

This is a proposed workflow/framework for development of Google APPs scripts here at DialogTech.

Prerequisites

1. Not required, however all of the testing for this environment was performed on Chrome.
2. Node version .12.x or later installed on your machine
3. Node module for integrating with Google Drive installed

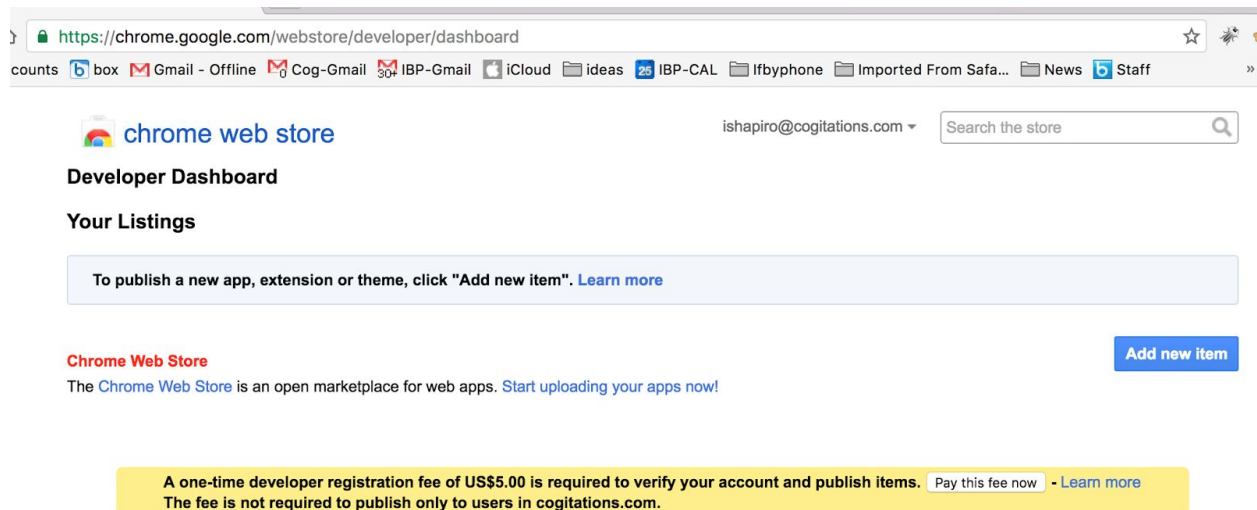
```
npm install -g node-google-apps-script
```

4. The documentation for node-google-apps-script is located at:

<https://github.com/danthareja/node-google-apps-script>

Getting Started

1. Set up your Google Developer Console environment
2. <https://chrome.google.com/webstore/developer/dashboard>
3. Before accessing this link from Chrome you need to sign Chrome into a gmail accounts.



4. The first time you access this link you will be prompted to pay a \$5 registration fee. You need to do this to gain access to all of the resources you will need to create a Google Add-On and use the Google APIs.

5. Create a Google Developer Console Project

<https://console.developers.google.com/iam-admin/projects>

6. I named my project DialogTech Google Drive Access
7. Under the Library Tab -- add google drive API access. Click on enable.
8. Now click on Credentials on the left side of the page.

VERY IMPORTANT. Click on OAuth Consent Screen on the top of the page.
DO NOT CLICK ON create credentials.

9. The only field you need on the OAuth page is the email address. Then click on SAVE.
10. Now click on Create Credentials from the main screen.

Select OAuth Client ID.

Select other as the type.

Given the credentials a name.

Click on create.

11. Download the client secret file to your Google Scripts Directory as a json file. You cannot recreate this file without starting over so do not lose it.

Authenticating gapps (gapps is the node to google application)

1. Open a terminal window in your Google Scripts Directory on your Mac. (I use the "Go2Shell.app" OSX finder extension to make this easy to do.
2. Execute the following unix command (where the client_secret ... is the file you downloaded)

```
gapps auth client_secret_130360900166-8g4u8u6d4br0bcpvu0qnf34mmfka7csr.apps.googleusercontent.com.json
```

3. Copy the URL displayed and open it up in Chrome. This will display an OAuth security page.
4. Click on allow.

OK. You are done the hard part.

Now Start/Initialize Your First Project

1. Navigate to your Apps Script project from Google Drive (must be a [standalone script](#))
2. Get your project ID from the address bar, located after /d/ and before /edit.

- For example,
`'//script.google.com/a/google.com/d/abc123-xyz098/edit?usp=drive_web'`

3. Navigate to a directory where your Apps Script project will live
4. Run `gapps init <fileId>` within your project directory.
5. For example, `gapps init abc123-xyz098`
6. This will have create a `.json` configuration file and also a `src` directory. All of your modules will be in the `src` directory renamed from `.gs` to `.js`. When you upload the files back to google they will be renamed back to `.gs`.

Ready for Coding Locally

1. Edit the local copy of your code in the `src` directory that was created by the `gapps init` command.
2. When you are ready to update the Google Drive copy just type from the root directory not from the `src` directory. The `gapps` app looks for the `json` config file in the current directory.

`gapps upload`

3. Like magic your copy of the script in Google Drive will be updated.

Usage Notes

1. The `gapps` tools does not currently have the ability to copy files back down from a google drive if you update them with the Google Apps UI. There is a pull request from a couple of months ago for this feature but I have not investigated if it is stable.
2. Therefore once you start using `gapps` you must do ALL of your editing locally, run `gapps upload`, then test in the Google APPs Script editor interface. If you find a bug, make sure you go back and edit locally and re-upload.
3. This is similar to using a DEV server when you would edit locally, ftp to test, and then edit locally. You can think of the Google APPs Script editor is the DEV server.