What are we building today ?

A simple telegram bot which calculates how much water you drank.

DRAFT workflow :

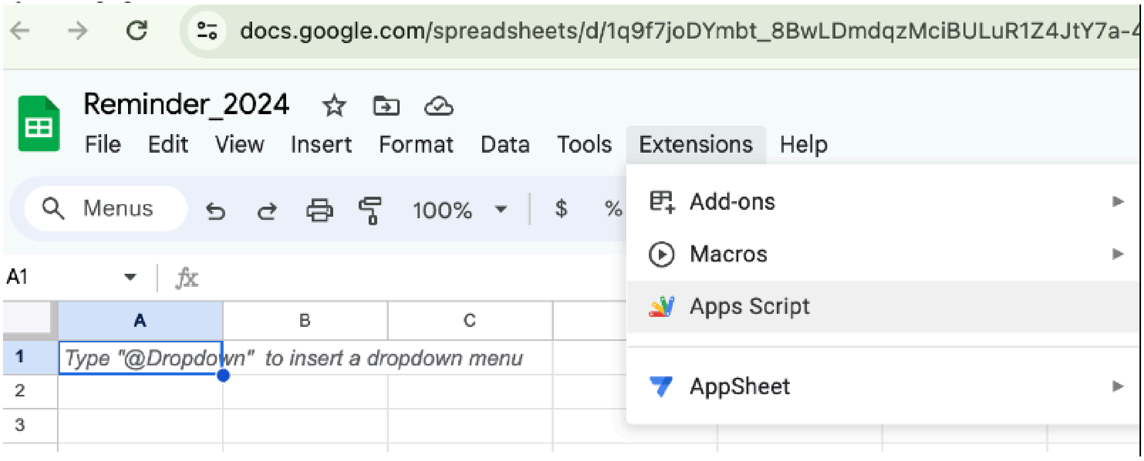
1. User enters liters of water
2. Check for error conditions (Not a positive number, >1.5 liters etc.)
3. Add content to google sheet
4. Finish off with a message highlighting liters of water drank today.

Pre-requisite

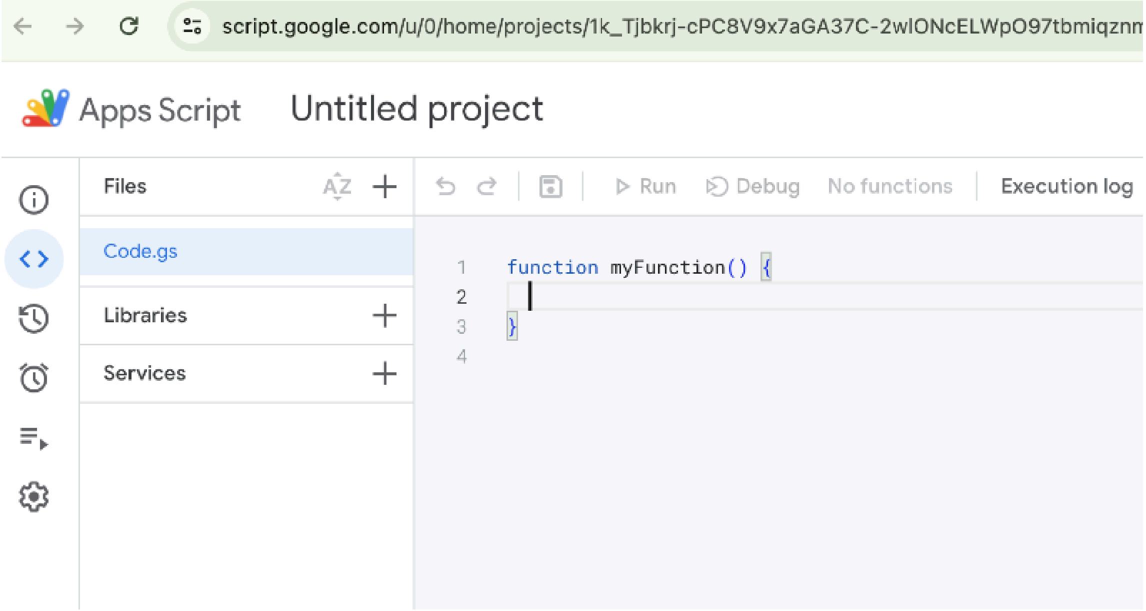
Setup a telegram bot

From BotFather / newbot , Collect the token

Open a new google sheet


docs.google.com/spreadsheets/d/1q9f7joDYmbt_8BwLDmdqzMciBULuR1Z4JtY7a-4
Reminder_2024
File
Edit
View
Insert
Format
Data
Tools
Extensions
Help
Menus
100%
$
%
A1
fx
A
B
C
1
2
3
Type "@Dropdown" to insert a dropdown menu
Add-ons
Macros
Apps Script
AppSheet

