## 

[**What are we building today ? 1**](#_jfsust95508u)

[**Pre-requisite 1**](#_d921kcwmubjq)

[Setup a telegram bot 1](#_6bukt9bqfyar)

[Open a new google sheet 2](#_j5ey8nks4erd)

[Open the code editor from Extensions/App Script 2](#_eagojscx4l2e)

[**Deployments 3**](#_4sg64kqhgmql)

[**Setup the Google App script code 7**](#_f0o781b44aye)

[Run your webhook to establish communication between Telegram bot and Google App script 7](#_o4gtkauljmik)

[Re-deploy changed code 8](#_wwkdtxc0mpsm)

[**Communicate from Google app script -> Telegram bot 9**](#_v6cuacnztk8k)

[**Redeploy 9**](#_uju5yfwt29d)

[Test sending messages from the Google script to Telegram 10](#_3z41mg519qsy)

[**Communicate from Telegram Bot -> Google App script and Google sheet 12**](#_mdp24qk0837j)

[Final Modifications to the script 13](#_gbw3exb48izi)

## 

## What are we building today ?

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

DRAFT workflow :

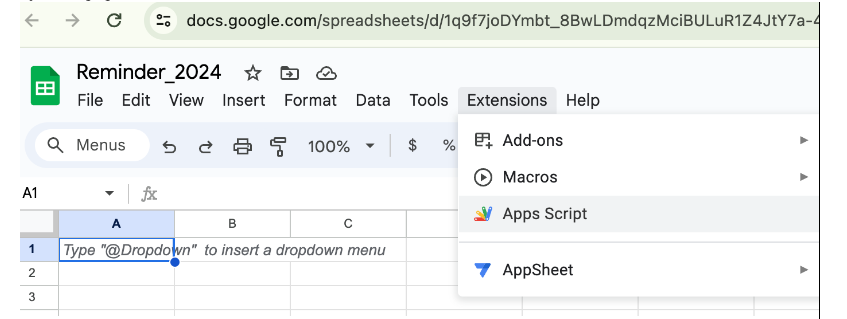
1. User enters liters of water
2. Check for error conditions (Not a positive number, >1.5 liters etc.)
3. Add content to google sheet
4. Finish off with a message highlighting liters of water drank today.

