

Box 5 - Bookkeeping Module

Connect Bookkeeping File Template to
DIY1 Script Library
(an unbound Google Script Project)

Table of Contents

Table of Contents	2
PART 2 - Detailed Instructions	3
Open your working copy of the External/Unbound Google Script Project	3
Create / Save a "Version" of the Llibrary for Template Reference	4
Copy the "Script ID" to your clipboard for paste into the file template for Reference	5
Open the original copy of the software and Open the Internal/Bound Google Script Project	6
Paste the Script ID into the Internal/Bound Google Script Project Interface	7
Configure the Library in the Internal Google Script Project Interface	9
Authenticate the Script (intro only)	10
PART 2 - Detailed Generic Instructions for making file copies	11

sdsd

PART 1 - Summary Instructions

The entire math and accounting portion of the software operates withOUT the External Library.

Thus you can download a copy of the file and use it without needing the External Library.

The External Library only controls User Interface features.

One key feature you should want to use is the Nav1 dynamic navigation system (think “dynamic list of tabs” aka “a dynamic table of contents” that can be updated with the click of a button (it can’t update manually in this system).

The External Library requires a process to connect it to the core file.

The first time you use an External Library feature, the file will ask for Authorization to access the separate script library.

This process looks scary. It looks like you are giving up your rights. In fact you are only giving up your rights in one file to another file on your own GDrive, so in that sense it’s no big deal (although it looks scary).

NOTE: it’s only no big deal because the file you are giving access to has no functions in it that share data beyond your own GDrive. You would need to check for that type of thing with any publisher who creates similar setups.

Open your working copy of the External/Unbound Library

Select File > Manage Versions

In the textbox, typep “Version1” or comparable

Save

Close the popup window

Select File > Project Properties

Highlight the Script ID

Ctrl-C to copy it

You are done with this file. Either leave it open in case you lose your clipboard item or close it.

In your Spreadsheet File Template

Select Tools > Script Editor

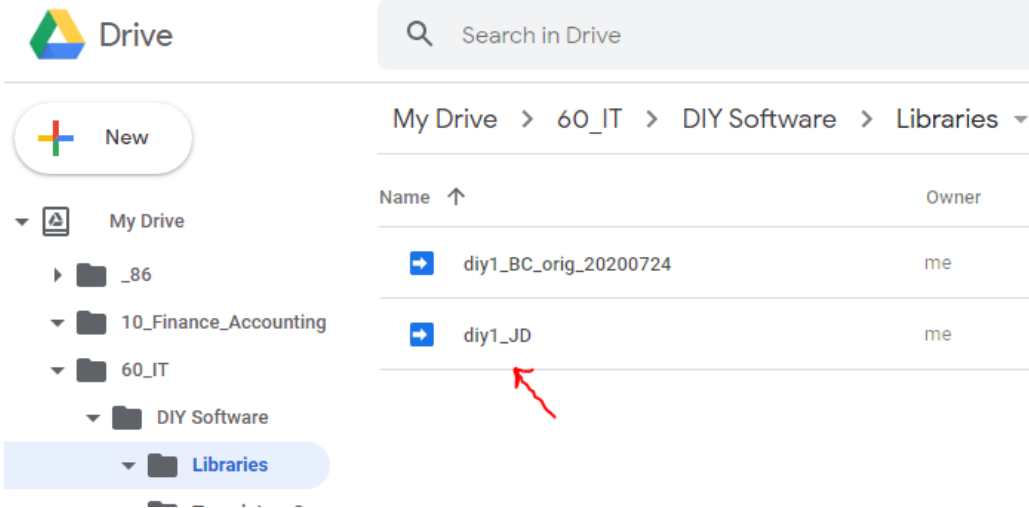
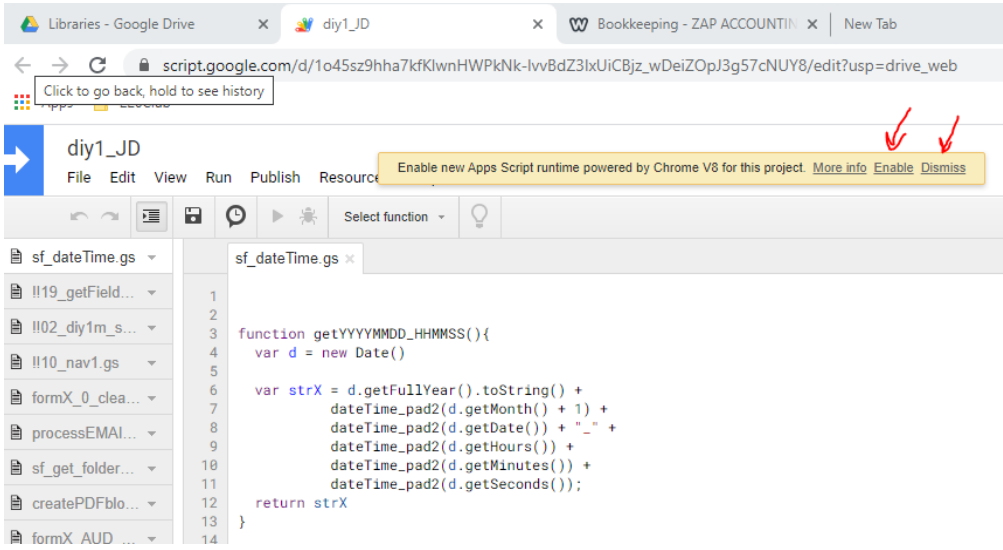
(the internal/bound google script file opens in a new window)

