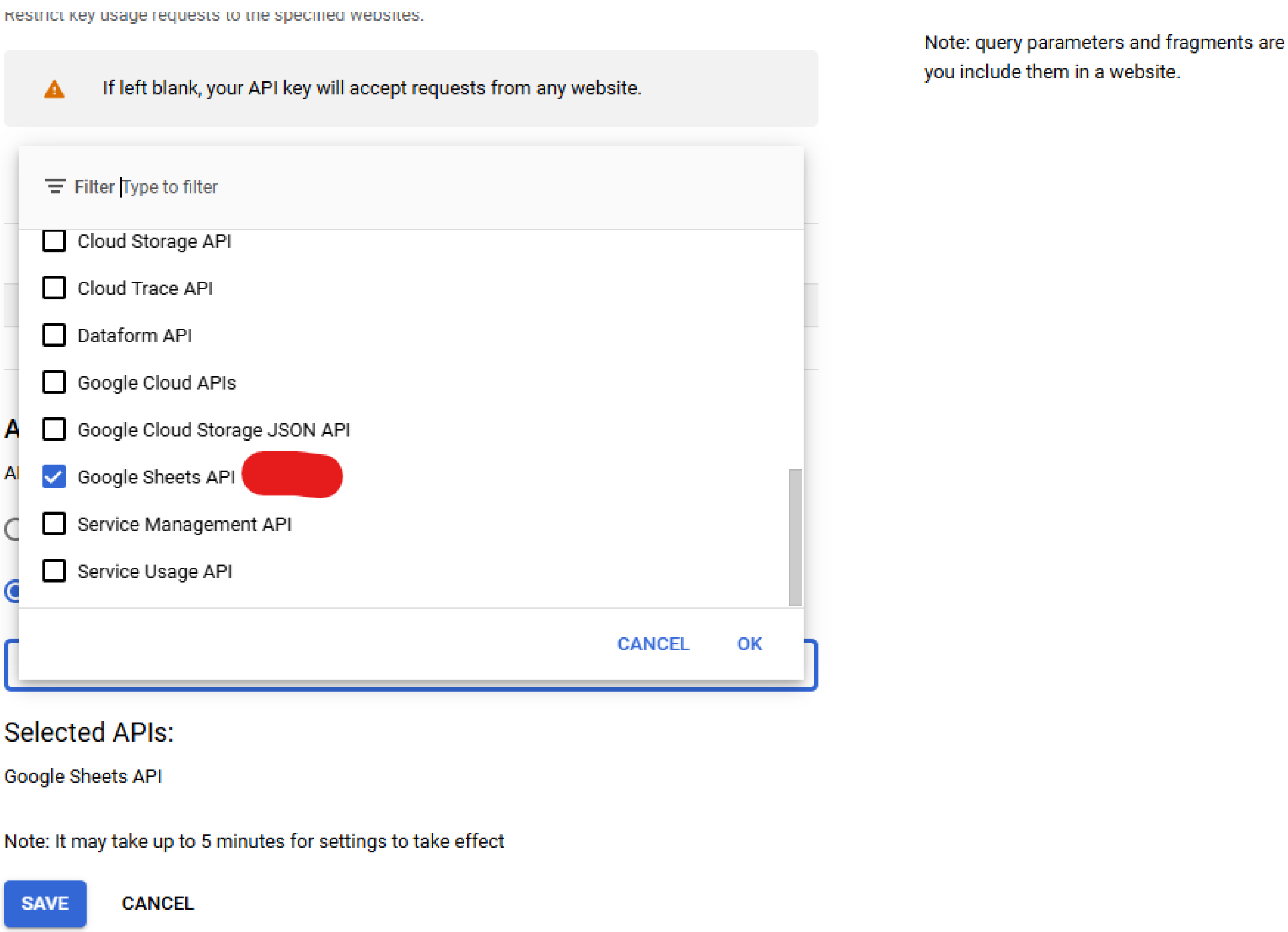
Restrict key usage requests to the specified websites.

If left blank, your API key will accept requests from any website.

□	Status	Website	Edit

No rows to display

!API Key Editing



Restrict key usage requests to the specified websites.

⚠ If left blank, your API key will accept requests from any website.

Note: query parameters and fragments are you include them in a website.

