

# 07-1 Pushing Data to the Cloud via Sheets

Using Google Sheets with the Bone

# Steps to and from the Cloud

- ▶ Create a new Sheet
- ▶ Get Credentials
- ▶ Run the script
- ▶ Plot the data

# Create a Sheet

- ▶ Go to <https://sheets.google.com>
- ▶ Make a new sheet
- ▶ Rename it
- ▶ Add some column headings

The screenshot shows a web browser window with the Google Sheets interface. The title bar indicates the document is 'Logging to Sheets'. The address bar shows the URL 'https://docs.google.com/s...'. The top navigation bar includes a 'SHARE' button and a user profile icon. The main menu bar contains 'File', 'Edit', 'View', 'Insert', 'Format', and 'Data'. The toolbar shows various editing tools and a zoom level of 100%. The spreadsheet grid has columns labeled A through E and rows numbered 1 through 10. The first row (row 1) contains the following data: 'Time' in column A, 'Var 1' in column B, and 'Var 2' in column C. The bottom status bar shows 'Sheet1' and 'Count: 3'.

	A	B	C	D	E
1	Time	Var 1	Var 2		
2					
3					
4					
5					
6					
7					
8					
9					
10					

# Note the sheetID



# Get Credentials

```
bone$ cd exercises/iot/google/sheets
```

- ▶ JavaScript: <https://developers.google.com/sheets/api/quickstart/nodejs>
- ▶ Python: <https://developers.google.com/sheets/api/quickstart/python>

## Step 1: Turn on the Google Sheets API

Click this button to create a new console project and automatically enable the Google Sheets API:

ENABLE THE GOOGLE SHEETS API

This opens a new dialog. In the dialog, do the following:

- Select **+ Create a new project**.
- Download the configuration file.
- Move the downloaded file to your working directory and ensure it is named `credentials.json`

## Step 2: Install the client library

Run the following commands to install the libraries using npm:

```
$ npm install googleapis@27 --save
```

## Step 1: Turn on the Google Sheets API

Click this button to create a new console project and automatically enable the Google Sheets API:

ENABLE THE GOOGLE SHEETS API

This opens a new dialog. In the dialog, do the following:

- Select **+ Create a new project**.
- Download the configuration file.
- Move the downloaded file to your working directory and ensure it is named `credentials.json`.

## Step 2: Install the Google Client Library

Run the following command to install the library using pip:

```
$ pip install --upgrade google-api-python-client oauth2client
```

# Get Credentials

- ▶ You will download the file `credentials.json`
- ▶ Put it in the sheets directory

# Run the example

- To log 3.14 and 10 run:

```
bone$ ./demo.js 3.14 10
```

Authorize this app by visiting this url:

```
https://accounts.google.com/o/oauth2/v2/auth?access_type=offline&scope=http%3A%2F%2Fwww.googleapis.com%2Fauth%2Fspreadsheets&response_type=code&client_id=834314057708-fucjrsgnm1nvmvzbqkfr9008g63d7k2r.apps.googleusercontent.com&redirect_uri=urn%3Aietf%3Awg%3Aoauth%3A2.0%3Aoob
```

Enter the code from that page here:

- Paste the URL into your browser and follow the directions.

 Sign in with Google



Choose an account

to continue to **Logging Data**



Mark Yoder

mark.a.yoder@gmail.com



Use another account

 Sign in with Google

**Logging Data** wants to access  
your Google Account



mark.a.yoder@gmail.com

This will allow **Logging Data** to:



View and manage your spreadsheets in Google  
Drive



Make sure you trust **Logging Data**

You may be sharing sensitive info with this site or app.

Sign in

Please copy this code, switch to your application and paste it there:

4/dAAX8\_DywtUOVZYoGXzah9ffdezpDI4J0dOeq-xn1kA0mz3h7rRjAJM

Eng



# Run the example

- To log 3.14 and 10 run:

```
bone$ ./demo.js 3.14 10
```


Authorize this app by visiting this url:

```
https://accounts.google.com/o/oauth2/v2/auth?access_type=offline&scope=http%3A%2F%2Fwww.googleapis.com%2Fauth%2Fspreadsheets&response_type=code&client_id=834314057708-fucjrsgnm1nvmvzbqkfr9008g63d7k2r.apps.googleusercontent.com&redirect_uri=urn%3Aietf%3Awg%3Aoauth%3A2.0%3Aoob
```


Enter the code from that page here:

- Paste the URL into your browser and follow the directions.