1. Either enable or dismiss the Apps Script Run time. Irrelevant and can be toggled on the run menu.
2. Rename the file to match the file template
3. Click on Resources > Libraries

4. Delete the existing reference to a library by clicking the X on the far right.
5. In the text box, paste (ctrl-v) the Script ID and click Add
6. You will need to make 3 changes to the configuration and save the changes.
 1. Version (1) - Click the drop down and select the Version of the library you are going to reference.
 2. Identifier (2) - Change the identifier to just "diy1" (so remove any trailer characters)
 3. Development Mode (3) - Change to "on" originally.
 4. Save

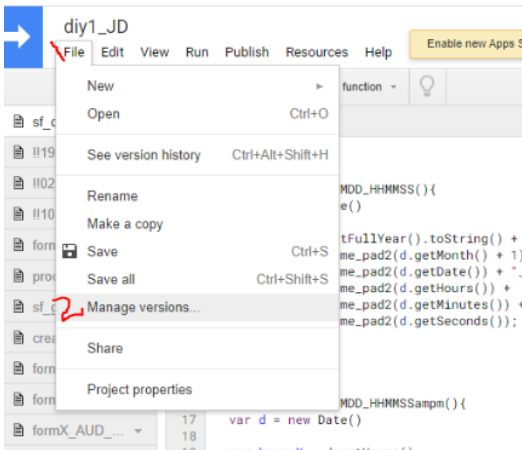
PART 2 - Detailed Instructions

Open your working copy of the External/Unbound Google Script Project

	In drive.google.com click on the version you named diy1_{your initials}.
	In a new tab, the following will appear. If the question comes up about enabling new Apps Script runtime either enable or dismiss it has no affect. You can change your decision later with an option on the “Run” Menu.

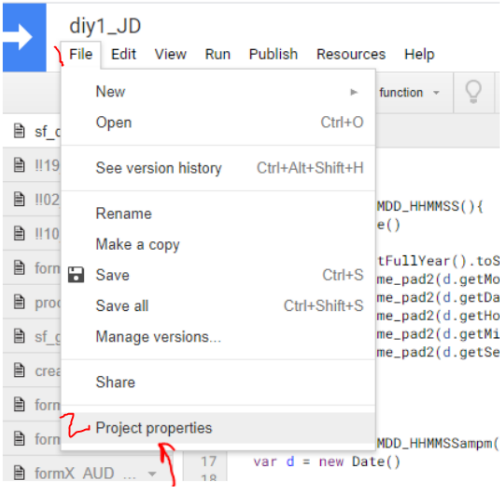
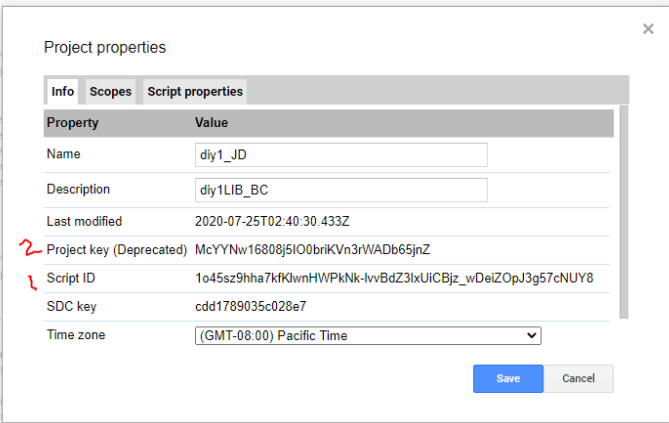
Create / Save a “Version” of the Library for Template Reference