## Pre-requisite

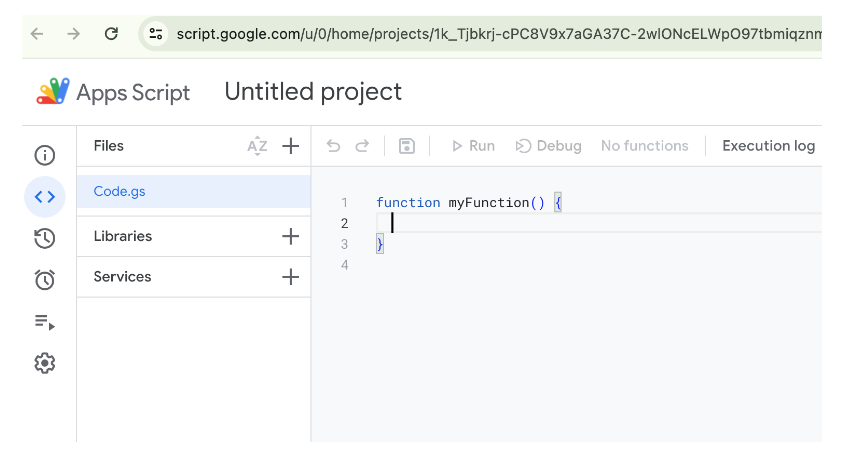
### Setup a telegram bot

From BotFather / newbot , Collect the token

### Open a new google sheet

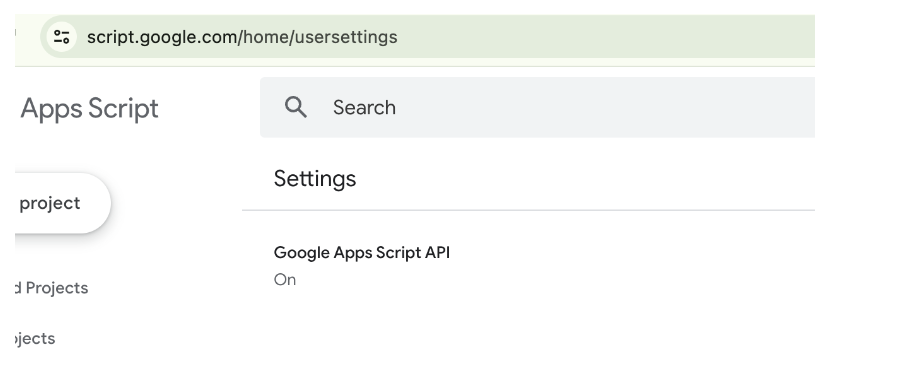


### Open the code editor from Extensions/App Script



Note : You might need to switch this to ON

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

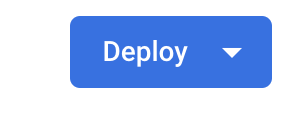
****

Otherwise, deployments might not work.

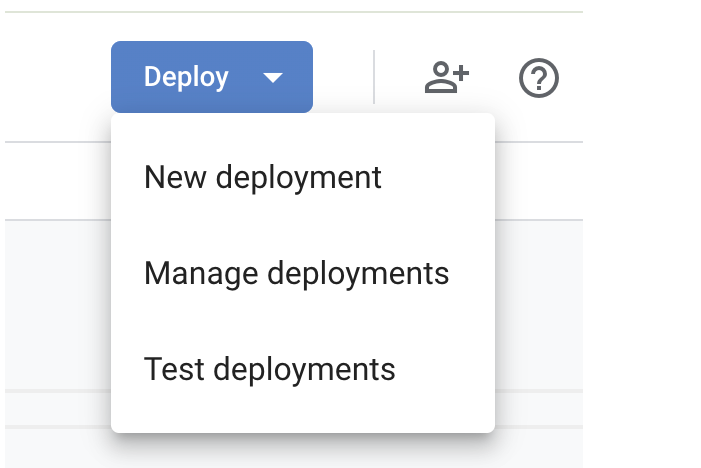
## Deployments

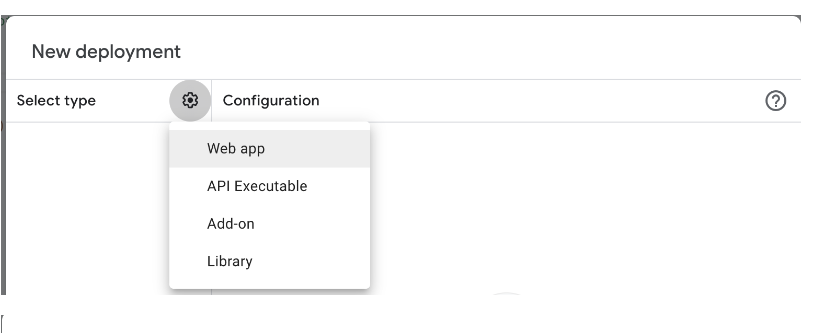
To the right of the screen is a Deploy button in Blue.