Open the code editor from Extensions/App Script

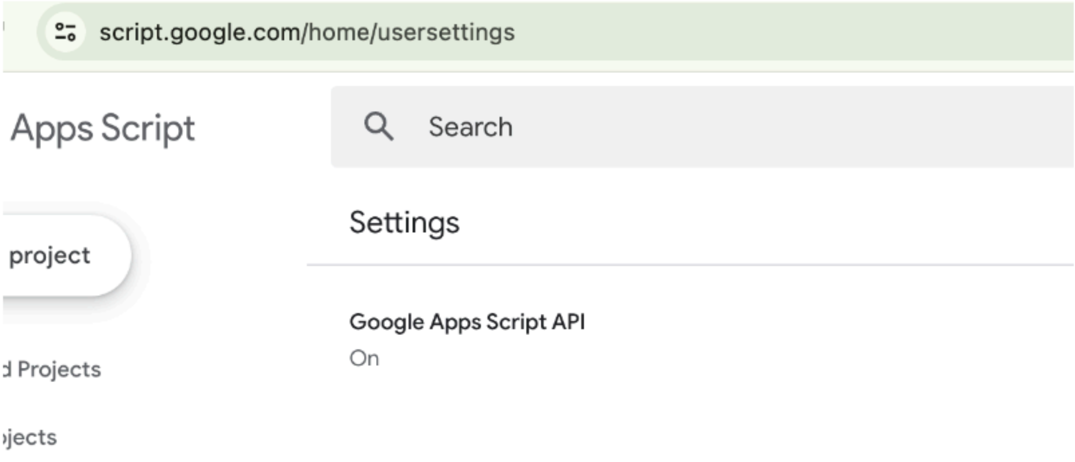


```
function myFunction() {

}
```

Note : You might need to switch this to ON

https://script.google.com/home/usersettings -> set this to On



script.google.com/home/usersettings

Settings

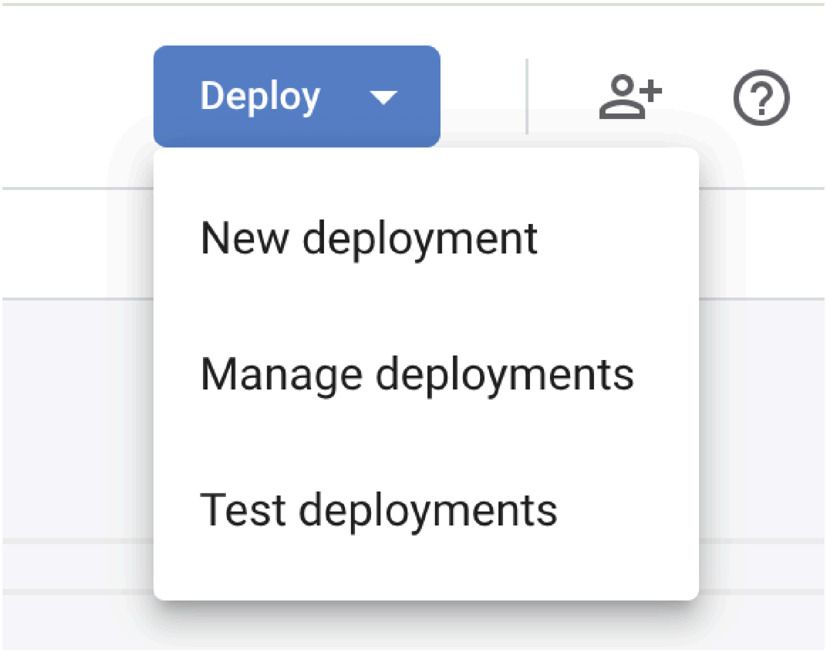
Google Apps Script API
On

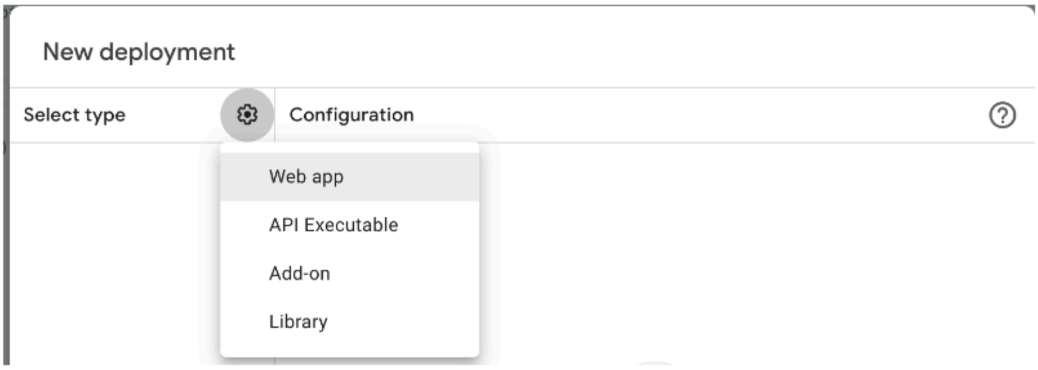
Otherwise, deployments might not work.