- □ Cloud Storage API
- □ Cloud Trace API
- □ Dataform API
- □ Google Cloud APIs
- □ Google Cloud Storage JSON API
- ☑ Google Sheets API
- □ Service Management API
- □ Service Usage API

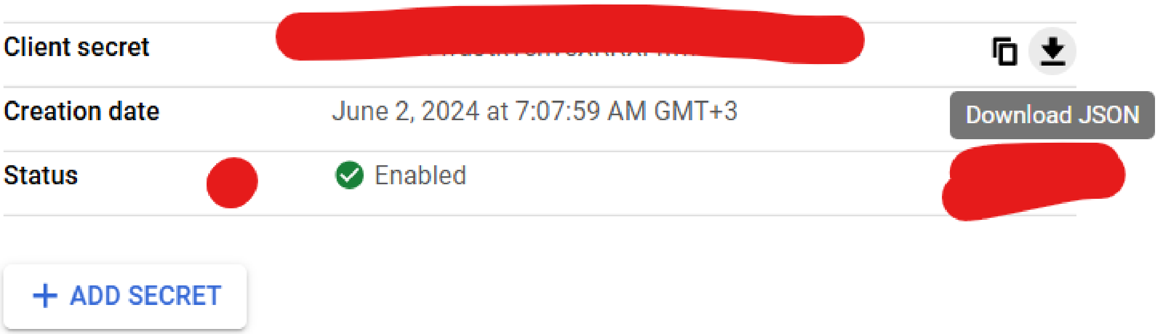
Selected APIs:

Google Sheets API

Note: It may take up to 5 minutes for settings to take effect

Step 3: WebClient

Enable the WebClient and download the JSON file.



Client secret [redacted]

