Log Sensor Data to Google Sheet using Raspberry Pi / Laptop using Python and Google App Script

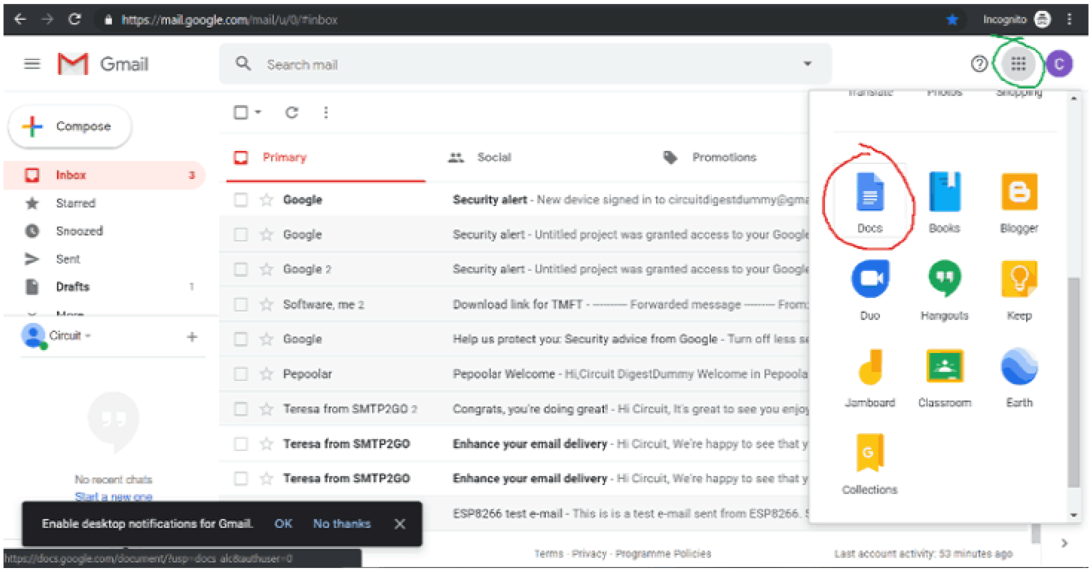
There are many IoT platforms available to store and monitor the data from various sensors. With the increasing popularity of IoT, more number of sensors are used every day and these sensors produce anonymous data which are useful to control other devices on the IoT network. These data can be studied further and used to modify design or improve the system performance. The Applications of Data Logging is vast and can be applied in Research Centres, Scientific Labs, Operation Theatres, Patient Monitoring System and many more.

Although there are many IoT platforms, we are restricted to the features provided by these platforms. If we want to have more control over our data and the freedom of working with our data, then we need a platform like Google sheets which is freely available and allows us to program its sheets and data contained in it through Google App Script.

In this tutorial, we will use the **Google Sheet as an IoT Database to log the data generated by a Sensor**. Here we will use Python to send the sensor data from Raspberry Pi to the Google Sheet over the internet.

I. <u>Creating Google Script in Google Sheet for Data Logging</u>:

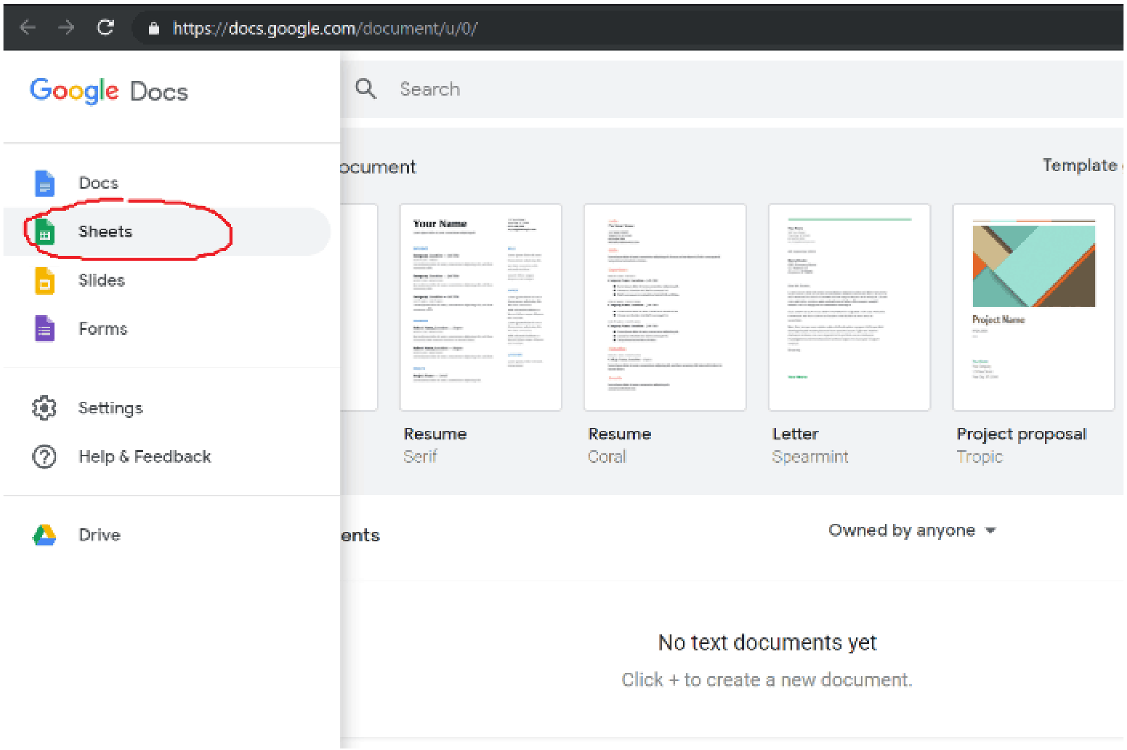
1. Login to your **Gmail** Account or **Google Drive** Account.
2. Go to the App Icon In Top Right Corner Highlighted in Green Circle and Click on Docs.



