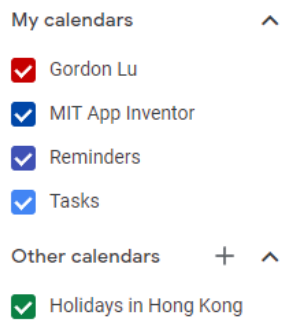
A. Obtaining your Calendar ID.

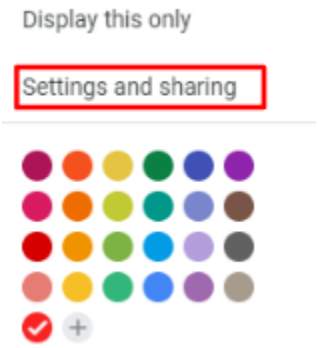
1. Open Google Calendar.

2. Locate the list of calendars on the left panel.

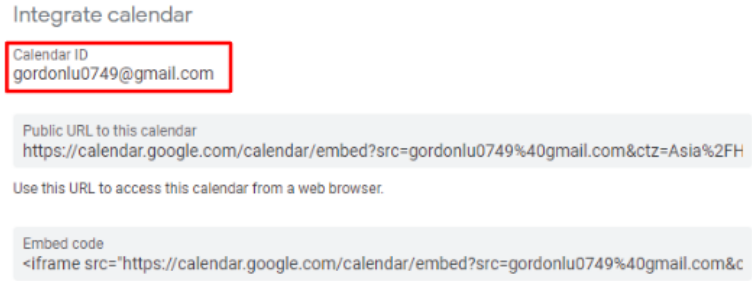


3. Hover over the specific calendar that you want to obtain and click the three dots menu. The three dots icon will only appear if you hover over the calendar item.

4. Select "Settings and sharing".



5. Scroll down and find a label that says "Calendar ID". Copy the calendar ID below it. For me, it is my email address, but not all calendars are the same as your email address.



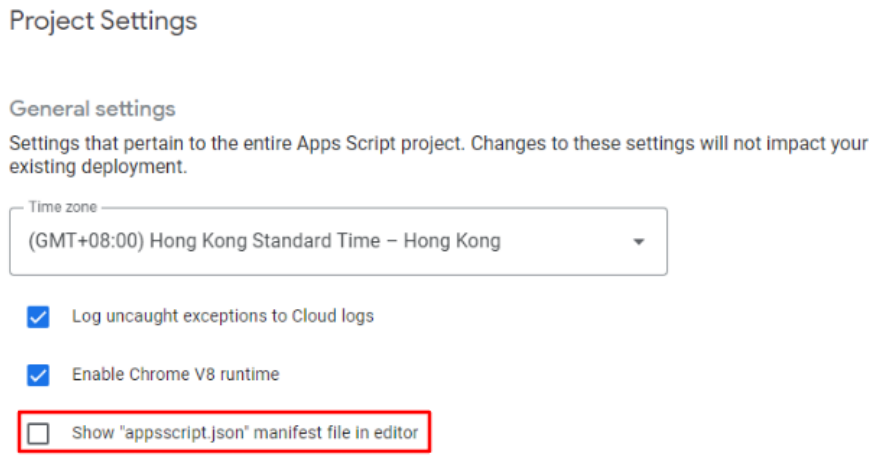
B. Deploying your Apps Script.

1. Create a new Apps Script project and clear all of the code already inside the script.

2. From the Downloads section, download the `Script.txt` file and copy all of its contents and paste it into the script. Click the Save button.

3. On the left panel, locate the gear icon and click on it.

4. You will be redirected to the settings of this project. Enable "Show 'appsscript.json' manifest file in editor".



5. Go back to the editor by clicking on the code icon on the left panel.

6. On the left control menu back in the script editor, you will see "appsscript.json" showing on top of "Code.gs". `Code.gs` is the Google Apps Script script that we code with normally, while `appsscript.json` is the manifest file of this apps script.

Android apps have an `AndroidManifest.xml` to define elements of this Android app, as you already know. This JSON file is the equivalent. Click on it.