Creation date June 2, 2024 at 7:07:59 AM GMT+3

Status ✓ Enabled

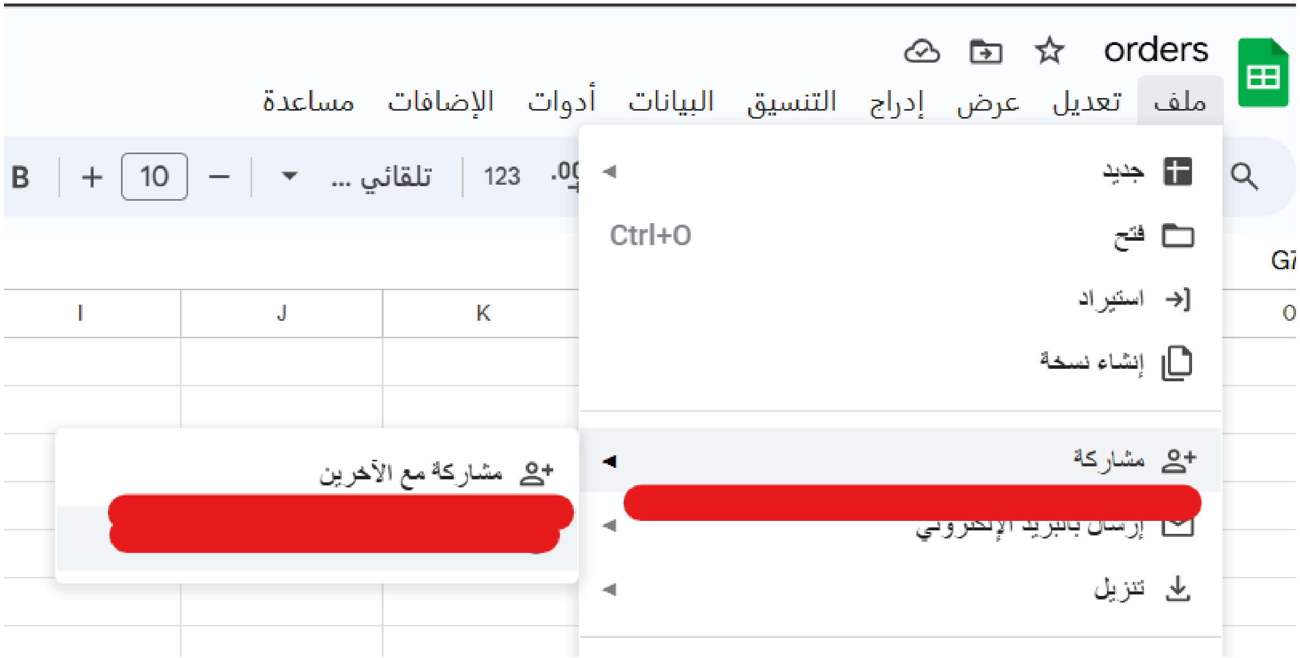
!WebClient JSON

Google Sheets Setup

Step 1: Access Google Sheets

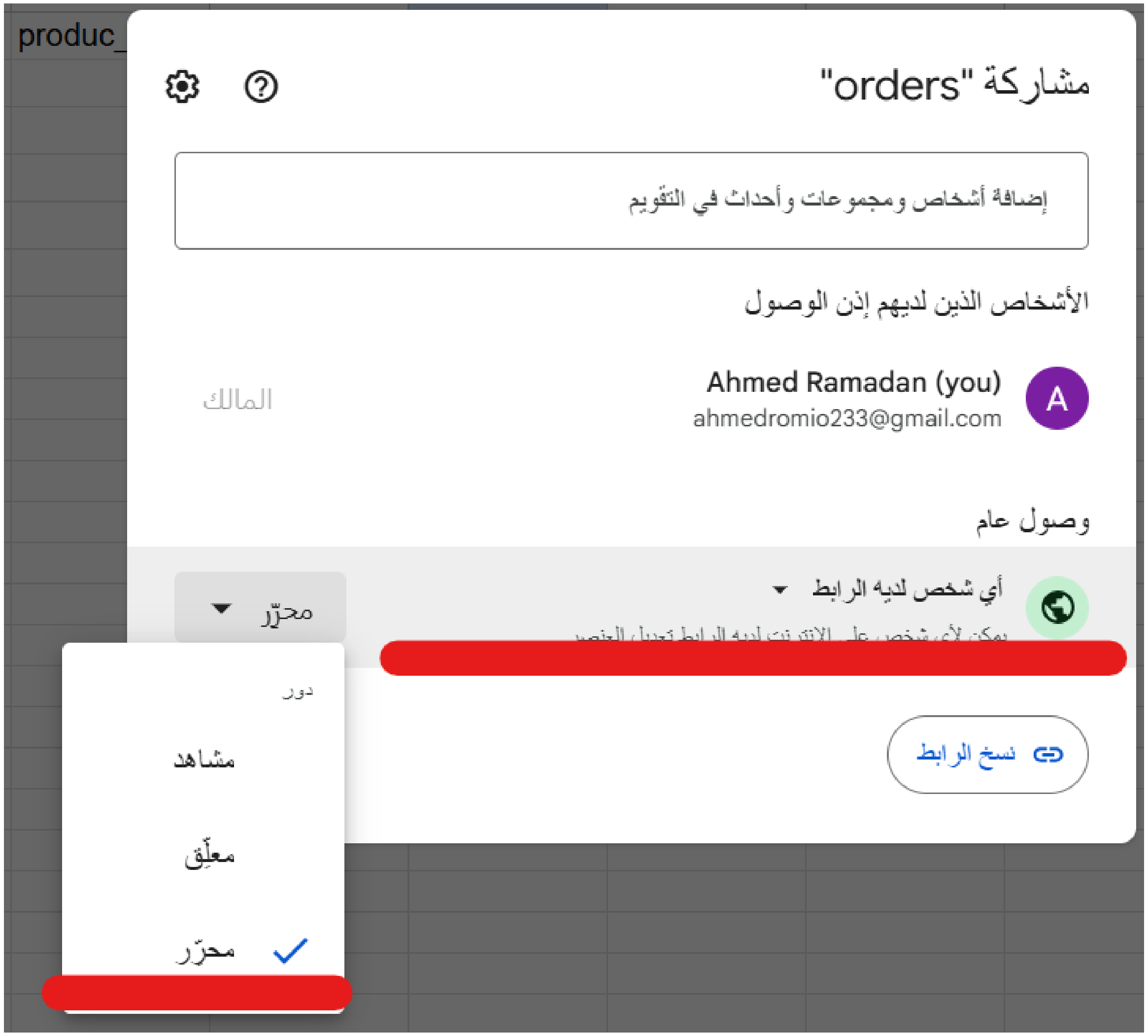
Go to your Google Sheet and ensure it is set to 'Share and Edit'.