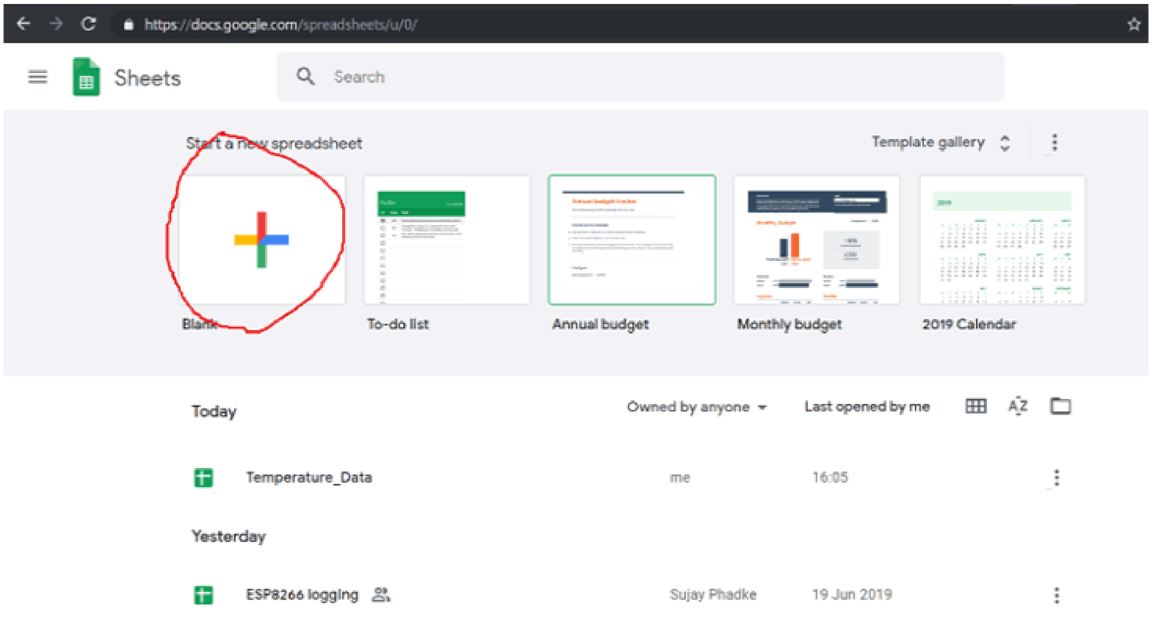
Google	**Security alert** - New device signed in to circuitdigestdummy@gma
Google	Security alert - Untitled project was granted access to your Google
Google 2	Security alert - Untitled project was granted access to your Google
Software, me 2	Download link for TMFT - --------- Forwarded message --------- From:
Google	Help us protect you: Security advice from Google - Turn off less se
Pepoolar	Pepoolar Welcome - Hi,Circuit DigestDummy Welcome in Pepoola
Teresa from SMTP2GO 2	Congrats, you're doing great! - Hi Circuit, It's great to see you enjoy
Teresa from SMTP2GO	**Enhance your email delivery** - Hi Circuit, We're happy to see that y
Teresa from SMTP2GO	**Enhance your email delivery** - Hi Circuit, We're happy to see that y
	ESP8266 test e-mail - This is is a test e-mail sent from ESP8266. S

Enable desktop notifications for Gmail. OK No thanks

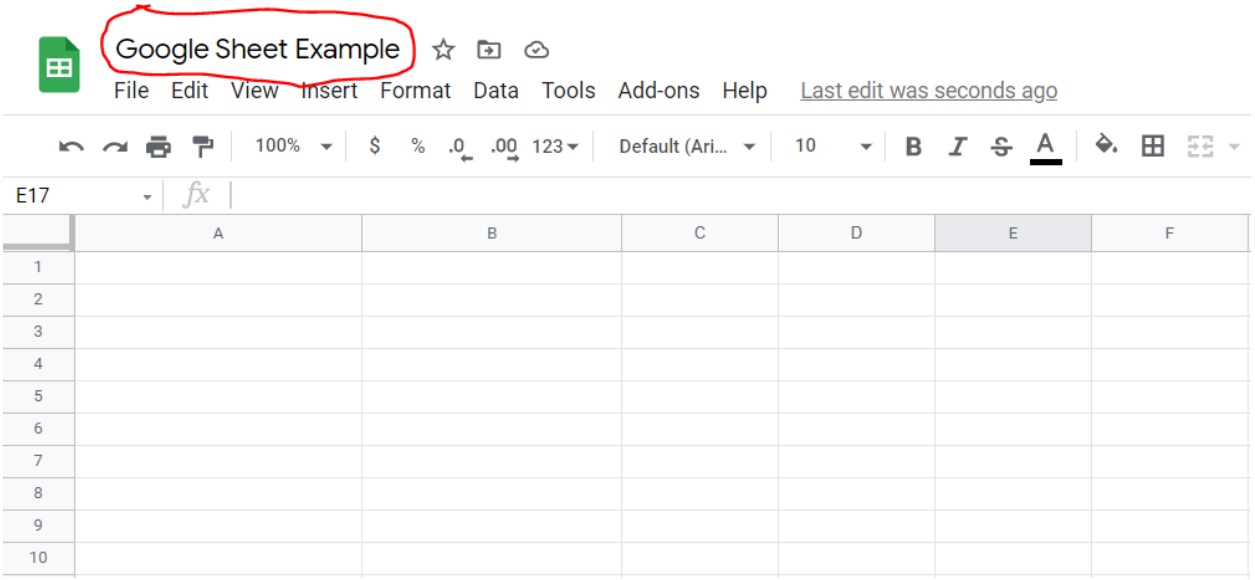
3. The Google Docs screen will appear. Now choose Sheets in the right sidebar.



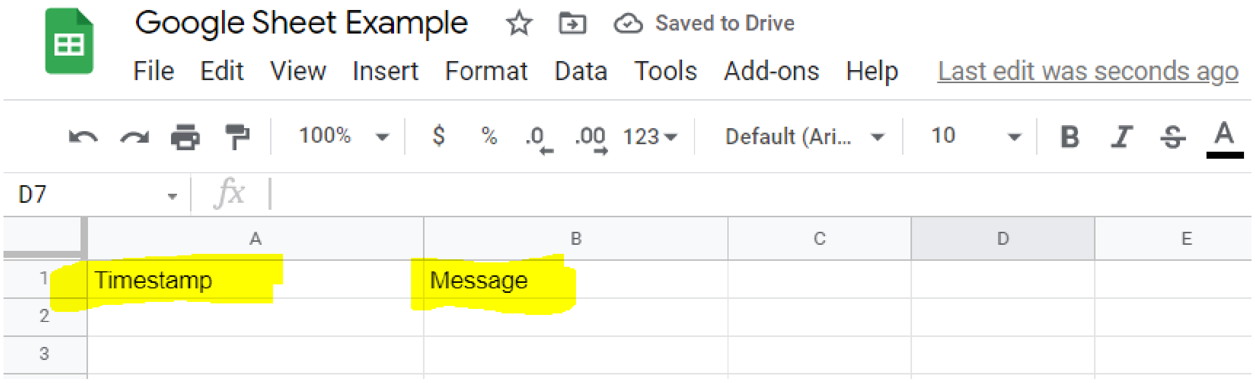
4. Create a New Blank Sheet.



5. The Blank Sheet will be created with an **“Untitled Spreadsheet”**. Just rename this created Spreadsheet Project to any name you want. Name is not important for this tutorial.