Versioning is like making “invisible” copies of a file / code library. If you have everything working fine in a software application with an external code library attached, do you want that to break when you start futzing with code in the external code library? In a production environment you do not. In a personal environment you might or might not. Versioning allows you to reference a copy (a snapshot) of the library at a point in time. This helps the developers and users manage what gets referenced as the code library changes with time. Google allows for static referencing. They also allow for updates to that copy to be ignored or engaged with. First we just need to create a version. (save a snapshot of the library for reference).

	Select File (1) > Manage Versions (2)								
Manage versions A version is a snapshot of your project that can be used as a library in other projects. Describe what has changed... Save new version 	In the text box, remove any text that is in there and Enter “Version 1” . This is an irrelevant descriptor at this point. You will see where it shows up in the next image.								
Manage versions A version is a snapshot of your project that can be used as a library in other projects. Describe what has changed... Save new version <table><thead><tr><th>Version</th><th>Description</th><th>Saved by</th><th>Date</th></tr></thead><tbody><tr><td>1</td><td>Version 1</td><td>rammick...@gmail.com</td><td>2020-07-24 22:05 </td></tr></tbody></table>	Version	Description	Saved by	Date	1	Version 1	rammick...@gmail.com	2020-07-24 22:05 	The window will refresh and the version will show up
Version	Description	Saved by	Date						
1	Version 1	rammick...@gmail.com	2020-07-24 22:05 						
(No Photo)	Close the popup window now. -- Click the X in the upper right of the popu up window or the close button at the bottom of the window.								

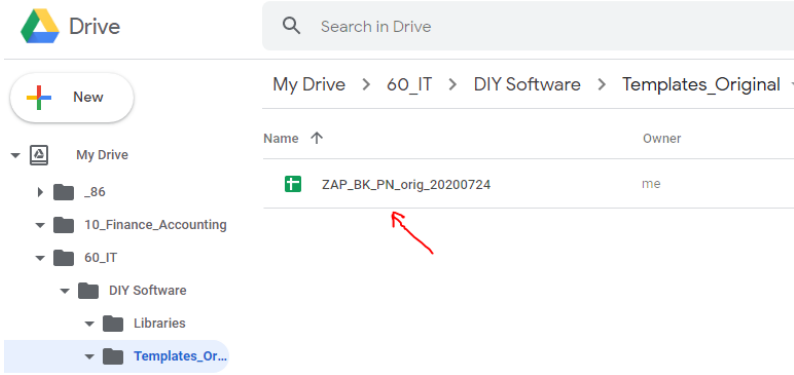
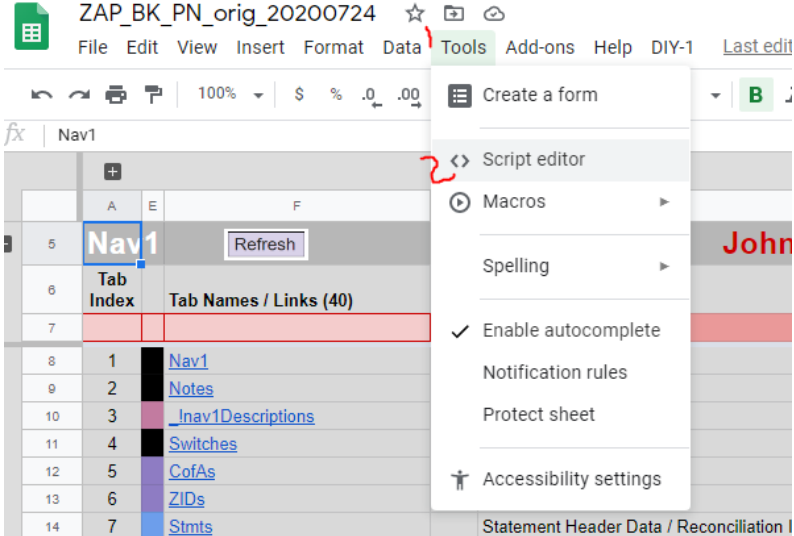
Copy the “Script ID” to your clipboard for paste into the file template for Reference

The file template needs to know what script to access and what version of that script to access, and if it should only use the static version or if it should access any changes to that version since it was created. This Script ID is the major connector that connects the two files. The other settings are configured in the Template itself.