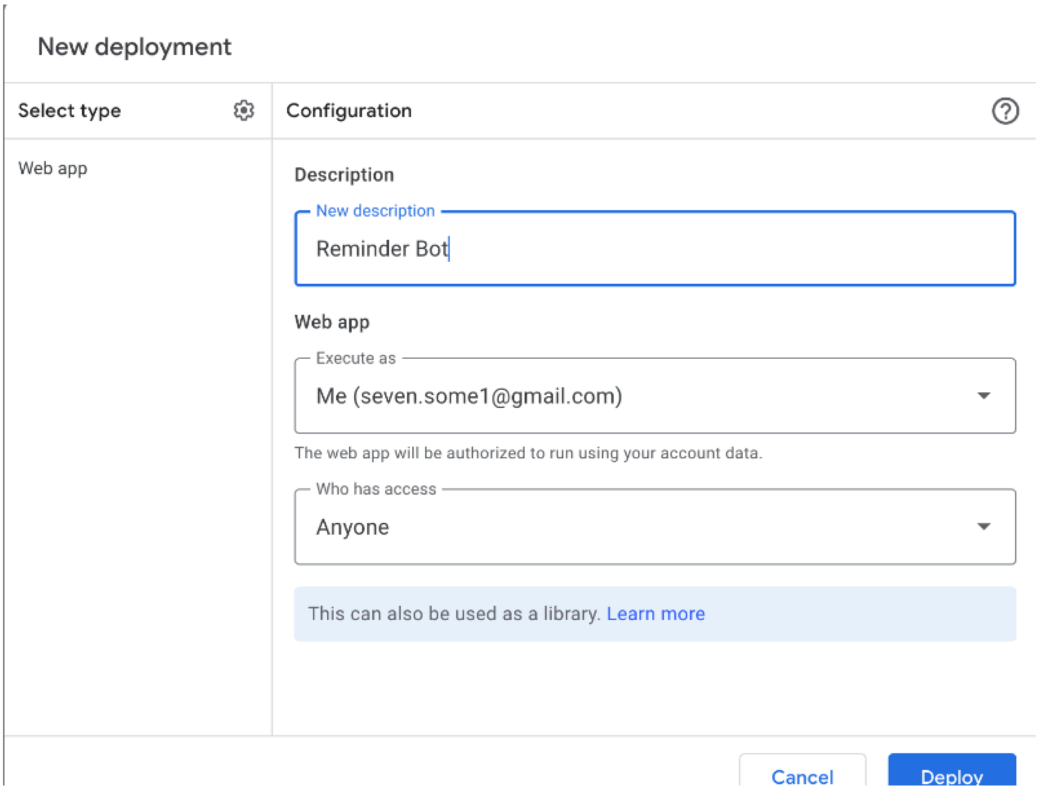
Deployments

To the right of the screen is a Deploy button in Blue.
Click on that

Select New Deployment , type as Web app


Deploy
New deployment
Manage deployments
Test deployments


New deployment
Select type
Configuration
Web app
API Executable
Add-on
Library



New deployment

Select type

Web app

Configuration

Description

New description: Reminder Bot

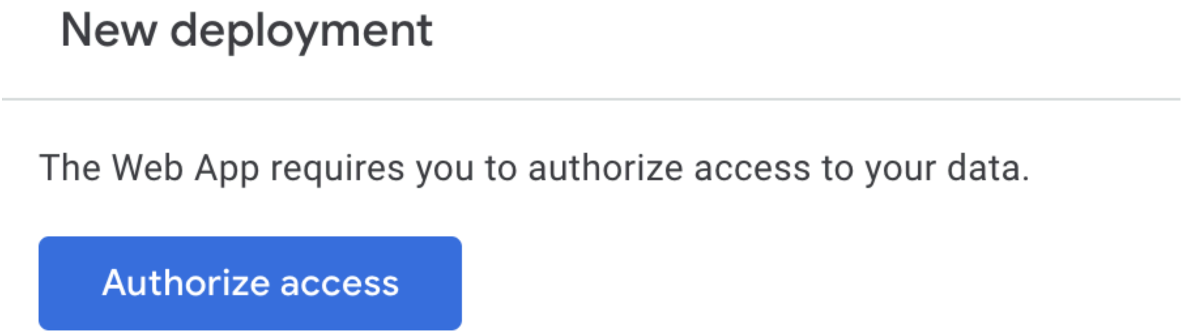
Web app

Execute as: Me (seven.some1@gmail.com)

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

Who has access: Anyone

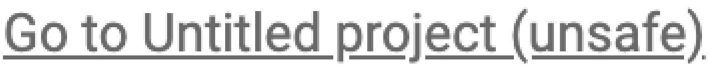
This can also be used as a library. Learn more