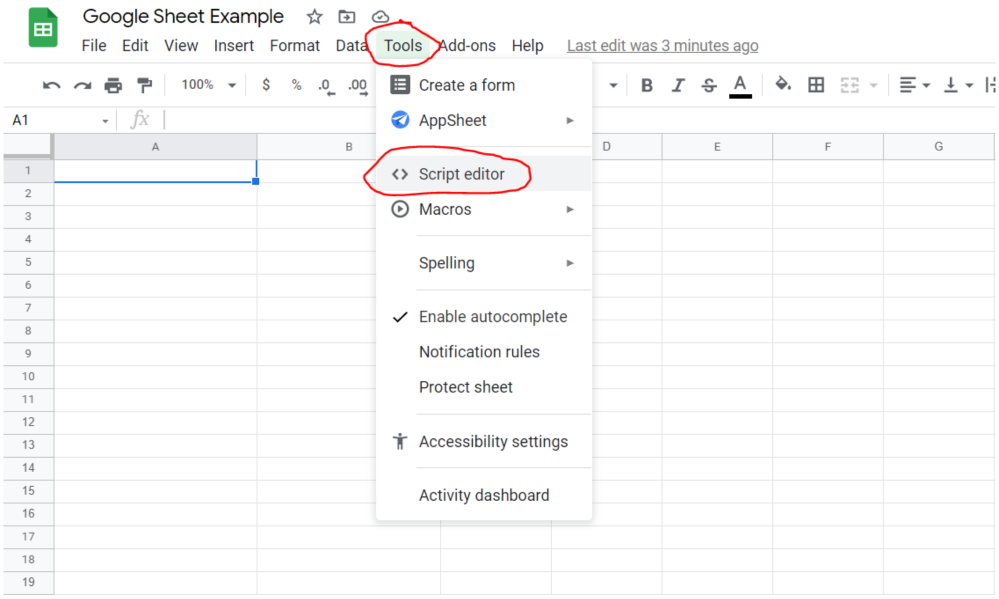


6. Add Header to the Spreadsheet as shown below:

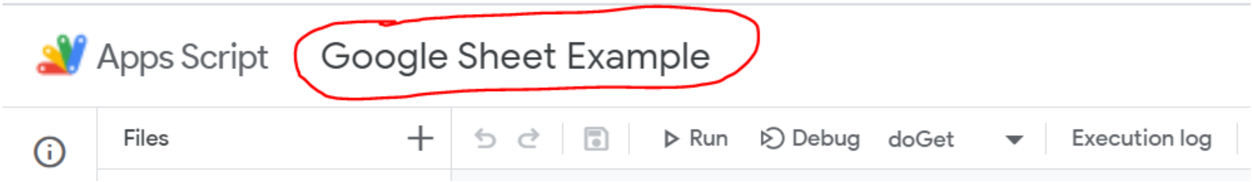

Google Sheet Example
Saved to Drive
File Edit View Insert Format Data Tools Add-ons Help
Last edit was seconds ago
100%
$ % .0 .00 123
Default (Ari...
10
D7
fx

	A	B	C	D	E
1	Timestamp	Message			
2					
3					

7. Now, in the menu bar, go to Tools → Script Editor



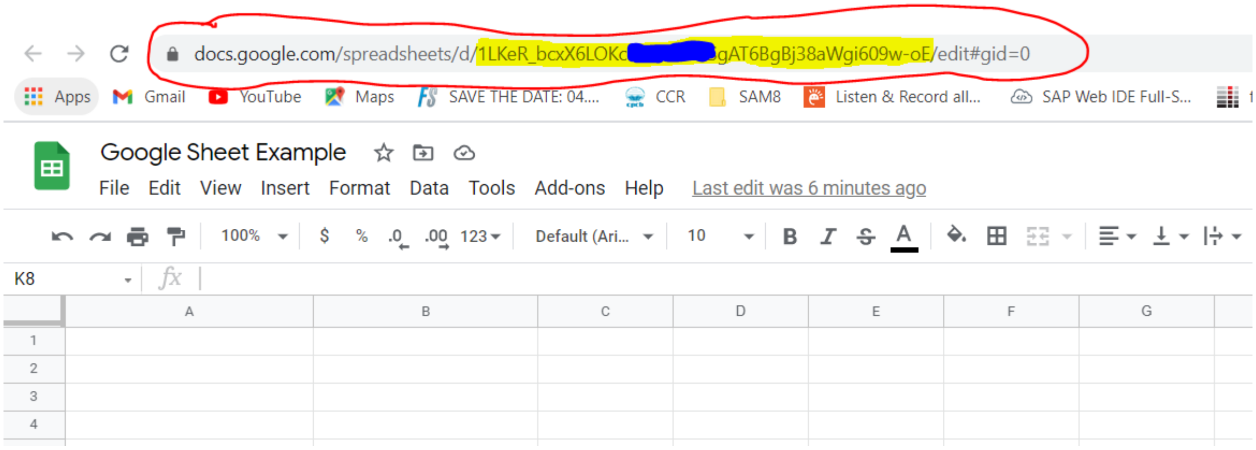
8. The new Google Script is created with “Untitled project”. You can rename this Google Script File to any name you want. We have named it as “Google Sheet Example”.



9. Download the code from https://github.com/ShubhamMadhavi99/Rpi-Workshop/blob/main/4.%20Raspberry%20Pi%20Cloud%20Networking/Google%20Sheet%20Example/Google_Sheet_code.gs

Copy and Paste this code in your Google Script Editor

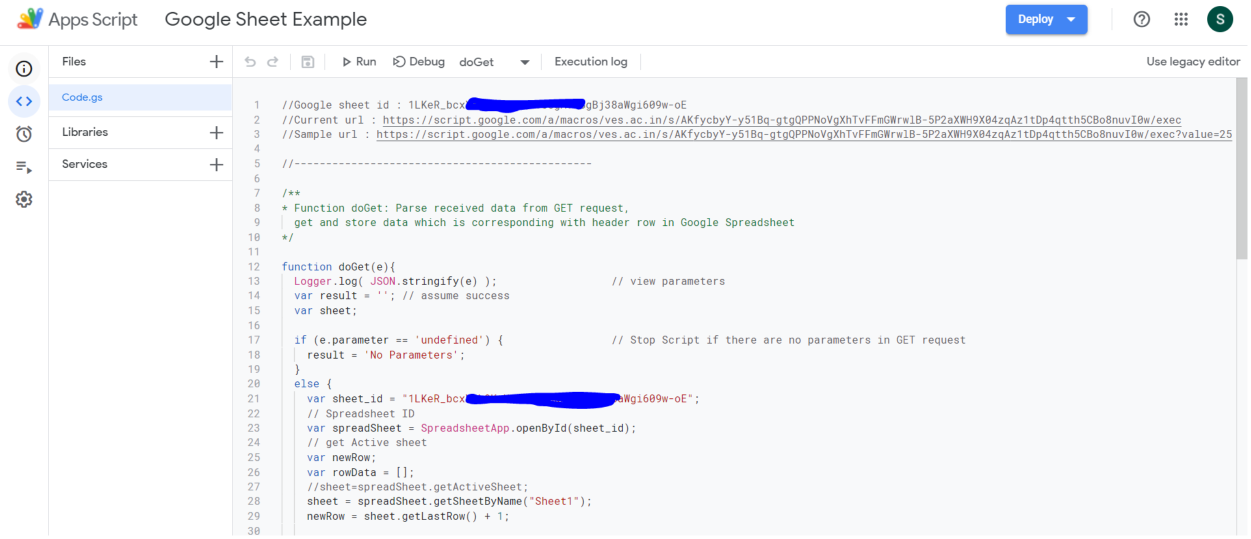
10. Then edit the Sheet name and Sheet ID in the code. You can get the Sheet ID from the Sheet URL just like shown. https://docs.google.com/spreadsheets/d/xxxxxxxxyyyyyyzzzzzzzzzzz/edit#gid=0 , where “xxxxxxxxyyyyyyzzzzzzzzzzz” is your Sheet ID. Every document has an ID on google drive. Check your sheet id and copy it.