	Go to File (1) > Project Properties (2)
	1) Copy the Script ID With your mouse, highlight the script ID characters and use ctrl-C to copy that text to your clipboard 2) HISTORICAL NOTE: This process used to use the Project Key Field (one line above the Script ID). The location we’ll paste into might still say Project key field. That ID has been deprecated (meaning they are trying to phase it out)
(see image above)	Close the window -- Hit X in upper corner or save or cancel.

You will be leaving this browser tab now. Just leave it open and go to another tab...

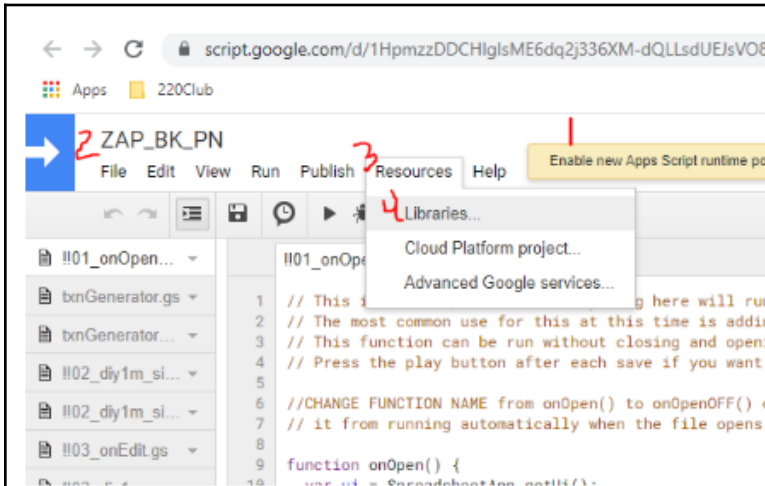
Open the original copy of the software and Open the Internal/Bound Google Script Project

	In drive.google.com click the ORIGINAL copy you made of the file template.
	It will appear in a new browser tab. Click on Tools (1) > Script Editor (2) A new tab will open with another Google Script Project Interface

Paste the Script ID into the Internal/Bound Google Script Project Interface

Each google sheet has a built in, 'internal' google script project. This project interface is identical to the external project you've seen prior. The only difference is that this one is "bound" to this one file only and it can not be located separate from the file template in the file system.

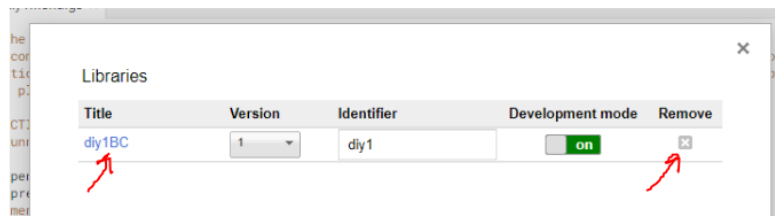
When you clicked on Tools > Script Editor a new browser tab opened with another Google Script Project Interface.



(1) Either enable or dismiss the Apps Script Run time

(2) The Name of the file is odd. Ideally, it would match the name of the template you have open. It is intrinsic to that file and can not be found separate from it. Not updating this name when the file name is changed is a "bug" in my opinion. You really should change the name to match the file name. IF not on this copy, because it's the original copy, do it for sure on your working copies you make later. You can click on the file name and change it here if you'd like

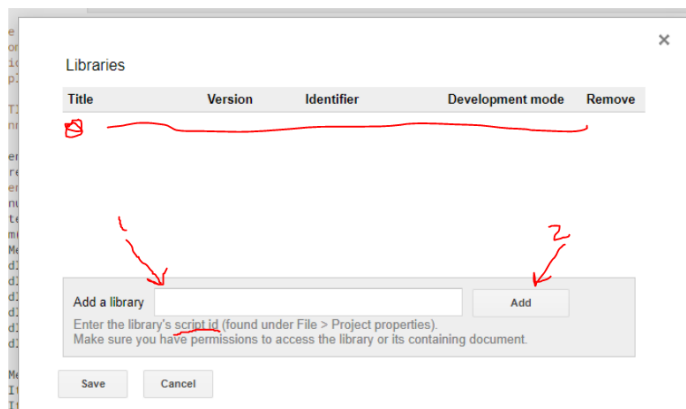
Click on Resources (3) > Libraries (4)