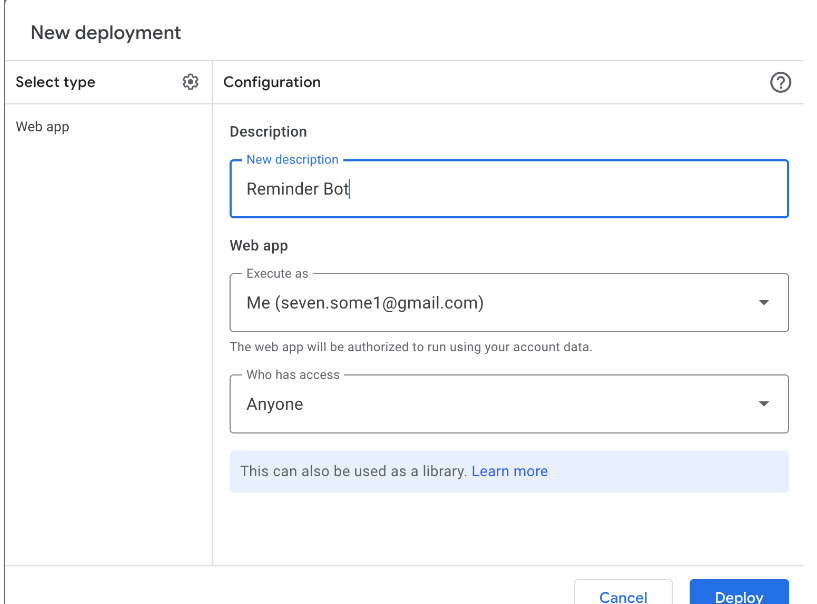
Click on that

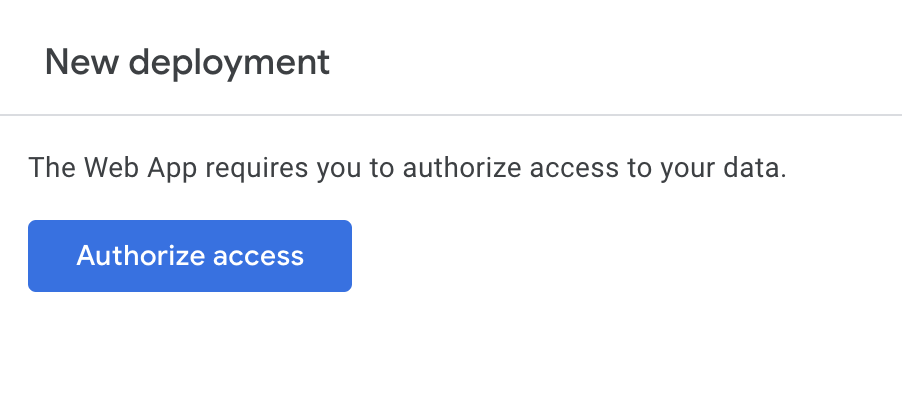


Select New Deployment , type as Web app



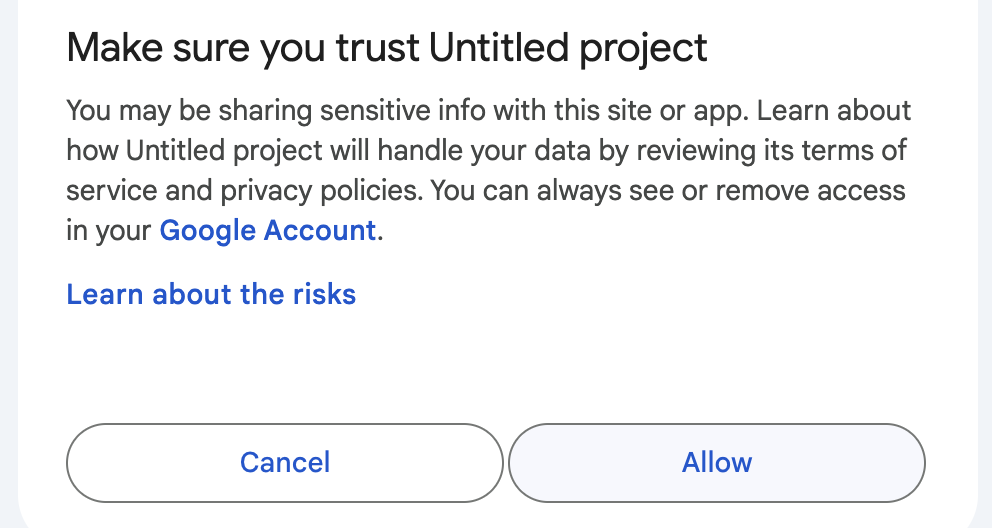




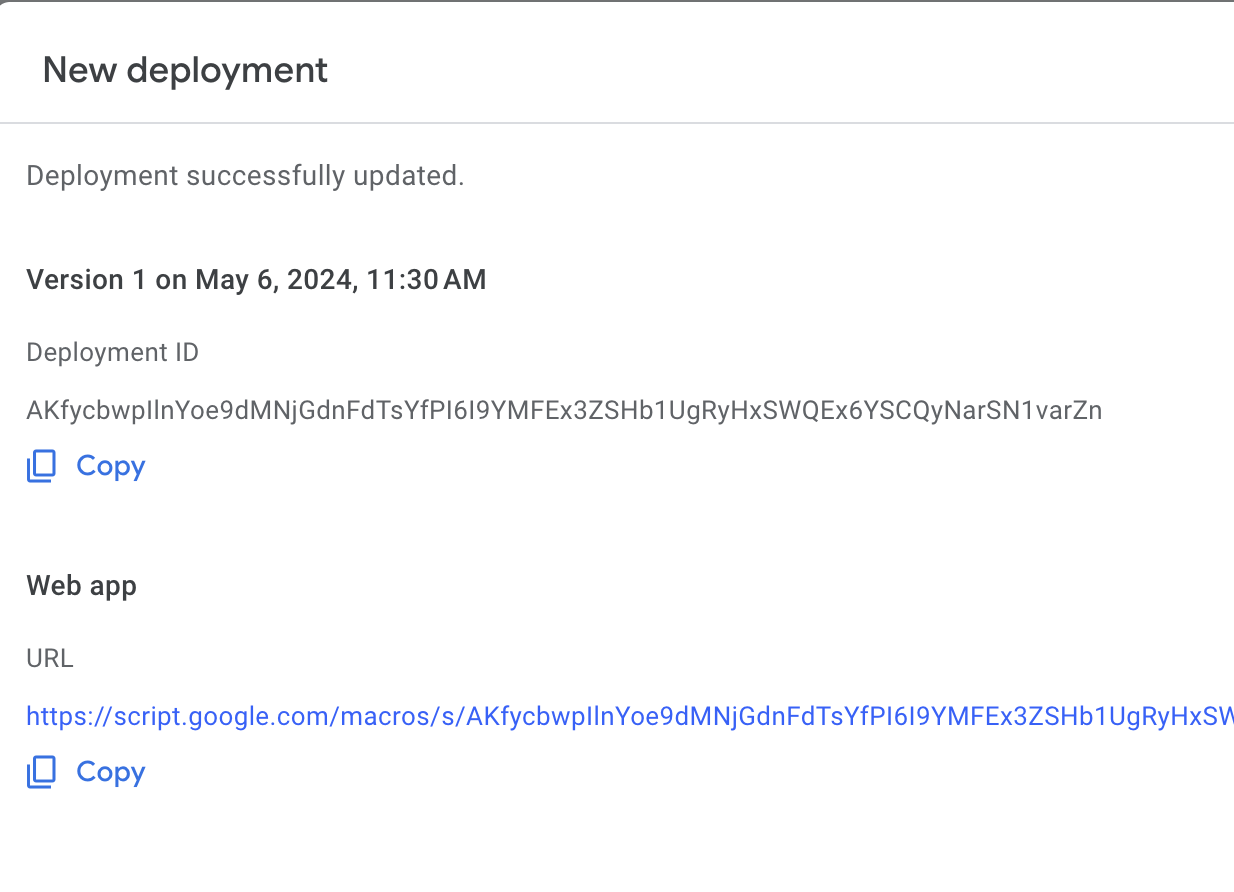


Advanced -> Authorize





Copy the deployment ID



## 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() {

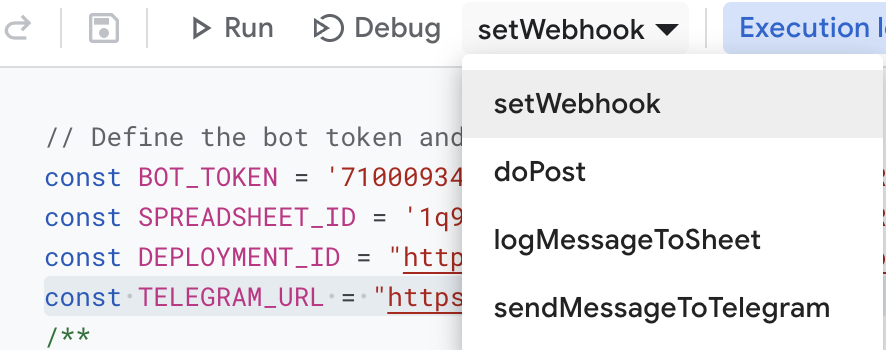
var url = TELEGRAM\_URL + "/setWebhook?url=" + DEPLOYMENT\_ID;

var response = UrlFetchApp.fetch(url);