docs.google.com/spreadsheets/d/1LKeR_bcxX6LOKc[redacted]gAT6BgBj38aWgi609w-oE/edit#gid=0

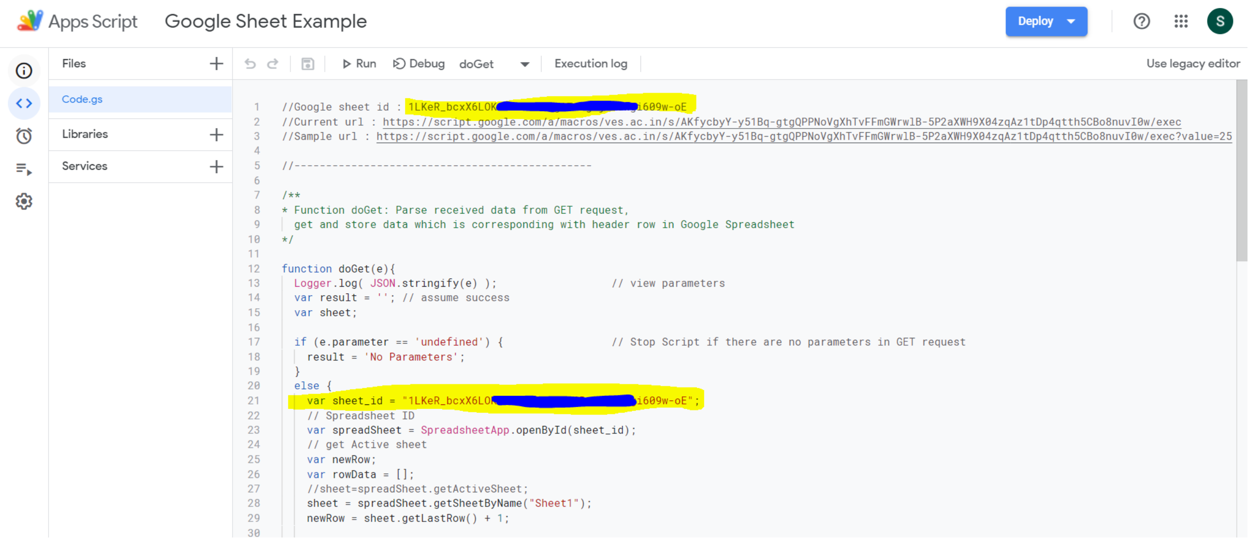
Note: Some part has been hidden using blue highlight to keep the ID hidden.

11. When you copy and paste the Google Script then it will look like the following.



```
//Google sheet id : 1LKeR_bcx[redacted]gBj38aWgi609w-oE
//Current url : https://script.google.com/a/macros/ves.ac.in/s/AKfycbyY-y51Bq-gtgQPPNoVgXhTvFFmGWrwlB-5P2aXWH9X04zqAz1tDp4qtth5CBo8nuvI0w/exec
//Sample url : https://script.google.com/a/macros/ves.ac.in/s/AKfycbyY-y51Bq-gtgQPPNoVgXhTvFFmGWrwlB-5P2aXWH9X04zqAz1tDp4qtth5CBo8nuvI0w/exec?value=25

//---------------------------------------------

/**
* Function doGet: Parse received data from GET request,
  get and store data which is corresponding with header row in Google Spreadsheet
*/

function doGet(e){
  Logger.log( JSON.stringify(e) );                    // view parameters
  var result = ''; // assume success
  var sheet;

  if (e.parameter == 'undefined') {                   // Stop Script if there are no parameters in GET request
    result = 'No Parameters';
  }
  else {
    var sheet_id = "1LKeR_bcx[redacted]aWgi609w-oE";
    // Spreadsheet ID
    var spreadSheet = SpreadsheetApp.openById(sheet_id);
    // get Active sheet
    var newRow;
    var rowData = [];
    //sheet=spreadSheet.getActiveSheet;
    sheet = spreadSheet.getSheetByName("Sheet1");
    newRow = sheet.getLastRow() + 1;

```

12. In this code, replace sheet id with your own sheet id as shown below:



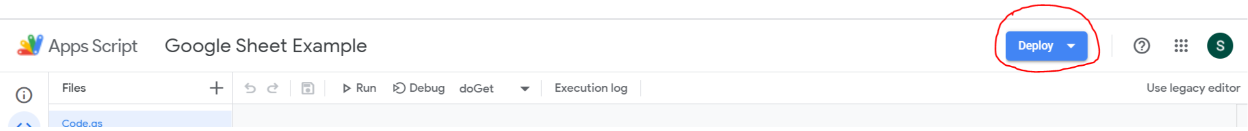
```
//Google sheet id : 1LKeR_bcxX6LOK[redacted]i609w-oE
//Current url : https://script.google.com/a/macros/ves.ac.in/s/AKfycbyY-y51Bq-gtgQPPNoVgXhTvFFmGWrwlB-5P2aXWH9X04zqAz1tDp4qtth5CBo8nuvI0w/exec
//Sample url : https://script.google.com/a/macros/ves.ac.in/s/AKfycbyY-y51Bq-gtgQPPNoVgXhTvFFmGWrwlB-5P2aXWH9X04zqAz1tDp4qtth5CBo8nuvI0w/exec?value=25

//---------------------------------------------

/**
* Function doGet: Parse received data from GET request,
  get and store data which is corresponding with header row in Google Spreadsheet
*/

function doGet(e){
  Logger.log( JSON.stringify(e) );                  // view parameters
  var result = ''; // assume success
  var sheet;

  if (e.parameter == 'undefined') {                 // Stop Script if there are no parameters in GET request
    result = 'No Parameters';
  }
  else {
    var sheet_id = "1LKeR_bcxX6LO[redacted]i609w-oE";
    // Spreadsheet ID
    var spreadSheet = SpreadsheetApp.openById(sheet_id);
    // get Active sheet
    var newRow;
    var rowData = [];
    //sheet=spreadSheet.getActiveSheet;
    sheet = spreadSheet.getSheetByName("Sheet1");
    newRow = sheet.getLastRow() + 1;