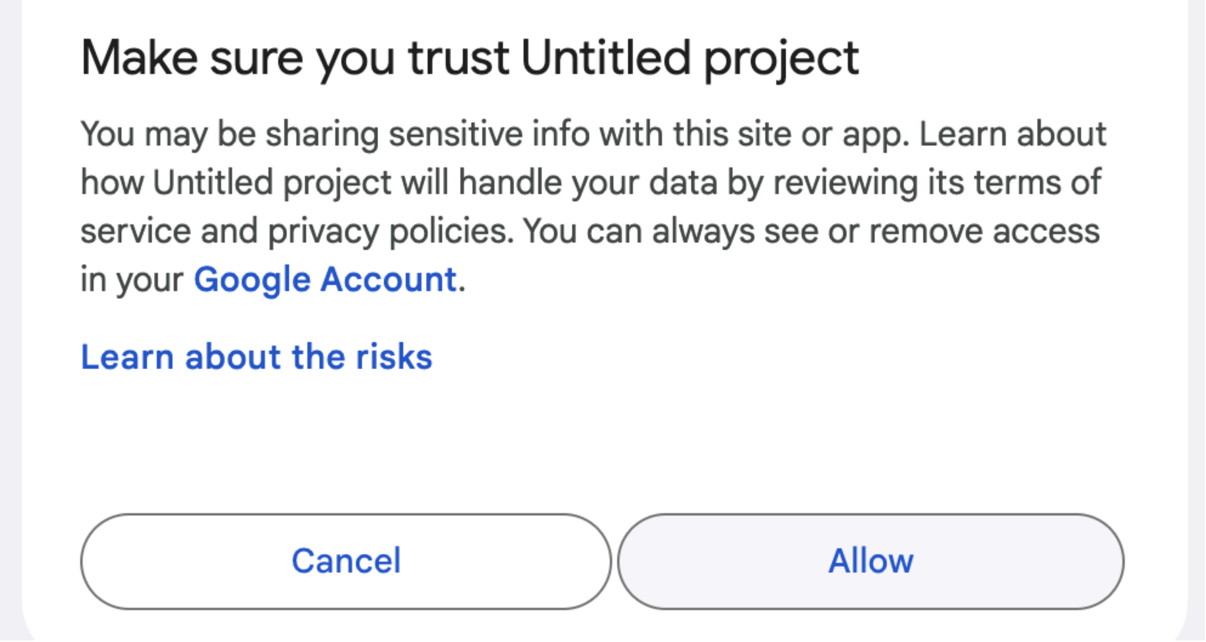
New deployment

The Web App requires you to authorize access to your data.

Advanced -> Authorize



Go to Untitled project (unsafe)

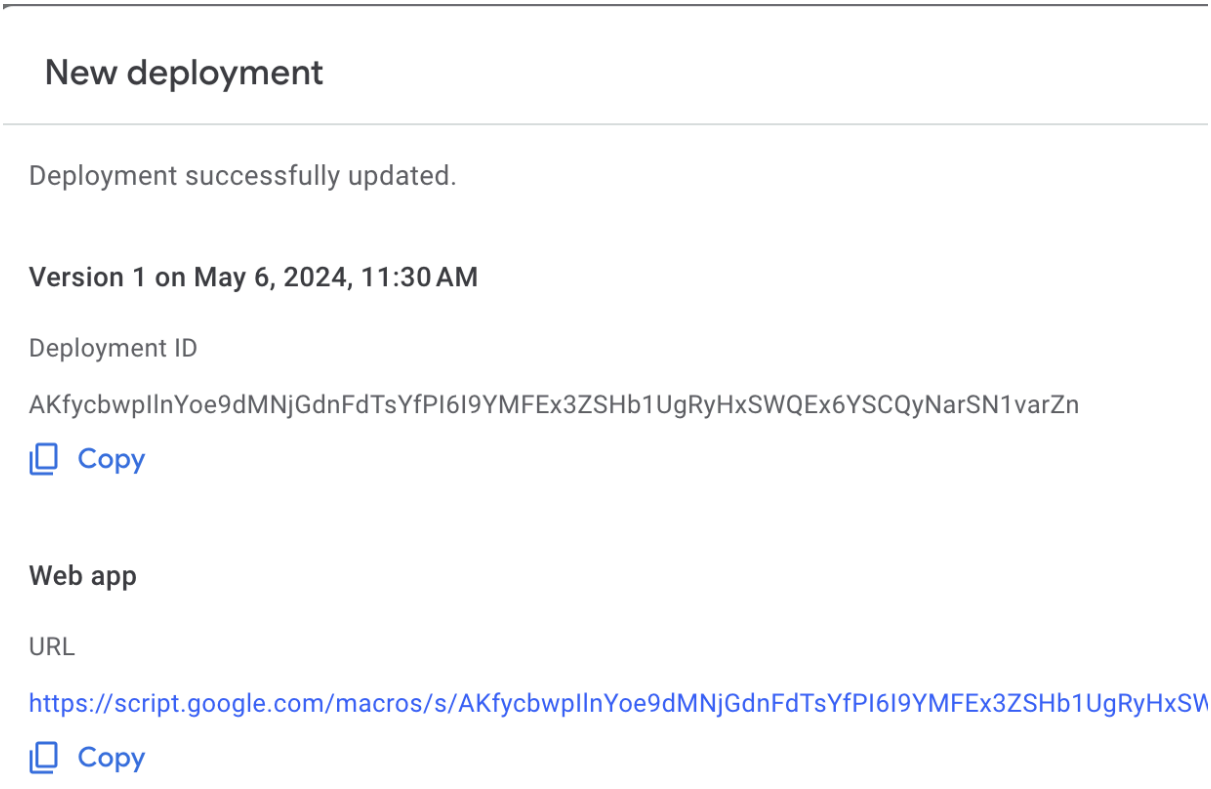


Make sure you trust Untitled project

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

Learn about the risks

Copy the deployment ID



New deployment

Deployment successfully updated.