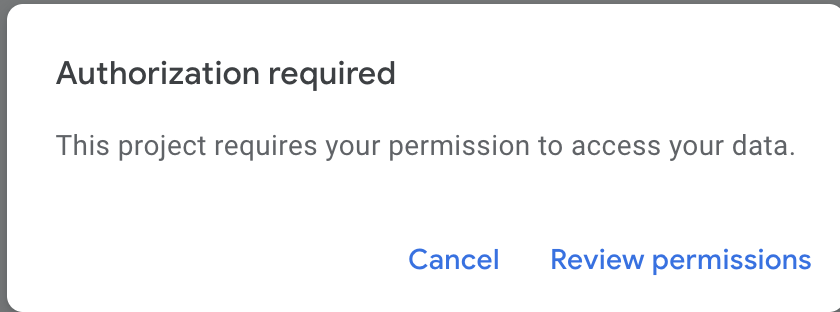
Logger.log(response);

}

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



It might ask for Permission



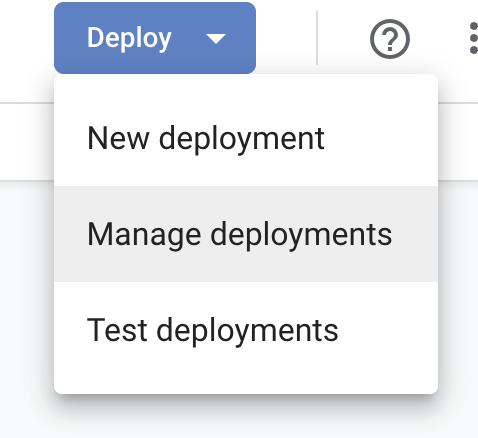
Review Permission - Read through and Allow.

Warning : Be very cautious if you are allowing other bots and do only after you understand the risks completely.

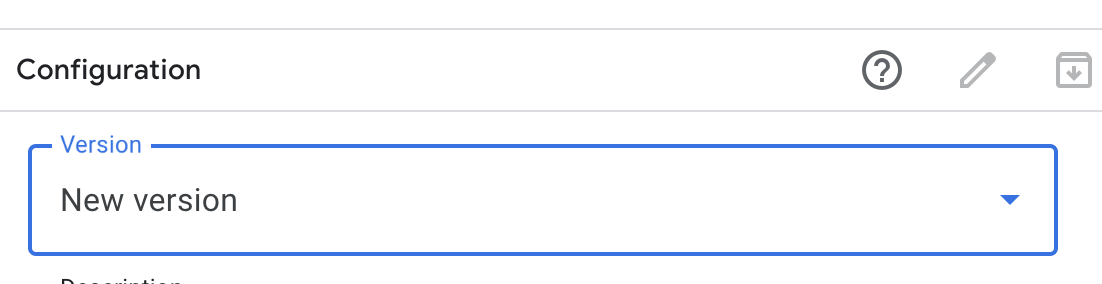
You should get a response like below :



### Re-deploy changed code



Click on the pencil icon and select New Version



Deploy.