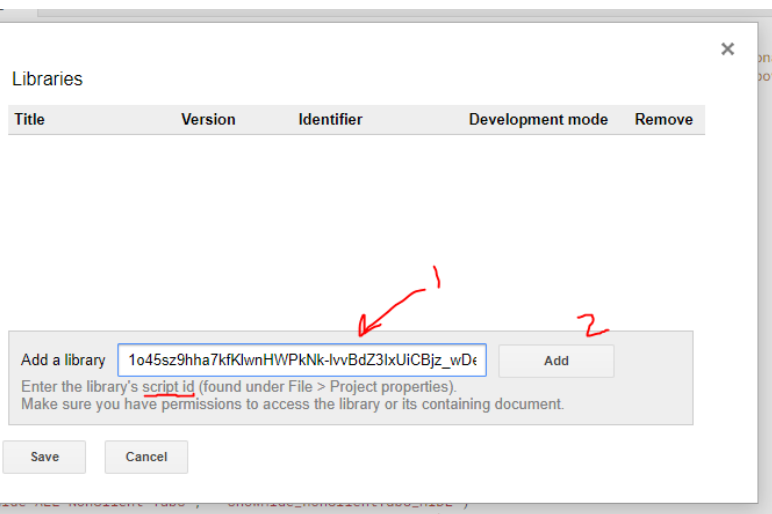
A popup window will appear.

Delete the existing reference to a library by clicking the X on the far right.

That is a reference to the file that was on the gDrive you copied from. Since these are now on your own gDrive this file doesn't have access to that library anymore.

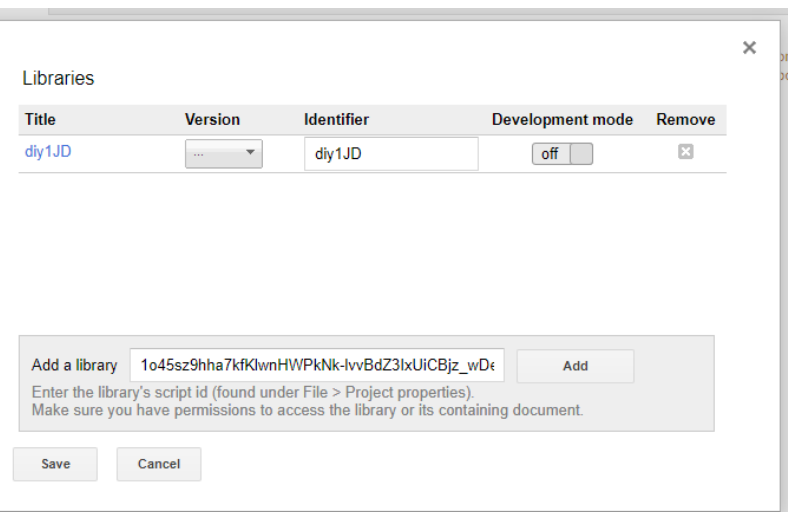


The prior library reference has been deleted. (0)



In the text box (1), paste (ctrl-v) the Script ID and click Add (2).

If you lost the Script ID from your clipboard due to other actions, go back to the External Google Script Project you left open in another tab and copy it again. Then return here and paste it.



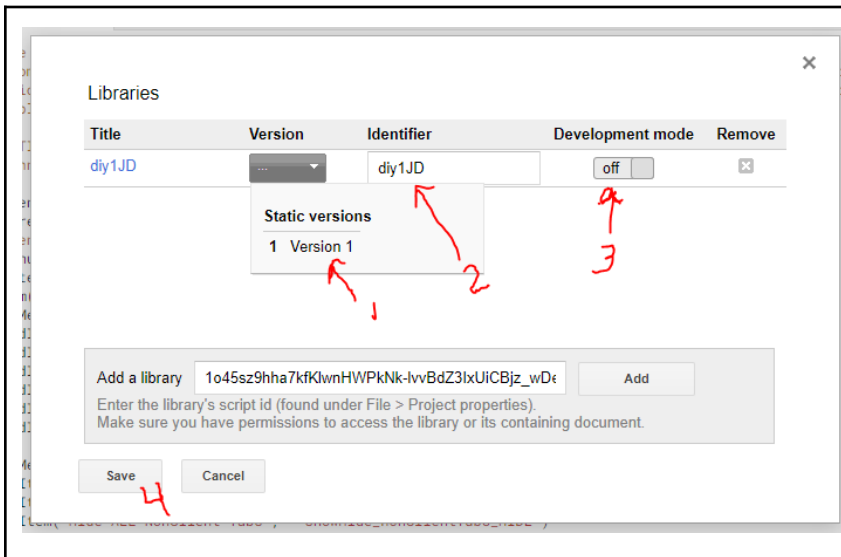
When you do the Library Reference will be added where the other was.