Version 1 on May 6, 2024, 11:30 AM

Deployment ID

AKfycbwpIlnYoe9dMNjGdnFdTsYfPI6I9YMFEx3ZSHb1UgRyHxSWQEx6YSCQyNarSN1varZn

Web app

URL

https://script.google.com/macros/s/AKfycbwpIlnYoe9dMNjGdnFdTsYfPI6I9YMFEx3ZSHb1UgRyHxSW

Setup the Google App script code

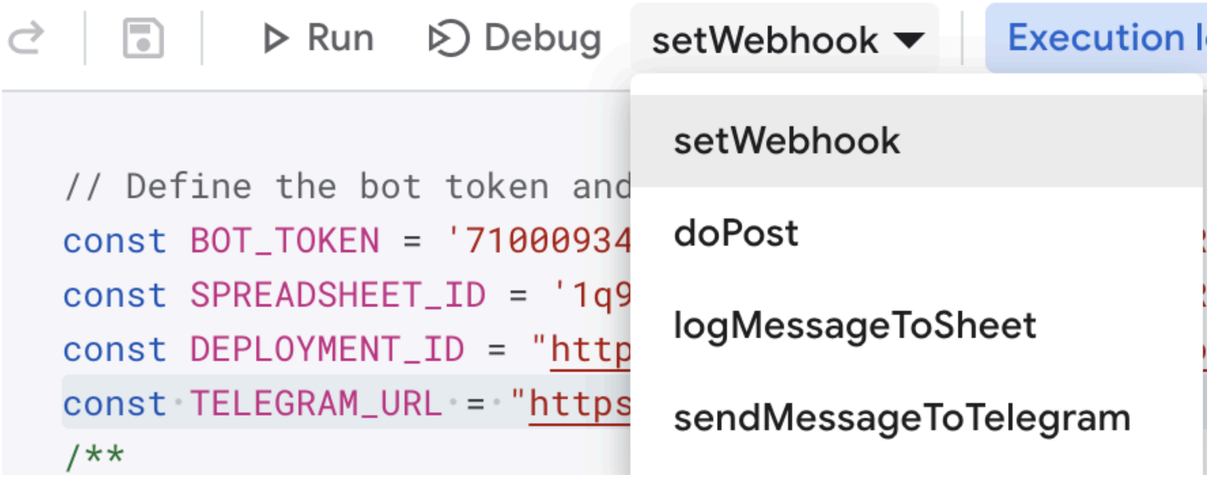
In your Code.gs

```
const BOT_TOKEN = '7100093471:AAGj59wvHA4b_co';  // this is received from Botfather
const SPREADSHEET_ID = '1q9f7joDYmbt_8BwLDmds';  // this is from google sheet https://docs.google.com/spreadsheets/d/{}/edit#gid=0
const DEPLOYMENT_ID = "https://script.google.com/macros/s/AKfycbwpIarSN1varZn/exec" ; // you get this from the deployment
const TELEGRAM_URL = "https://api.telegram.org/bot" + BOT_TOKEN;
```

Remember to save your project / file (Control + S)

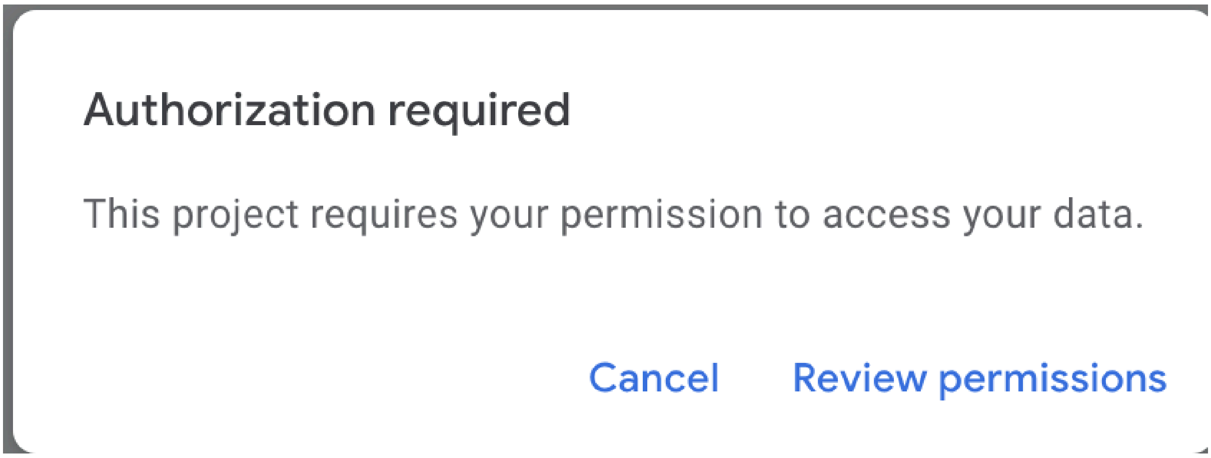
```
function setWebhook() {
var url = TELEGRAM_URL + "/setWebhook?url=" + DEPLOYMENT_ID;
var response = UrlFetchApp.fetch(url);
Logger.log(response);
}
```