```

13. Save the file.

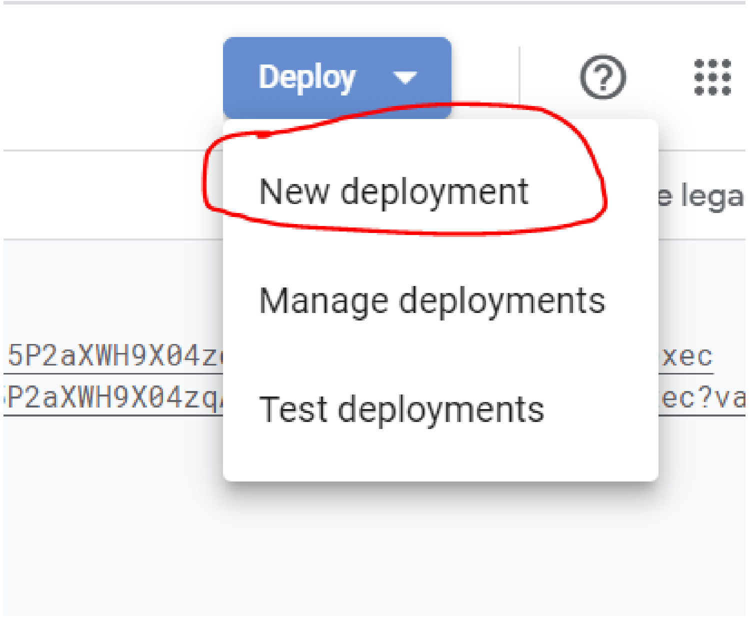
14. Now we have **finished the Setting up Google Script in Spreadsheet**. Now we will deploy the google script as a web app so that we can send requests to the sheet.