!Google Sheets Share and Edit



[redacted]

[redacted]



مشاركة "orders"

إضافة أشخاص ومجموعات وأحداث في التقويم

الأشخاص الذين لديهم إذن الوصول

Ahmed Ramadan (you) ahmedromio233@gmail.com — المالك

وصول عام

أي شخص لديه الرابط — محرّر

يمكن لأي شخص على الإنترنت لديه الرابط تعديل العنصر

دور

مشاهد

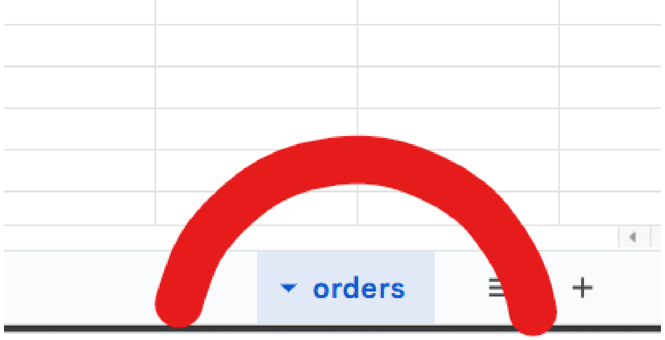
معلّق

✓ محرّر

Step 2: Sheet Name

Note that the name you see is the name of the paper, not the sheet.

!Paper Name



Visual Studio Code Setup

Step 1: JSON File

Place the downloaded JSON file into the storage directory of your Laravel project.

!JSON File in Storage

Step 2: Environment Variables

Set the following environment variables in your `.env` file:

```
GOOGLE_API_KEY=your-google-api-key
GOOGLE_CLIENT_ID=your-google-client-id
GOOGLE_CLIENT_SECRET=your-google-client-secret
GOOGLE_SERVICE_ACCOUNT_EMAIL=your-service-account-email
GOOGLE_SERVICE_ACCOUNT_JSON=your-json-file-name
GOOGLE_REDIRECT_URL=your-redirect-url
```

Step 3: Composer Commands

Run the following commands to install the necessary package and publish the configuration:

```
composer require revolution/laravel-google-sheets
```

```
php artisan vendor:publish --provider="PulkitJalan\Google\GoogleServiceProvider" --tag="config"
```

Step 4: Configuration File

Create a google.php file in the config folder with the following content:

```
<?php

return [
    'application_name' => env('GOOGLE_APPLICATION_NAME', 'My Laravel Application'),
    'client_id'        => env('GOOGLE_CLIENT_ID'),
    'client_secret'    => env('GOOGLE_CLIENT_SECRET'),
    'redirect_uri'     => env('GOOGLE_REDIRECT_URL'),
    'scopes'           => [
        \Google\Service\Sheets::DRIVE,
        \Google\Service\Sheets::SPREADSHEETS,
    ],
    'access_type'      => 'offline',
    'approval_prompt'  => 'force',
    'developer_key'    => env('GOOGLE_API_KEY'),
    'service'          => [
        'enable' => true,
        'file'   => env('GOOGLE_SERVICE_ACCOUNT_JSON', storage_path('app/google-service-account.json')),
    ],
    'config'           => [],
];
```

Step 5: Clear Configuration Cache

Remember to clear the configuration cache after making changes:

```
php artisan config:clear
```

```
php artisan config:cache
```