You will see some code in the manifest editor. Yours might be different, but mine is this.

```
{
  "timeZone": "Asia/Hong_Kong",
  "dependencies": {
  },
  "exceptionLogging": "STACKDRIVER",
  "runtimeVersion": "V8"
}
```

According to the official Google documentation, if you are an editor of this calendar, no matter if you are the owner or no matter if you can share this calendar, you need two permissions: https://www.google.com/calendar/feeds and https://www.googleapis.com/auth/calendar. This includes both **read and write** permissions. Else, if you are a viewer and you do not have permission to edit the calendar, then you have to use the permission https://www.googleapis.com/auth/calendar.readonly. This **only** includes read permission.

We will have to add an element in this manifest file that will tell Google what OAuth Scopes (or if you do not understand permissions) we will use. So, we will add another element in the manifest.

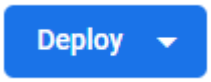
```
"oauthScopes": [
    "https://www.googleapis.com/auth/calendar",
    "https://www.google.com/calendar/feeds"
  ]
```

Replace this link if you do not need to edit the folder.

For me, the whole manifest would become:

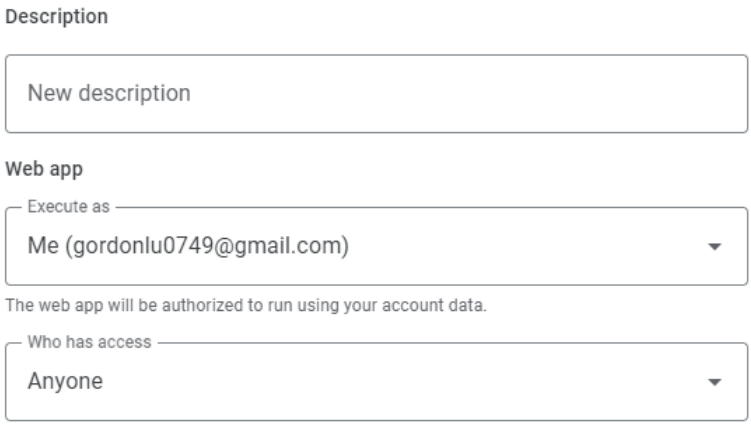
```
{
  "timeZone": "Asia/Hong_Kong",
  "dependencies": {
  },
  "exceptionLogging": "STACKDRIVER",
```

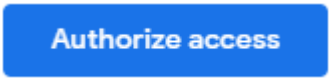
```
  "runtimeVersion": "V8",
"oauthScopes": [
    "https://www.googleapis.com/auth/calendar",
    "https://www.google.com/calendar/feeds"
  ]
}
```

7. Click on that  Save button again *in the manifest editor*. Now, the web app is ready to be deployed! To deploy the script, click on 

, select "New deployment", click on the  icon in the popup, and select "Web app".

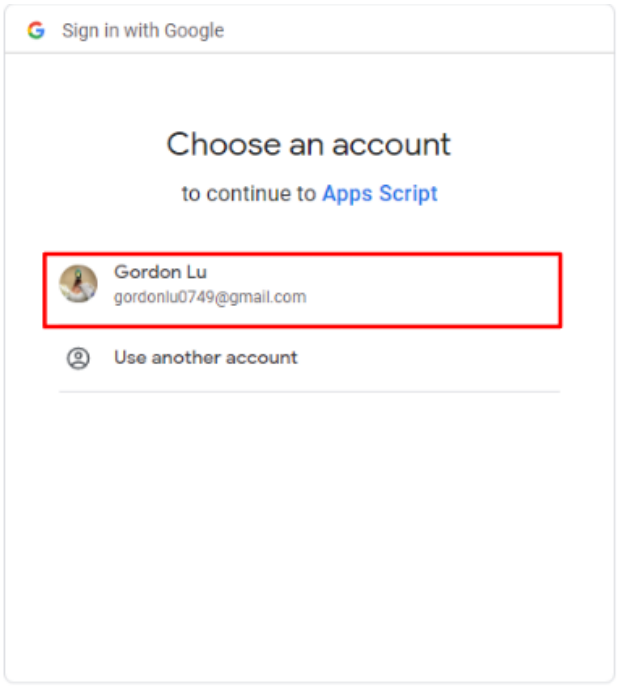
Make sure you are executing as **you** and **anyone have access to this script**. Then click Deploy.