II. <u>Deploying Google Script</u>:

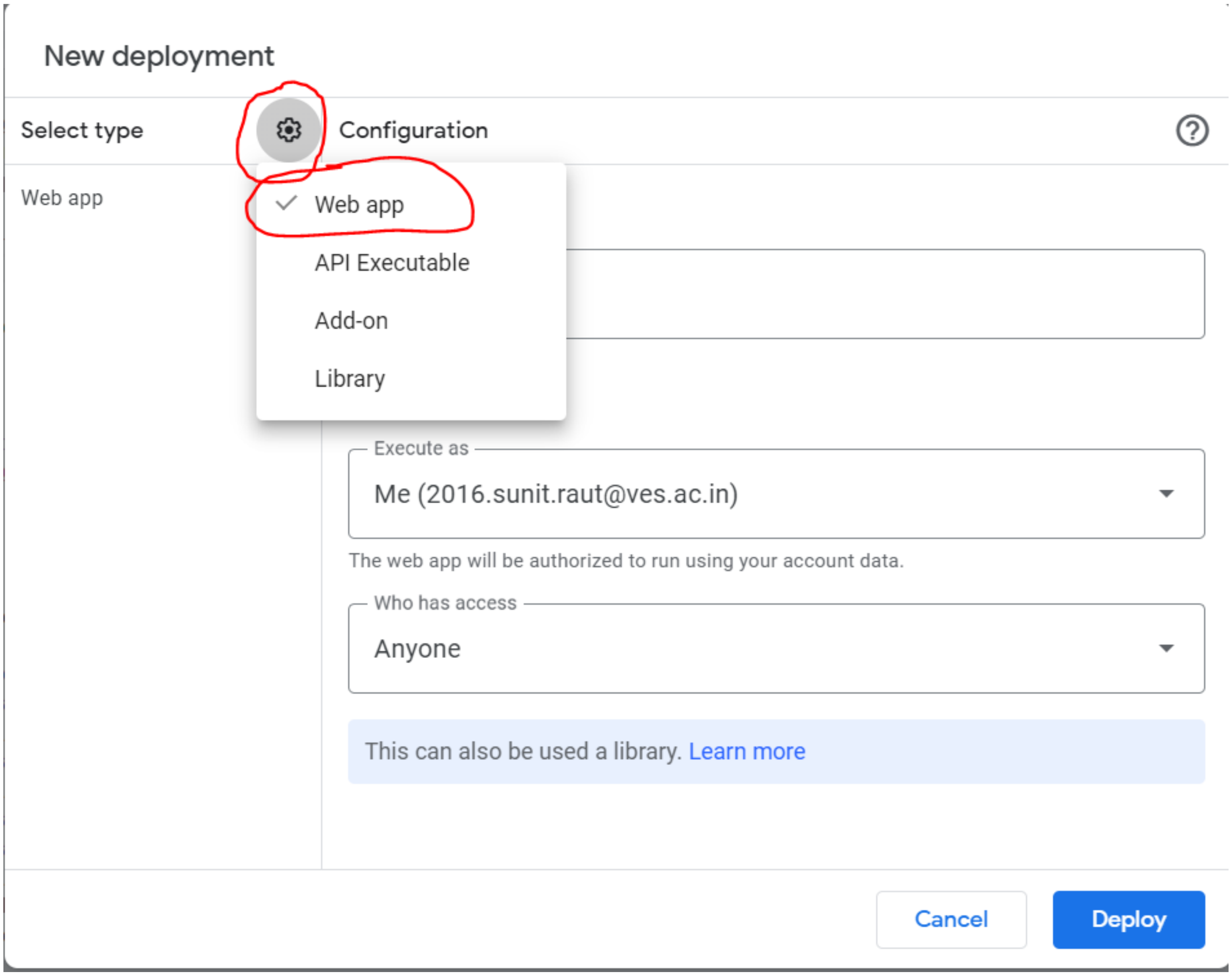
1. Click on Deploy Button



2. Click on New Deployment



3. A New window opens. Click on Setting Icon and select Web App.



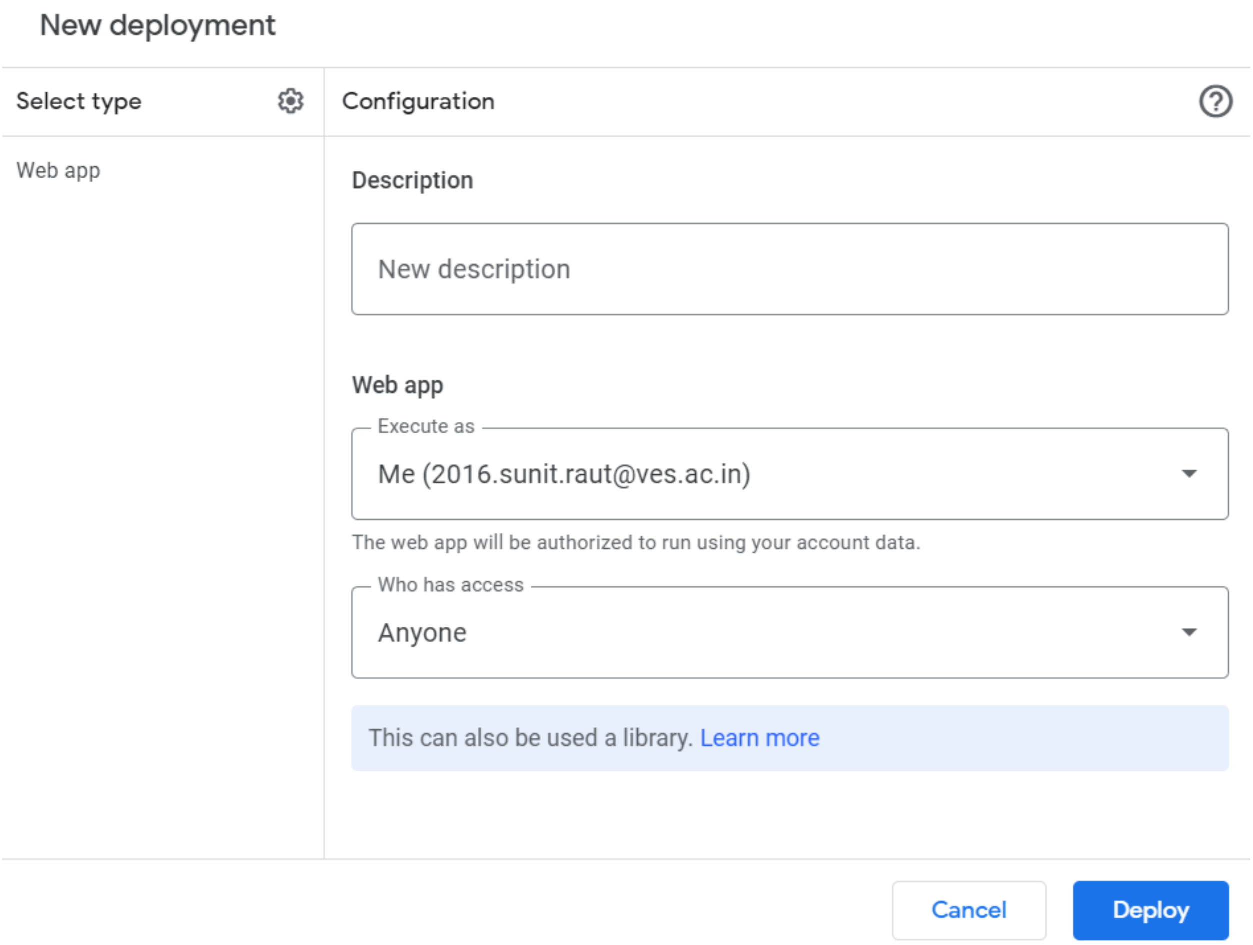
Execute as: Me (2016.sunit.raut@ves.ac.in)

The web app will be authorized to run using your account data.

Who has access: Anyone

This can also be used a library. Learn more

4. Select the options as below:



New deployment

Description

New description

Web app

Execute as: Me (2016.sunit.raut@ves.ac.in)

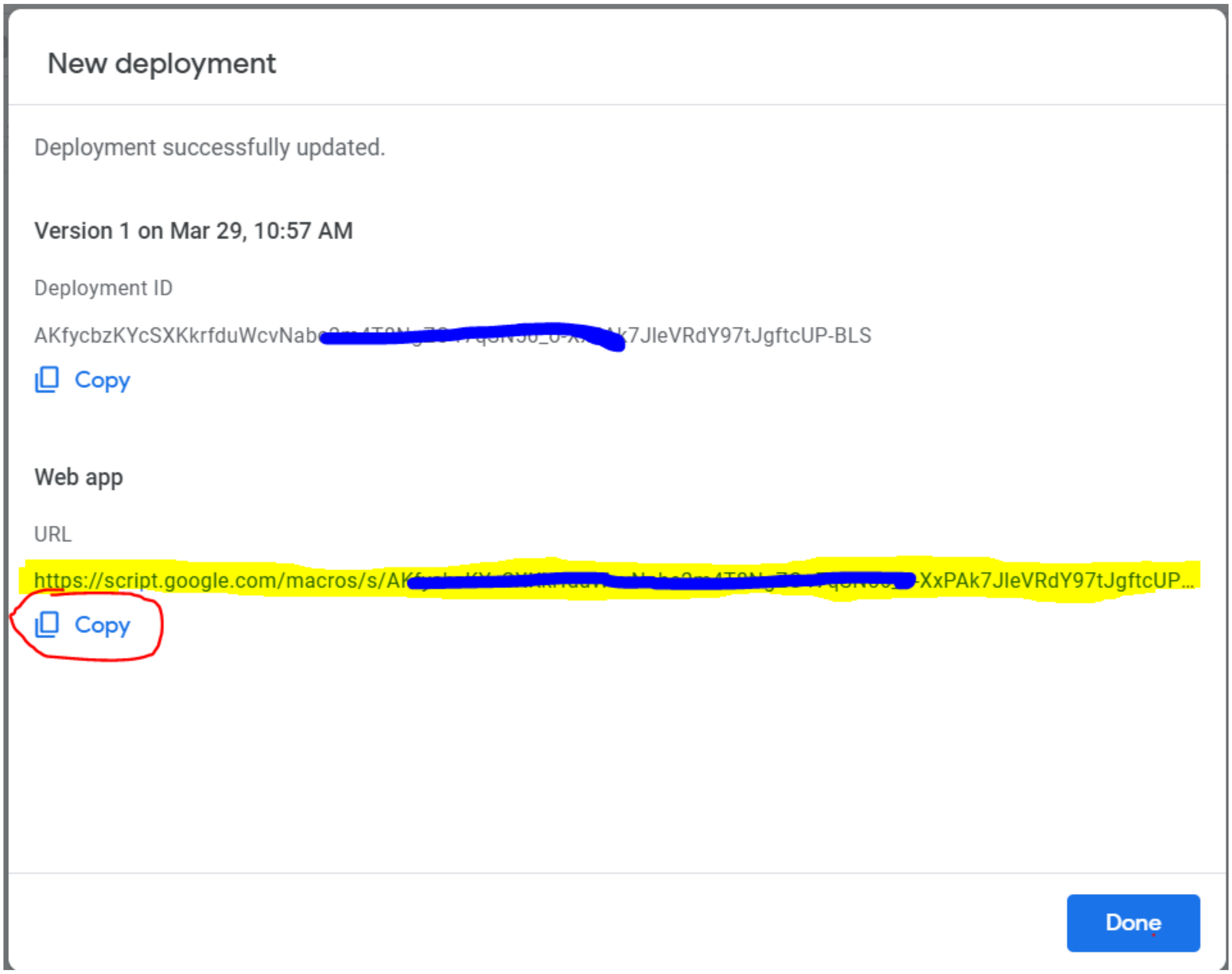
The web app will be authorized to run using your account data.

Who has access: Anyone

This can also be used a library. Learn more

Click on the Deploy button.