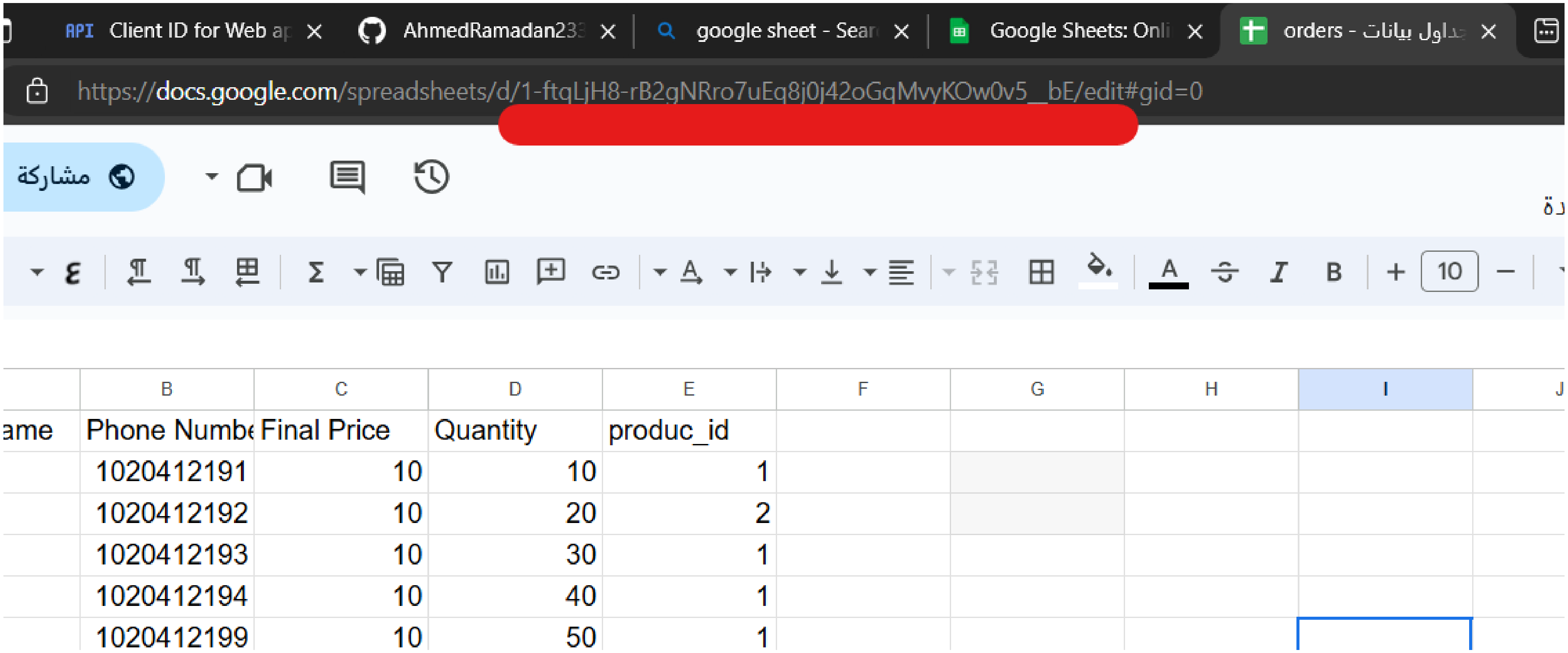
```
php artisan cache:clear
```

Generate Process Google Sheets Data Controller

Step 1: Retrieve Data

Get the $spreadsheetId from the URL of the sheet.

!Spread sheet Id



https://docs.google.com/spreadsheets/d/1-ftqLjH8-rB2gNRro7uEq8j0j42oGqMvyKOw0v5__bE/edit#gid=0

	B	C	D	E	F	G	H	I	J
ame	Phone Numbe	Final Price	Quantity	produc_id					
	1020412191	10	10	1					
	1020412192	10	20	2					
	1020412193	10	30	1					
	1020412194	10	40	1					
	1020412199	10	50	1					

Define the `$ordersSheetName` as `orders!A2:F6`, which is the name of the sheet paper and its range.

```
use Revolution\Google\Sheets\Facades\Sheets;

public function index()
{
    $spreadsheetId = '1-ftqLjH8-rB2gNRro7uEq8j0j42oGqMvyKOw0v5__bE';
    $spreadsheetProductId = '14jNcQTMwAlRcx_JJpNszqovWzuBwrjKWvw4eZeotzJw';
    $ordersSheetName = 'orders!A2:f6';

    $productsSheetName = 'products!A2:D6';
    $orders = Sheets::spreadsheet($spreadsheetId)
        ->sheet($ordersSheetName)
        ->get();

    $products = Sheets::spreadsheet($spreadsheetProductId)
        ->sheet($productsSheetName)
        ->get();

    return response()->json(['orders' => $orders, 'products' => $products]);
}
```