Deployment

Introduction

If this Inventory System is to be implemented into a different location, the following steps must be taken. Please note that this system was designed for NUSTARS so some substantial change will be necessary.

Sectioning off your space

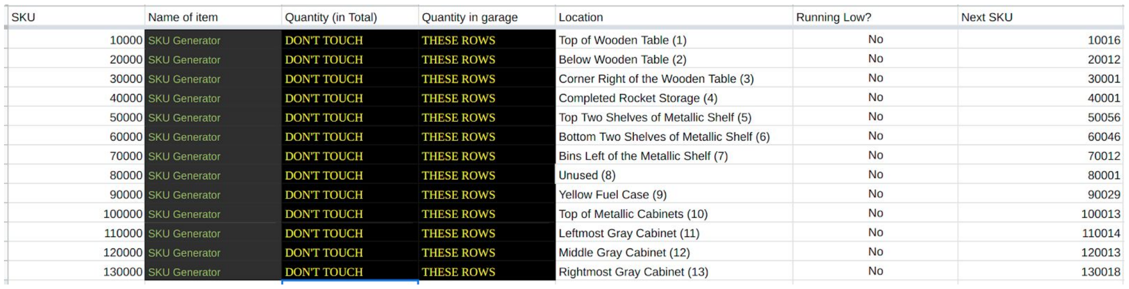
First, you must section your space off into several locations. These locations must be small enough so finding an item easier, but not too small, otherwise the users will be overwhelmed with the number of locations. For example, below is several sections of the NUSTARS Inventory.

The top of the table is one location, while below the table is another location. To the right of the table is another location.

Setting up the Spreadsheet and Forms for the RockeTracker

First obtain the Template Spreadsheet and Forms here: https://drive.google.com/open?id=1i14BAkBiI2-uU1SGIE4tgF1qOtPO9GSO.

Once you have made a copy of the Spreadsheet and the Forms, and saved them in your Drive, enter the Locations in this manner into each of the SKU generating rows: (The SKU generating rows are the ones labeled “DON’T TOUCH THESE ROWS”.)



SKU	Name of item	Quantity (in Total)	Quantity in garage	Location	Running Low?	Next SKU
10000	SKU Generator	DON'T TOUCH	THESE ROWS	Top of Wooden Table (1)	No	10016
20000	SKU Generator	DON'T TOUCH	THESE ROWS	Below Wooden Table (2)	No	20012
30000	SKU Generator	DON'T TOUCH	THESE ROWS	Corner Right of the Wooden Table (3)	No	30001
40000	SKU Generator	DON'T TOUCH	THESE ROWS	Completed Rocket Storage (4)	No	40001
50000	SKU Generator	DON'T TOUCH	THESE ROWS	Top Two Shelves of Metallic Shelf (5)	No	50056
60000	SKU Generator	DON'T TOUCH	THESE ROWS	Bottom Two Shelves of Metallic Shelf (6)	No	60046
70000	SKU Generator	DON'T TOUCH	THESE ROWS	Bins Left of the Metallic Shelf (7)	No	70012
80000	SKU Generator	DON'T TOUCH	THESE ROWS	Unused (8)	No	80001
90000	SKU Generator	DON'T TOUCH	THESE ROWS	Yellow Fuel Case (9)	No	90029
100000	SKU Generator	DON'T TOUCH	THESE ROWS	Top of Metallic Cabinets (10)	No	100013
110000	SKU Generator	DON'T TOUCH	THESE ROWS	Leftmost Gray Cabinet (11)	No	110014
120000	SKU Generator	DON'T TOUCH	THESE ROWS	Middle Gray Cabinet (12)	No	120013
130000	SKU Generator	DON'T TOUCH	THESE ROWS	Rightmost Gray Cabinet (13)	No	130018

If you need more locations, more SKU Generating rows will need to be made. Please insert a new row in the Inventory Spreadsheet directly below the last SKU Generating Row, and then copy and paste one of the SKU generating rows into that new row. Make the value in the SKU column equal to Location Number times 10000. Notice in the Value in the column labeled "Next SKU", you will find a formula similar to this.