Run your webhook to establish communication between Telegram bot and Google App script



```
// Define the bot token and
const BOT_TOKEN = '71000934
const SPREADSHEET_ID = '1q9
const DEPLOYMENT_ID = "http
const TELEGRAM_URL = "https
/**
```

It might ask for Permission

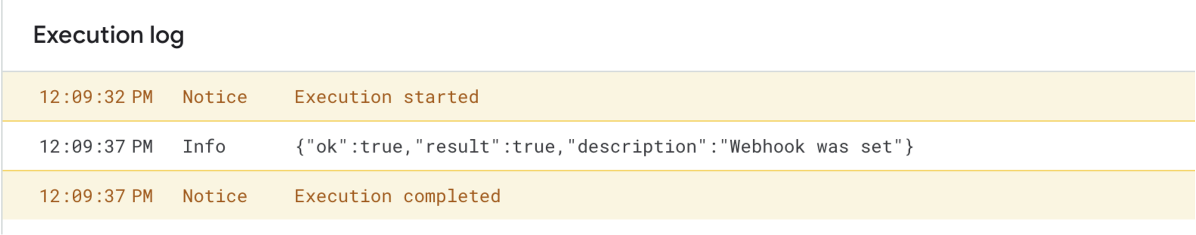


Authorization required

This project requires your permission to access your data.

Review Permission - Read through and Allow.
Warning : Be very cautious if you are allowing other bots and do only after you understand the risks completely.

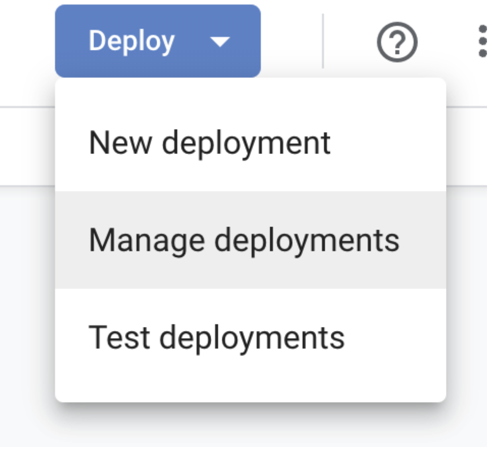
You should get a response like below :



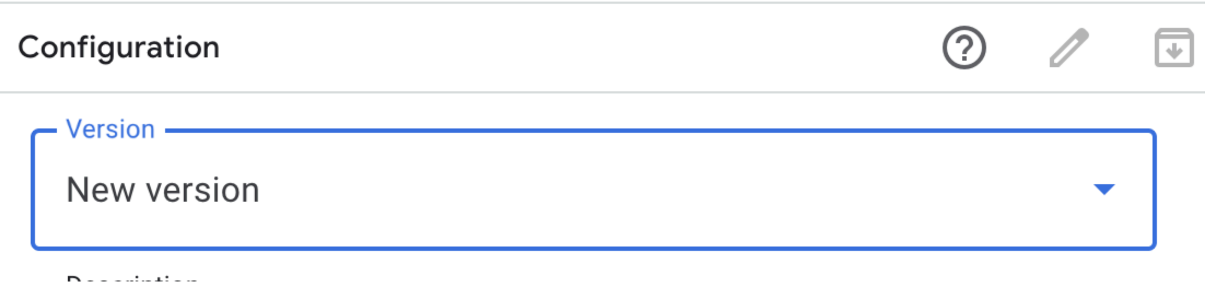
Execution log

```
12:09:32 PM   Notice    Execution started
12:09:37 PM   Info      {"ok":true,"result":true,"description":"Webhook was set"}
12:09:37 PM   Notice    Execution completed
```