# Run the €

 Sign in with Google

**Logging Data** wants to access  
your Google Account

 mark.a.yoder@gmail.com

This will allow **Logging Data** to:

☒ View and manage your spreadsheets in Google Drive 

**Make sure you trust Logging Data**

You may be sharing sensitive info with this site or app.  
Learn about how Logging Data 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](#)

Cancel [Allow](#)

# Run the example

- To log 3.14 and 10 run:

```
bone$ ./demo.js 3.14 10
```

Authorize this app by visiting this url:

```
https://accounts.google.com/o/oauth2/v2/auth?access_type=offline&scope=http%3A%2F%2Fwww.googleapis.com%2Fauth%2Fspreadsheets&response_type=code&client_id=834314057708-fucjrsgnm1nvmvzbqkfr9008g63d7k2r.apps.googleusercontent.com&redirect_uri=urn%3Aietf%3Awg%3Aoauth%3A2.0%3Aoob
```






Enter the code from that page here:

- Paste the URL into your browser and follow the directions.





# token.json

- ▶ A file **token.json** is created
- ▶ Don't have to authorize again

# View the Data





 Logging to Sheets ☆    

File Edit View Insert Format Data Tools Add-ons Help [All changes saved in Drive](#)

100% \$ % .0 .00 123 Arial 10 B I S A    

*fx* | 10/10/2018 16:12:51

	A	B	C	D	E	F	G	H	I
1	Time	Var 1	Var 2						
2	10/10/2018 15:59:07	5	10						
3	10/10/2018 16:04:22	10	5						
4	10/10/2018 16:05:45	10	5						
5	10/10/2018 16:10:22	-5	-10						
6	10/10/2018 16:11:22	3.14	6.28						
7	10/10/2018 16:12:51	3.14	6.28						
8	10/10/2018 16:14:16	6.28	3.14						
9									
10									
11									
12									
13									
14									
15									
16									
17									
18									
19									
20									
21									

  Sheet1  

# Plot the data

The screenshot shows the Google Sheets interface with the 'Format' menu open. The 'Number' option is highlighted in the main menu, and the 'Date time' option is highlighted in the submenu. The spreadsheet data is as follows:

	A	B	C
1	Time	Var 1	Var 2
2	10/10/2018 12:05	1	2
3	10/10/2018 12:05	10	20
4	10/10/2018 14:05	5	10
5			
6			
7			
8			
9			
10			

The 'Format' menu options include: Number, Bold (Ctrl+B), Italic (Ctrl+I), Underline (Ctrl+U), Strikethrough (Alt+Shift+5), Font size, Align, Merge cells, Text wrapping, Text rotation, Automatic, Plain text, Number (1,000.12), Percent (10.12%), Scientific (1.01E+03), Accounting (\$ (1,000.12)), Financial ((1,000.12)), Currency (\$1,000.12), Currency (rounded) (\$1,000), Date (9/26/2008), Time (3:59:00 PM), **Date time** (9/26/2008 15:59:00), and Duration (24:01:00).

- ▶ Select all columns
- ▶ Select **Insert chart**

[illegible]



Logging to Sheets



File Edit View Insert Format Data Tools Add-ons Help [All changes saved in Drive](#)