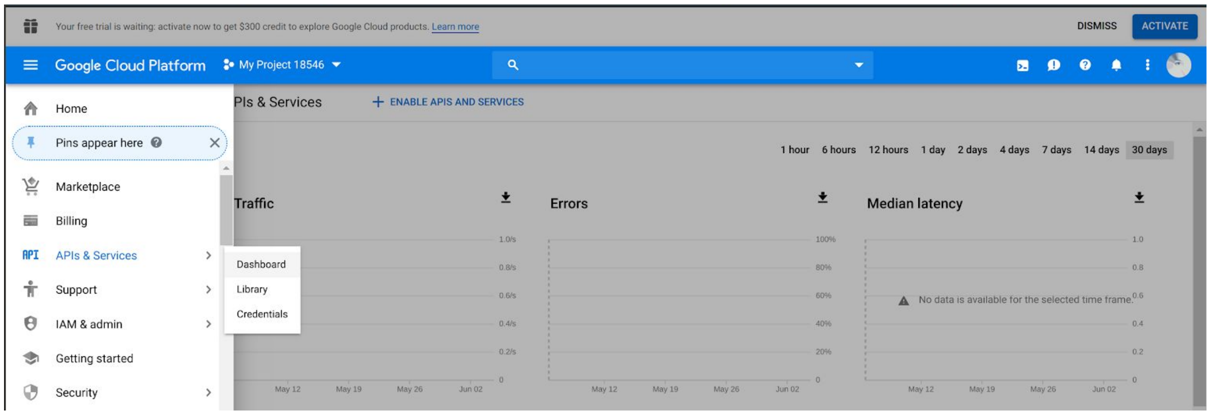
```
=MAX(FILTER(A:A, A:A >129999, A:A < 140000))+1
```

This formula calculates the next SKU that will be used when an item is added to the inventory. The formula above is for Location 13, since the range of values for Location 13 are between 129999 and 140000 exclusive. The first numeric value should one less that the value in the SKU column, and the second numeric value should be 10000 greater than the value in the SKU column.

Configuring the Google API

Log into this site with a **non-Northwestern** google account: https://console.cloud.google.com/. We cannot use a Northwestern account, since NUIT has disabled Google Cloud Services for student G Suite accounts. Press "Select a Project" in the upper left hand corner, and then press "New Project". Choose the name of the project, then press Create.

While you have the project selected, press the sandwich menu Icon in the upper left hand corner, then press APIs & Services > Library.