Re-deploy changed code



Click on the pencil icon and select New Version



Version

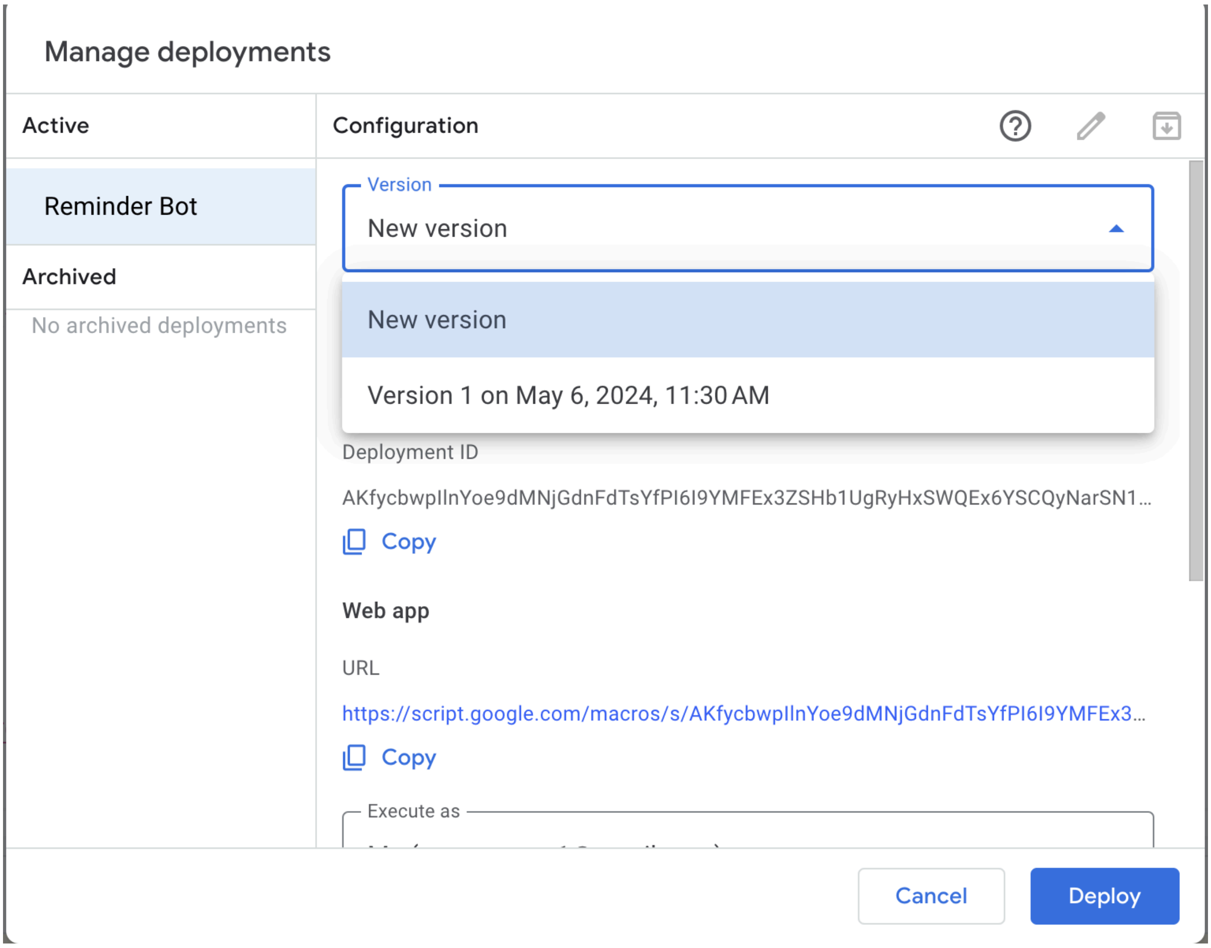
New version

Deploy.

Communicate from Google app script -> Telegram bot

```
/**
* Send message to Telegram bot from Google App script
*/
function sendMessage(chat_id, text) {
var url = TELEGRAM_URL + "/sendMessage?chat_id=" + chat_id + "&text=" +
text;
var response = UrlFetchApp.fetch(url);
Logger.log(response);
}
```

Redeploy



Manage deployments

Active

Reminder Bot

Archived

No archived deployments

Configuration

Version: New version

New version

Version 1 on May 6, 2024, 11:30 AM

Deployment ID

AKfycbwpIlnYoe9dMNjGdnFdTsYfPI6I9YMFEx3ZSHb1UgRyHxSWQEx6YSCQyNarSN1…

Web app

URL

https://script.google.com/macros/s/AKfycbwpIlnYoe9dMNjGdnFdTsYfPI6I9YMFEx3…

Execute as

Save your code, Click on Edit and Deploy New Version

Test sending messages from the Google script to Telegram

This is s slightly convoluted step, but listing down the steps nevertheless

1. deleteWebHook
2. Send any message from Telegram bot
3. Run the following method
4. Set webhook back again

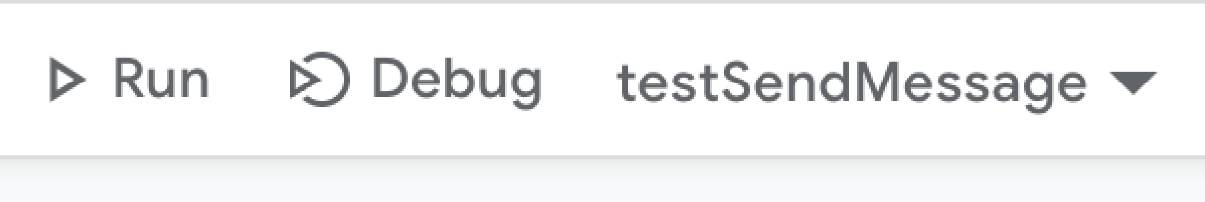
```
function deleteWebHook(){
 var url = TELEGRAM_URL + "/deleteWebhook?url=" + DEPLOYMENT_ID;
var response = UrlFetchApp.fetch(url);
Logger.log(response);
}
```

```
/**
* Send message to Telegram bot from Google App script
*/
function sendMessage(chat_id, text) {
var url = TELEGRAM_URL + "/sendMessage?chat_id=" + chat_id + "&text=" + text;
var response = UrlFetchApp.fetch(url);
Logger.log(response);
}
```