SHARE



100% \$ % .0 .00 123 Arial 10 B I S A

fx

Time

Var 1 and Var 2

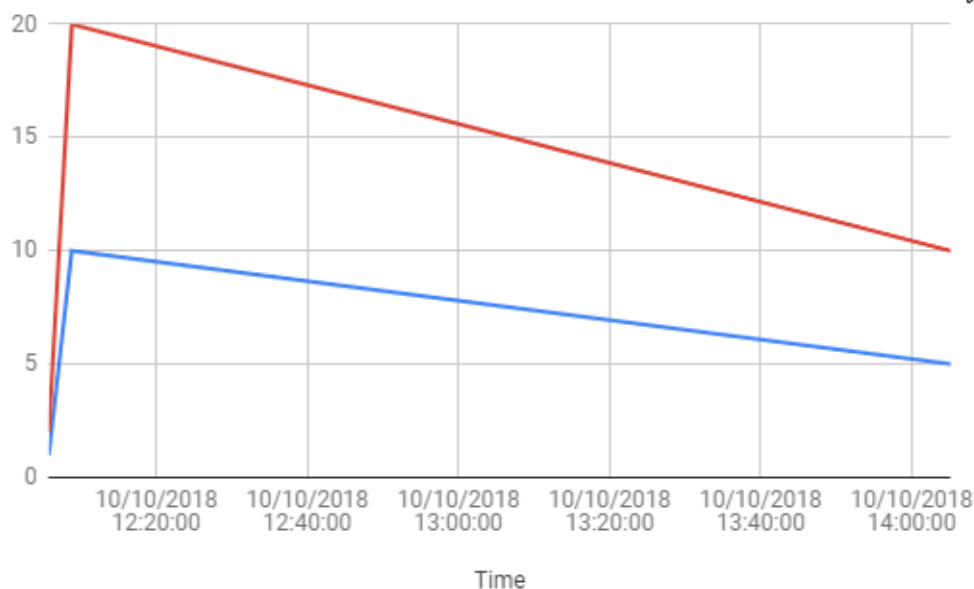


Chart editor

DATA

CUSTOMIZE

Chart type

Line chart

Data range

A1:C1003

X-axis

Time

Series

123 Var 1

123 Var 2

Add Series

☐ Switch rows / columns

☒ Use row 1 as headers

☒ Use column A as labels

☐ Aggregate column A

+ Sheet1

Sum: 6/19/2256 14:19...



# Plot the Data

- ▶ Your data should be plotted
- ▶ Running `demo.js` again will add more data

# The code - JavaScript

- ▶ Lot's of code for authentication

```
#!/usr/bin/env node
```

```
// From:
```

```
https://developers.google.com/sheets/api/quickstart/nodejs#step\_3\_set\_up\_the\_sample
```

```
const fs = require('fs');
```

```
const util = require('util');
```

```
const readline = require('readline');
```

```
const {google} = require('googleapis');
```

```
const sheetID = '1hGMHMLwiG3zEDM19zJehpLjTDbVi8LOjVKhD8R0dB00';
```

```
// If modifying these scopes, delete credentials.json.
```

```
const SCOPES = ['https://www.googleapis.com/auth/spreadsheets'];
```

```
const TOKEN_PATH = 'token.json';
```

# The code - JavaScript

```
function recordTemp(auth) {  
    const sheets = google.sheets({version: 'v4', auth});  
    var date = new Date();  
    // Read the variable data  
    var var1 = process.argv[2];  
    var var2 = process.argv[3];  
    console.log('Got: ' + var1 + ", " + var2);  
}
```

# The code - JavaScript

```
sheets.spreadsheets.values.append({
  spreadsheetId: sheetID,
  range: 'A2',
  // How the input data should be interpreted.
  valueInputOption: 'USER_ENTERED',
  resource: {
    values: [ // getTime returns ms. Convert to days.
      [
        date.getTime()/1000/60/60/24 + 25569 - 4/24,
        var1,
        var2
      ]
    ]
  }
})
```

# The code - JavaScript

```
    sheets.spreadsheets.values.append({  
...  
    }, (err, res) => {  
        if (err) return console.log('The API returned an error: ' + err);  
        console.log("res: " + util.inspect(res.data.tableRange));  
    });  
}
```

# The code - Python

```
# Call the Sheets API
# Compute a timestamp and pass the first two arguments
values = [ [time.time()/60/60/24+ 25569 - 4/24, sys.argv[1], sys.argv[2]]]
body = { 'values': values }
result =
service.spreadsheets().values().append(spreadsheetId=SPREADSHEET_ID,
                                       range=RANGE_NAME,
                                       # How the input data should be interpreted.
                                       valueInputOption='USER_ENTERED',
                                       # How the input data should be inserted.
                                       # insertDataOption='INSERT_ROWS'
                                       body=body
                                       ).execute()
```