5. A Window will open up and display your WebApp URL. Copy it.



New deployment

Deployment successfully updated.

Version 1 on Mar 29, 10:57 AM

Deployment ID

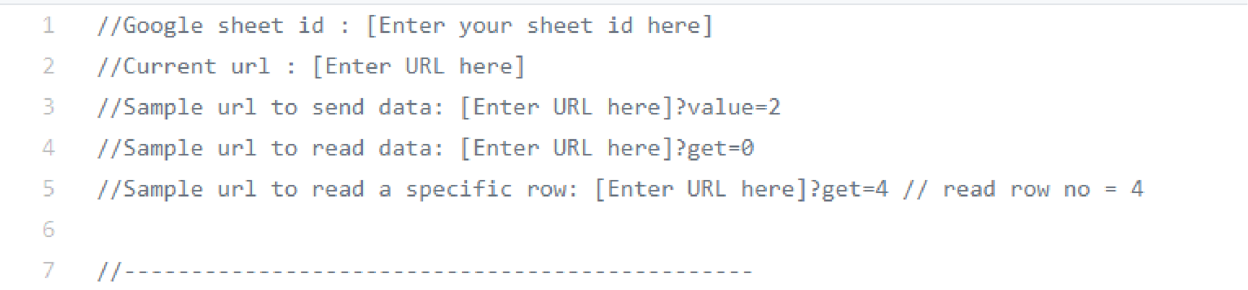
AKfycbzKYcSXKkrfduWcvNab[redacted]7JleVRdY97tJgftcUP-BLS

Web app

URL

https://script.google.com/macros/s/AK[redacted]XxPAk7JleVRdY97tJgftcUP…

6. Paste this in the code comments in Sample url for your reference. In the code, we have provided comments on top for your reference. Please enter your sheet id and webapp url in these comments for reference.



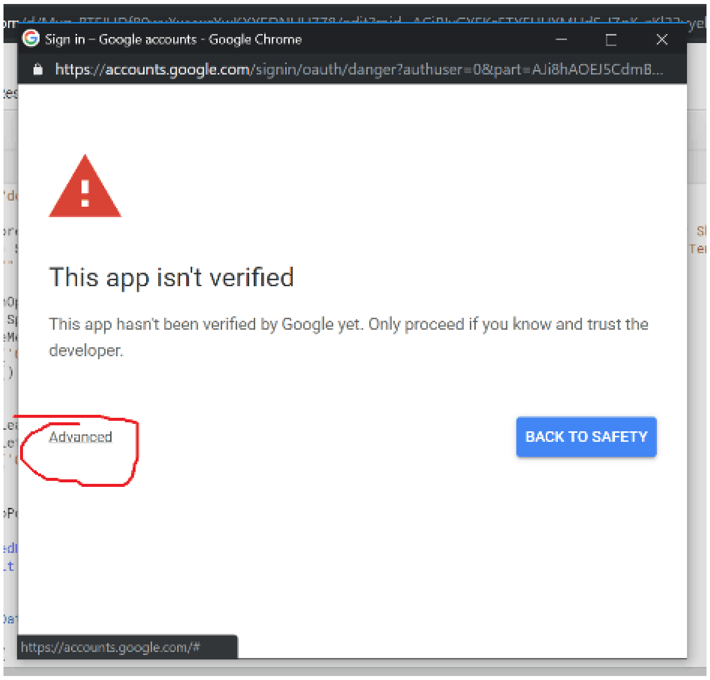
```
//Google sheet id : [Enter your sheet id here]
//Current url : [Enter URL here]
//Sample url to send data: [Enter URL here]?value=2
//Sample url to read data: [Enter URL here]?get=0
//Sample url to read a specific row: [Enter URL here]?get=4 // read row no = 4

//----------------------------------------------
```

Note: Each deployment has a different URL. You can check deployments in ‘Manage Deployments’ and you can also check URL there in case you lose track of URL.

7. Now, copy the above three urls in comments with query parameters “?value=25”, “?get=0”, “?get=1” and see results.

8. You may have to give Google permission to deploy it as a web app.

9. Then choose your Email ID here using which you have created the spreadsheet.
10. Click on “Advanced”.



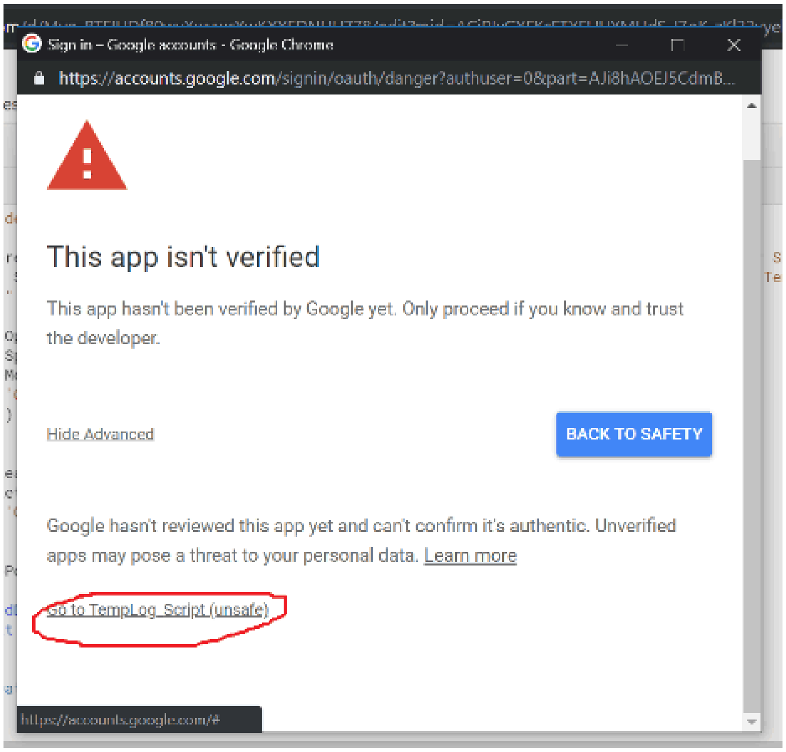
https://accounts.google.com/signin/oauth/danger?authuser=0&part=AJi8hAOEJ5CdmB...