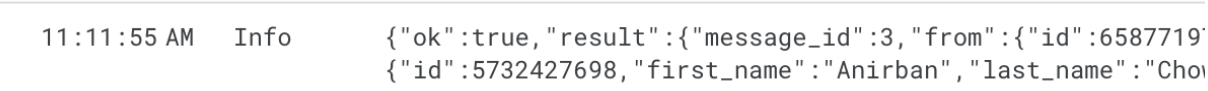
```
function getChatId() {
 var url = 'https://api.telegram.org/bot' + BOT_TOKEN + '/getUpdates';
 var response = UrlFetchApp.fetch(url);
 var responseData = JSON.parse(response.getContentText());
 if (responseData.ok && responseData.result.length > 0) {
   var chatId = responseData.result[0].message.chat.id;
   Logger.log('Chat ID: ' + chatId);
   return chatId;
 } else {
   Logger.log('No messages received');
   return null;
 }
}

/**
* Send message to Telegram bot from Google App script
*/
function testSendMessage() {
sendMessage(getChatId(),"Hi");
}
```

Run testSendMessage() from Appscript UI



You will see a response like below -



```
11:11:55 AM   Info      {"ok":true,"result":{"message_id":3,"from":{"id":6587719
                        {"id":5732427698,"first_name":"Anirban","last_name":"Cho
```

If all well, you should see a Hi on your Telegram Bot.

Communicate from Telegram Bot -> Google App script and Google sheet

This will save whatever entered in the Telegram bot in the google sheet referred earlier and throw back to the Telegram bot whatever was entered.

```
/**
* Handles the incoming webhook requests from the Telegram bot.
* @param {Object} e The event object containing the Telegram update.
* @returns {string} A response to the Telegram bot.
*/
function doPost(e) {
   // Parse the incoming JSON data
   const content = JSON.parse(e.postData.contents);

   // Check if the update contains a message
   if (content.message) {
     const chatId = content.message.chat.id;
     const text = content.message.text;

     // Log the message to the Google Sheet
     logMessageToSheet(chatId, text);
     // Send a response back to the user
     sendMessage(chatId,  text);
   }
}

/**
* Logs the message to the Google Sheet.
* @param {number} chatId The chat ID of the user.
* @param {string} text The message text.
*/
function logMessageToSheet(chatId, text) {
const sheet =
SpreadsheetApp.openById(SPREADSHEET_ID).getSheetByName('Sheet1');// Change
const lastRow = sheet.getLastRow() + 1;
sheet.getRange(lastRow, 1).setValue(new Date());
sheet.getRange(lastRow, 2).setValue(chatId);
sheet.getRange(lastRow, 3).setValue(text);
}
```

Final Modifications to the script

1. Change from chat_id to firstname to make the storage a little bit friendlier and you can even share the same bot with your friends.
2. Validations to guard against incorrect values entered
3. Calculate liters of water drank today

```
/**
* Handles the incoming webhook requests from the Telegram bot.
* @param {Object} e The event object containing the Telegram update.
* @returns {string} A response to the Telegram bot.
*/
```

```
function doPost(e) {
  // Parse the incoming JSON data
  const content = JSON.parse(e.postData.contents);

  // Check if the update contains a message
  if (content.message) {
    const chatId = content.message.chat.id;
    const firstname = content.message.from.first_name;
    const text = content.message.text;
    if(isPositiveNumberLessThan1point5(text)){
       // Log the message to the Google Sheet
        logMessageToSheet(firstname, text);

       // Send a response back to the user
       litersDrankToday = sumValuesForToday();
       sendMessage(chatId, "You entered = " + text +" . You drank "+ litersDrankToday+" litres today "  );
     }
    else {



     // Send a response back to the user
     sendMessage(chatId, "You entered = " + text + " which seems to be not a positive number less than 1.5, hence rejecting your input. We expect max of 1.5 l can be drunk at one time");
    }
  }
}


/**Utility Functions START */

function isPositiveNumberLessThan1point5(text) {
 // Convert the text to a number
 var number = parseFloat(text);
```

```
  // Check if the text is a positive number and less than 1.5
 if (!isNaN(number) && number > 0 && number < 1.5) {
   return true;
 } else {
   return false;
 }
}


function sumValuesForToday() {
 const sheet = SpreadsheetApp.openById(SPREADSHEET_ID).getSheetByName('Sheet1'); // Change
 const today = new Date();
 const values = sheet.getDataRange().getValues();
  // Filter values for today
 const filteredValues = values.filter(row => {
   const rowDate = new Date(row[0]);
   return rowDate.toDateString() === today.toDateString();
 });
  // Sum values for column 3
 const sum = filteredValues.reduce((acc, row) => acc + row[2], 0);
  Logger.log("Sum of values for today: " + sum);
 return sum;
}

/**Utility Functions END */
```