



Sheets and Node

The Web is great but it's not the best, or the only, solution for some problems. In this post we'll discuss how to integrate [Google Sheets](#) as the data source and a Node.js as the front end, either through a CLI or via an Express-based server.

This project does not use [Apps Script](#). We're not building add ons for Sheets so the technology would not be appropriate.

Project Definition

We want to create a central repository for information about projects we're working on and we don't want to change the spreadsheet that we use just for the sake of a web app.

Management wants a public web interface while keeping the spreadsheet secure with only team access to write and edit.

Before we start

Follow the Steps 1 and 2 in the [Node.js Quickstart](#) to add the Sheets API to the project and download the necessary credentials file that we'll use in the code below.

Keep the `token.json` file in the root directory of the project. Without it the project will not work, even in a read-only situation.

Caution

Do not add the `token.json` file to your Git repo. Doing so can lead to very expensive and fraudulent charges.

To make use of the API we need to install Google APIs libraries. Use the following command:

```
npm i googleapis@39
```

Yes, we've pinned a specific version to make sure that we don't get bit by unexpected changes to the API unless you want to make the change.

The Code

Cli

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

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

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

```
// Load client secrets from a local file.
fs.readFile('credentials.json', (err, content) => {
  'credentials.json'console.log('Error loading client secret file:', err);
  'Error loading client secret file://' Authorize a client with credentials
  authorize(JSON.parse(content), listProjects);
});

function authorize(credentials, callback) {
  const {client_secret, client_id, redirect_uris} = credentials.installed
  const oAuth2Client = new google.auth.OAuth2(
    client_id, client_secret, redirect_uris[0]);

  // Check if we have previously stored a token.
  if (err) return getNewToken(oAuth2Client, callback);
  oAuth2Client.setCredentials(JSON.parse(token));
  callback(oAuth2Client);
};
}
```

```

function getNewToken(oAuth2Client, callback) {
  const authUrl = oAuth2Client.generateAuthUrl({
    access_type: 'offline',
    scope: SCOPES,
  });
  console.log('Authorize this app by visiting this url:', authUrl);
  const rl = readline.createInterface({
    input: process.stdin,
    output: process.stdout,
  });
  rl.question('Enter the code from that page here: ', (code) => {
    rl.close();
    oAuth2Client.getToken(code, (err, token) => {
      if (err) return console.error('Error while trying to retrieve access token');
      oAuth2Client.setCredentials(token);
      // Store the token to disk for later 'offline' // Store the token to disk for later 'offline'
      fs.writeFile(TOKEN_PATH, JSON.stringify(token), (err) => {
        if (err) return console.error(err);
        console.log('Token stored to', TOKEN_PATH);
      });
      callback(oAuth2Client);
    });
  });
}

```

```

function listProjects(auth) {
  const sheets = google.sheets({version: 'v4', auth});
  // Replace the spreadsheetId value and range
  // with those from your own spreadsheet
  sheets.spreadsheets.values.get({
    spreadsheetId: '1JFRg7uFQCCvcGcRBeLTDliogVI-eb7YGW0q5XfE-tdE',
    range: 'Class Data!A2:E',
  }, (err, res) => {
    if (err) return console.log('The API returned an error: ' + err);
    '1JFRg7uFQCCvcGcRBeLTDliogVI-eb7YGW0q5XfE-tdE'gth) {
      console.log('project, leader:');
      rows.map((row) => {

```

```
        console.log(`${row[0]}, ${row[4]}`);
    });
    } else {
        console.log('No data found.');
```

```
    }
    });
}
```

```
'project, leader:'`${row[0]}, ${row[4]}`
```

Express-based service

- <https://github.com/theoephraim/node-google-spreadsheet>
- <https://codelabs.developers.google.com/codelabs/sheets-api>