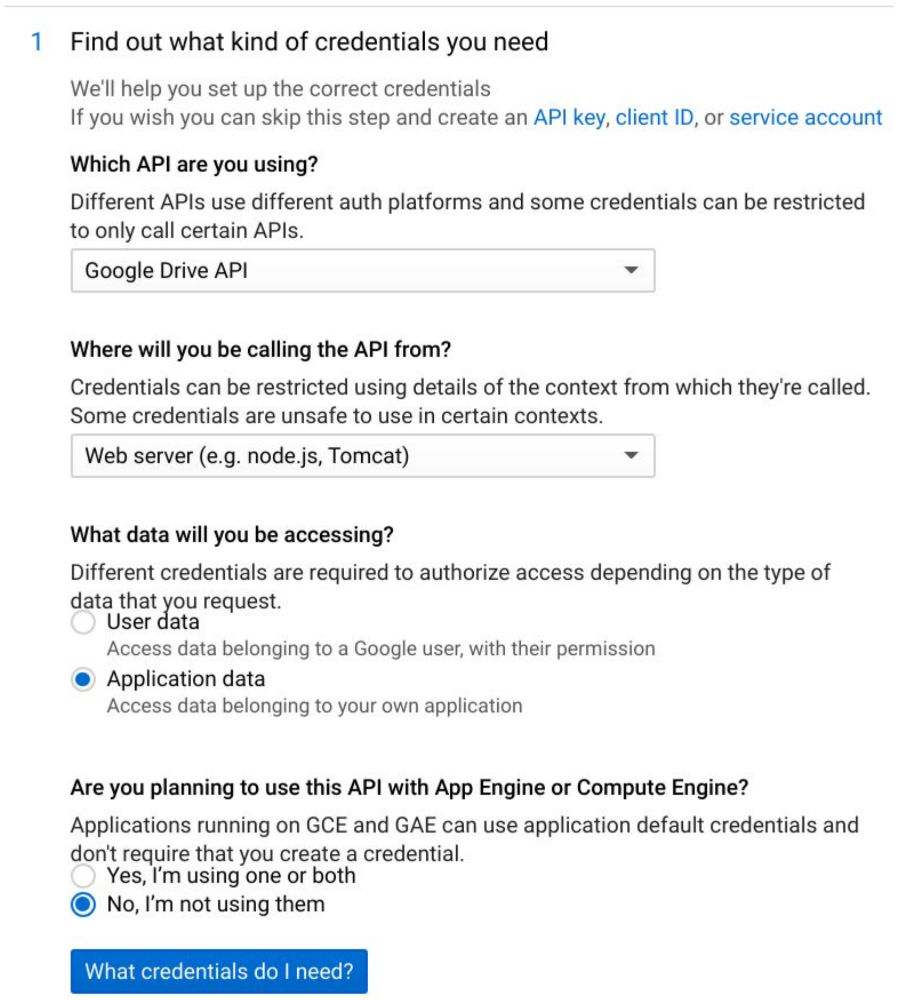


Your free trial is waiting: activate now to get $300 credit to explore Google Cloud products. Learn more

No data is available for the selected time frame.

Find the Google Drive API and the Google Sheets API in the API library, by using the search function.

For the Google Drive API, you will need to generate credentials however. When you “Enable” the Google Drive API, you will see the option to “Create Credentials”. Press it, and you should be navigated to a form. Fill out the form as shown below.


1 Find out what kind of credentials you need

We'll help you set up the correct credentials
If you wish you can skip this step and create an API key, client ID, or service account

Which API are you using?

Different APIs use different auth platforms and some credentials can be restricted to only call certain APIs.

Google Drive API

Where will you be calling the API from?

Credentials can be restricted using details of the context from which they're called. Some credentials are unsafe to use in certain contexts.

Web server (e.g. node.js, Tomcat)

What data will you be accessing?

Different credentials are required to authorize access depending on the type of data that you request.

User data
Access data belonging to a Google user, with their permission
[✓ Application data]
Access data belonging to your own application

Are you planning to use this API with App Engine or Compute Engine?

Applications running on GCE and GAE can use application default credentials and don't require that you create a credential.

Yes, I'm using one or both
[✓ No, I'm not using them]

Click on the “What credentials do I need?”, and you should be navigated to another form. The Service Account Name and the Service Account ID can be whatever you want, but the Key Type must be JSON. Make sure to also set Role to Project > Editor.

After you have done this, a JSON file will download on your computer. DO NOT POST THIS JSON FILE ONLINE. With this JSON file, anyone can have access to all the Inventory sheets.

Open the JSON file, and scroll until you see the field “client email”. You will need to share the spreadsheet with this email, and give it edit permissions.

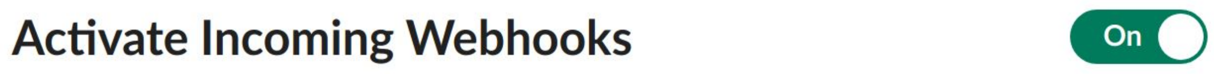
Configuring Slack

Next we are going to configure Slack Notifications. Log into the desired Slack Workspace, and then create a new Slack Channel. This Slack channel will have the sole purpose of receiving notifications about changes in Inventory.