This app isn't verified

This app hasn't been verified by Google yet. Only proceed if you know and trust the developer.

https://accounts.google.com/#

11. And then click on “Go to ‘your_script_name’ (unsafe)”.


Sign in – Google accounts - Google Chrome
https://accounts.google.com/signin/oauth/danger?authuser=0&part=AJi8hAOEJ5CdmB...
Hide Advanced
BACK TO SAFETY
https://accounts.google.com/#

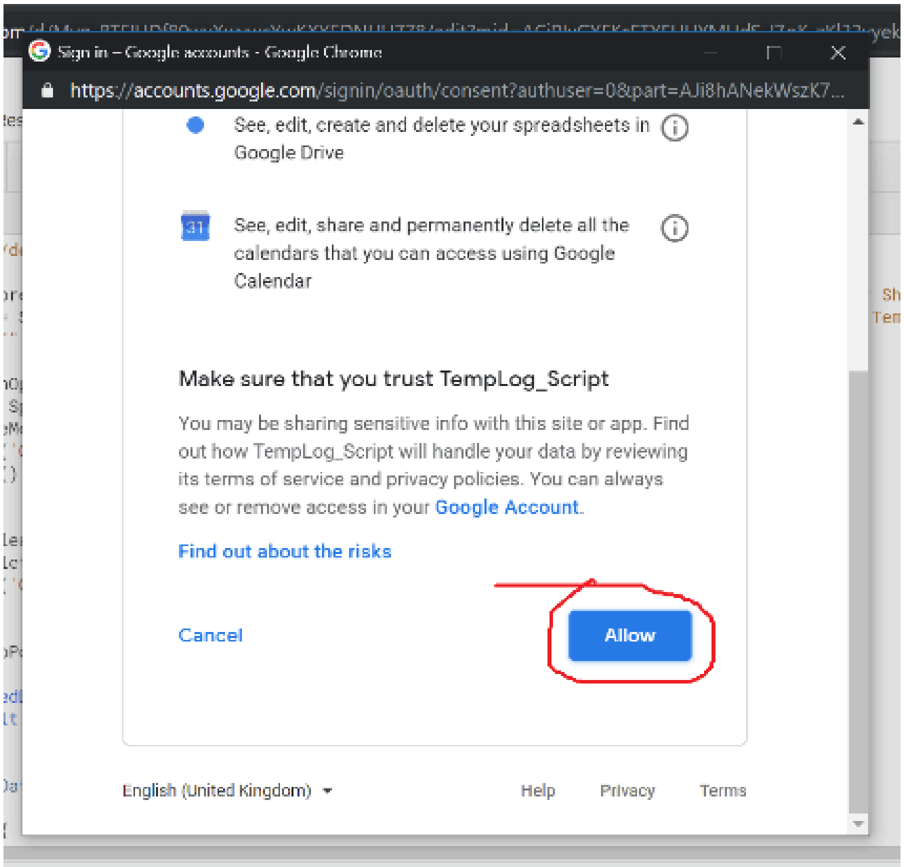
This app isn't verified

This app hasn't been verified by Google yet. Only proceed if you know and trust the developer.

Google hasn't reviewed this app yet and can't confirm it's authentic. Unverified apps may pose a threat to your personal data. Learn more

Go to TempLog_Script (unsafe)

12. Click on “*Allow*” and this will give the permission to deploy it as web app.



Sign in – Google accounts - Google Chrome
https://accounts.google.com/signin/oauth/consent?authuser=0&part=AJi8hANekWszK7...

See, edit, create and delete your spreadsheets in Google Drive

See, edit, share and permanently delete all the calendars that you can access using Google Calendar

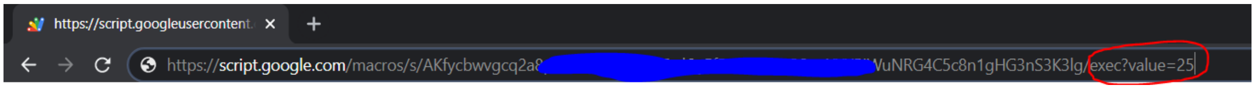
Make sure that you trust TempLog_Script

You may be sharing sensitive info with this site or app. Find out how TempLog_Script will handle your data by reviewing its terms of service and privacy policies. You can always see or remove access in your Google Account.

Find out about the risks

Sending Data to Google Sheet:

1. Enter URL in Browser and press Enter



https://script.google.com/macros/s/AKfycbwvgcq2a8[redacted]WuNRG4C5c8n1gHG3nS3K3lg/exec?value=25

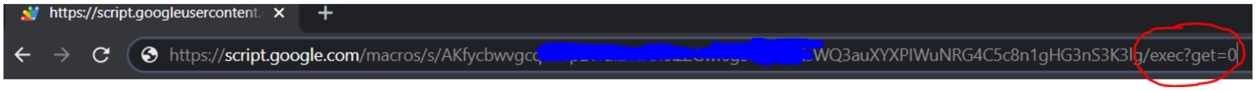
```
Written on column Value
```

2. Send request through Python using requests library

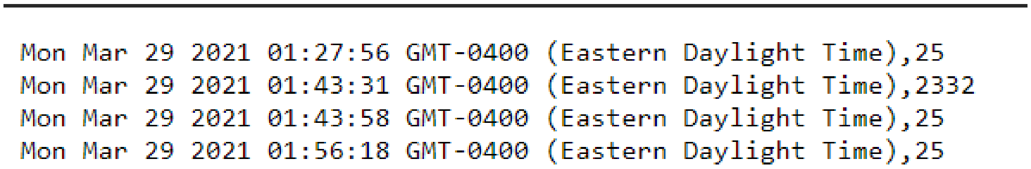
```
import requests
url = “[your url]?value=25”
requests.get(url)
```

Reading Data from Google Sheet:

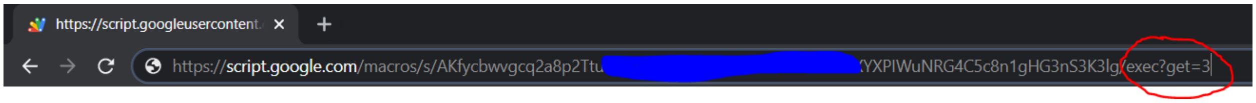
1. Enter URL in browser and press enter



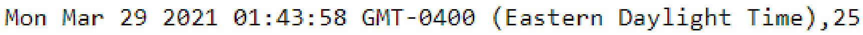
https://script.google.com/macros/s/AKfycbwvgcq[redacted]WQ3auXYXPIWuNRG4C5c8n1gHG3nS3K3lg/exec?get=0



```
Mon Mar 29 2021 01:27:56 GMT-0400 (Eastern Daylight Time),25
Mon Mar 29 2021 01:43:31 GMT-0400 (Eastern Daylight Time),2332
Mon Mar 29 2021 01:43:58 GMT-0400 (Eastern Daylight Time),25
Mon Mar 29 2021 01:56:18 GMT-0400 (Eastern Daylight Time),25
```



https://script.google.com/macros/s/AKfycbwvgcq2a8p2Ttu[redacted]YXPIWuNRG4C5c8n1gHG3nS3K3lg/exec?get=3



```
Mon Mar 29 2021 01:43:58 GMT-0400 (Eastern Daylight Time),25
```

2. Send request through Python using requests library

Read all rows and display

```
import requests
url = “[your url]?get=0”      // fetch all rows
x = requests.get(url)
print x.text                  //print results
```

Read particular row and display

```
import requests
url = “[your url]?get=3”     // fetch row no = 3
x = requests.get(url)
print x.text                 //print results
```