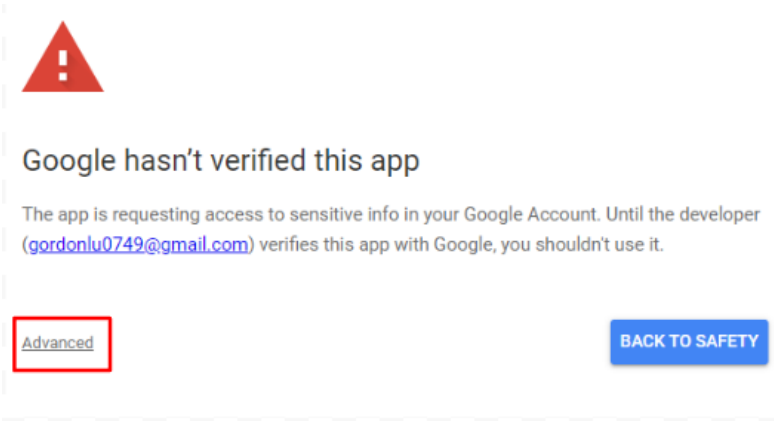
The web app **might** additionally ask you to authorize access. If so, click on 

, then a new popup would show up for you to log in.

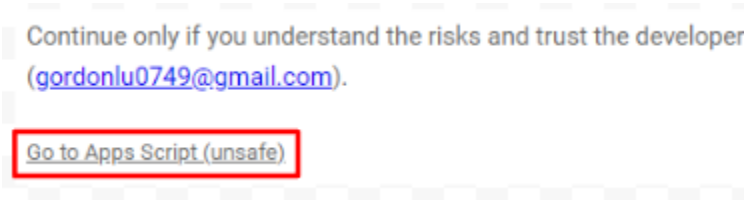
Click on your account in the popup.



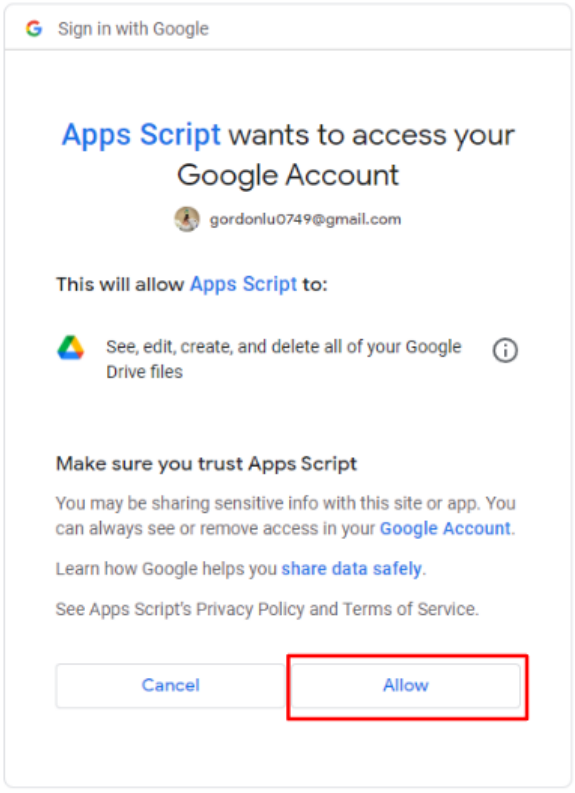
It will show that "Google hasn't verified this app". Don't worry, this script is totally safe. Click on `Advanced`



and



and



and the popup would close. Back in Apps Script, after it finished

, it would show the deployment ID and the deployment URL. We only need the URL, so click on Copy **for the URL** and head back to App Inventor.