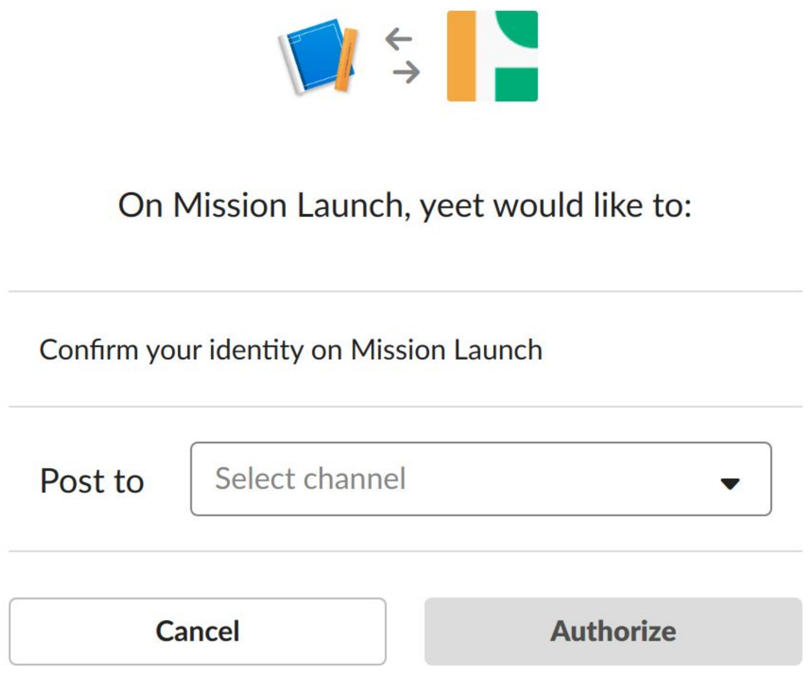
Then navigate to the website: https://[YourSlackWorkspace].slack.com/apps/manage, where [YourSlackWorkspace] should be the name of your Workspace. Once on this website, press “Build”, then press “Your Apps”.

Once there, you should press “Create a New App”. Make the name whatever you want, but make sure that the Workspace is the one you want the notifications to go to.

In the Left sidebar, click “Incoming Webhooks”, and then click the switch next to incoming webhooks. It should look like this:



Scroll down, and press “Add New Webhook to this Workplace”. The screen will then change to this:


On Mission Launch, yeet would like to:

Confirm your identity on Mission Launch

Post to: Select channel

Select the new channel you made earlier. Press Authorize to complete the Slack Notification Webhook setup. Now, if you navigate back to incoming webhooks. You will see a single Webhook URL. We will use this later, when configuring the script.

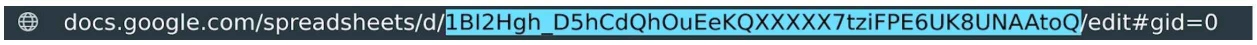
Configuring the Script

First, you will need to clone the github project by running this command on Terminal or Command Prompt: (You will need git installed)

```
git clone https://github.com/abugler/RockeTrackr.git
```

This will create a new folder, with all the python files needed inside. Drag the JSON file you download previously into this folder, and rename the JSON file creds.json.

You will then need to open the python file “main.py” with a text editor. On line 22 of main.py, you will find a variable named key. Change this variable to the key of the spreadsheet you copied earlier. You can find the key from the URL of the spreadsheet. The highlighted portion of the following URL is the key.



docs.google.com/spreadsheets/d/1BI2Hgh_D5hCdQhOuEeKQXXXXX7tziFPE6UK8UNAAtoQ/edit#gid=0

You will then need to open the python file “slack_notif.py”. On line 5 of slack_notif.py, you will find a variable named webhook_url. Change the value of this variable to the webhook URL you obtained from setting up the Slack App.

And the setup is complete! To run the script, navigate to the folder containing the scripts using Terminal or Command Prompt, and run the following command:

```
python main.py
```

The script should be running at all times, to check if any of the forms have been filled