## Communicate from Google app script -> Telegram bot

/\*\*

\* Send message to Telegram bot from Google App script

\*/

function sendMessage(chat\_id, text) {

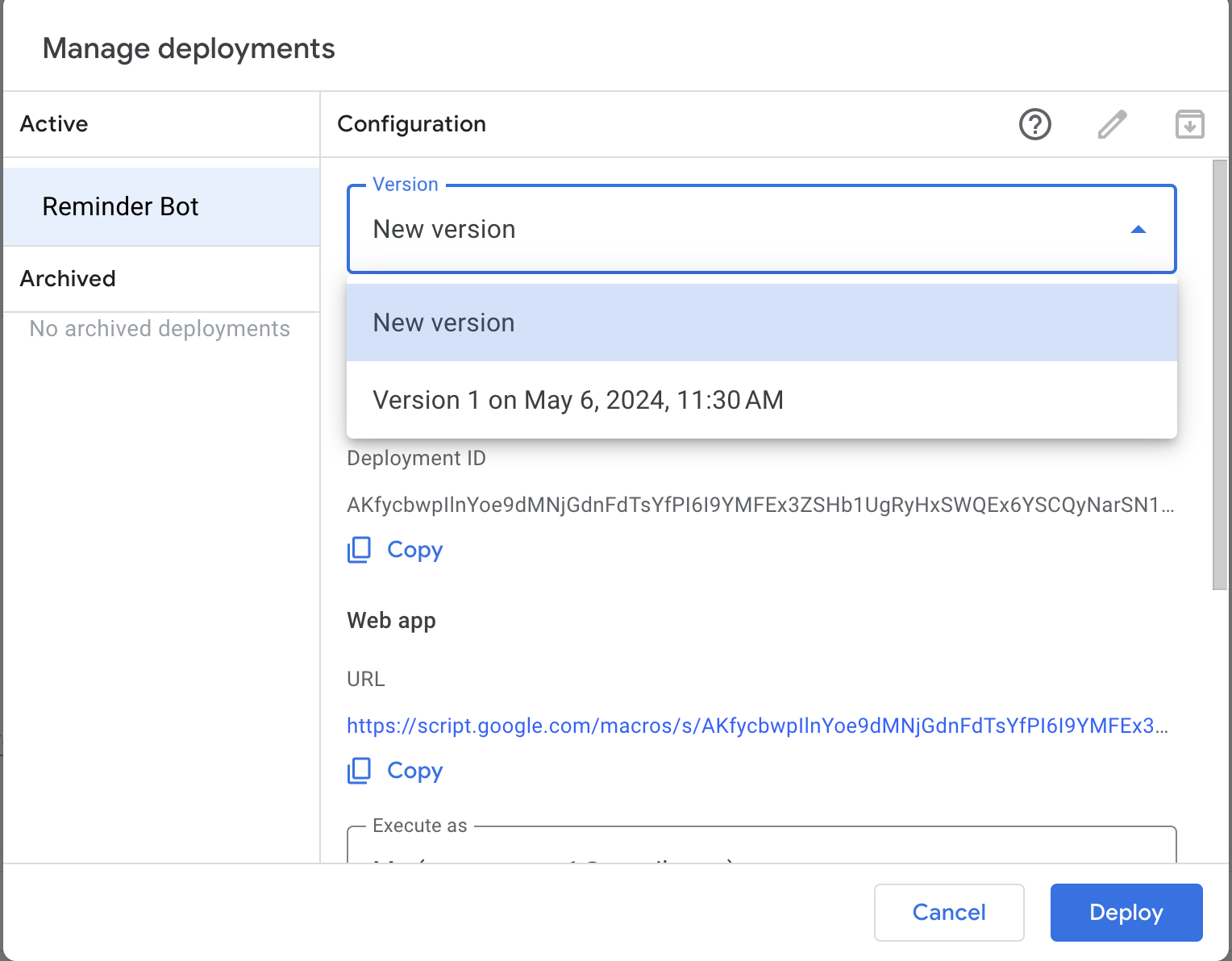
var url = TELEGRAM\_URL + "/sendMessage?chat\_id=" + chat\_id + "&text=" + text;

var response = UrlFetchApp.fetch(url);

Logger.log(response);

}

## Redeploy



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

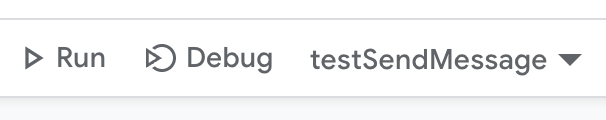
\*/

function testSendMessage() {

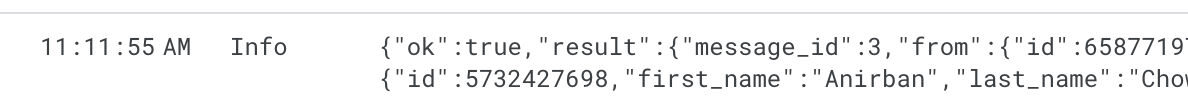
sendMessage(getChatId(),"Hi");

}

Run testSendMessage() from Appscript UI



You will see a response like below -



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 \*/