You can see it now has your file name in the title less any underscores.

Oddly enough it left the script id in the text box below. That appears to be a "bug" in google's add record process.

You need to make changes to this now. Don't close this window...

Configure the Library in the Internal Google Script Project Interface



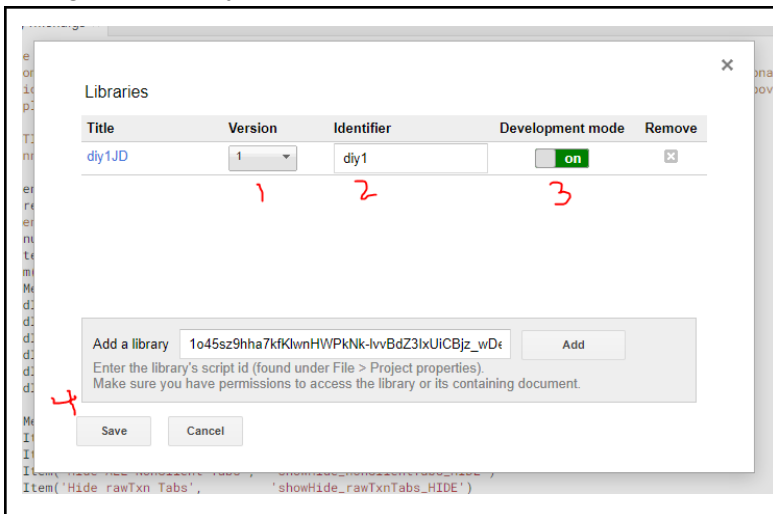
You will need to make 3 changes to the configuration and save the changes.

1. Version (1) - Click the drop down and select the Version of the library you are going to reference.
2. Identifier (2) - Change the identifier to just “diy1” (so remove any trailer characters)
3. Development Mode (3) - Change to “on” originally.

Version - This is why you had to create the version in the external file.

Identifier - When calling a routine from a library like “getDateMMYYDDDD”, you put a short extension (identifier) in front of the routine name to specify the library. In this case you are telling this file that when they see “diy1.getDateMMYYDDDD”, it can use diy1 to track back to the longer script ID to track back to the routine called.

Development Mode - When in development mode, it will not use the static version of the library. It will use the most recent version in real time showing all developer changes. This only works procedurally in this case because this is your own library and your own file. If you make changes to your library we don't want to have to re-register it every time.

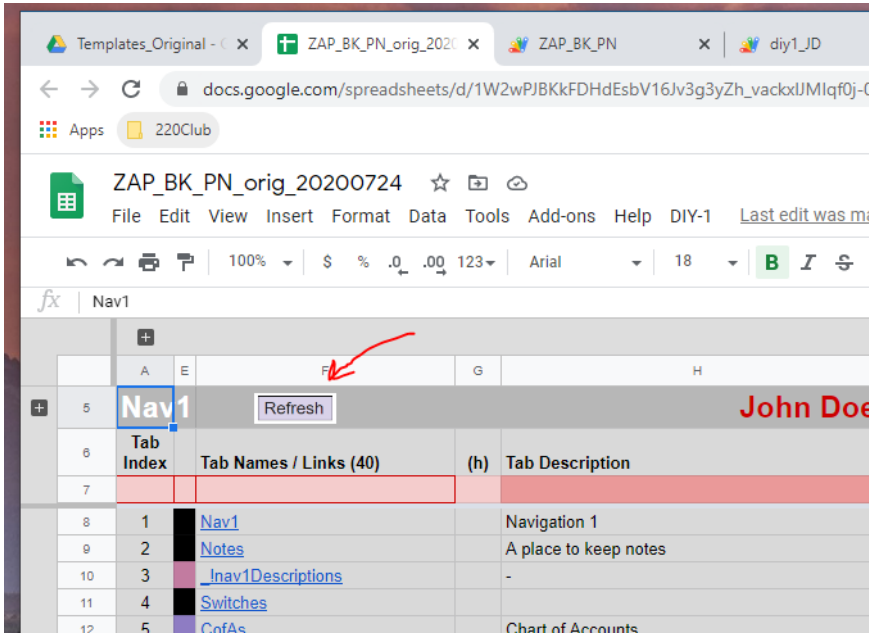


Click “save” (4)

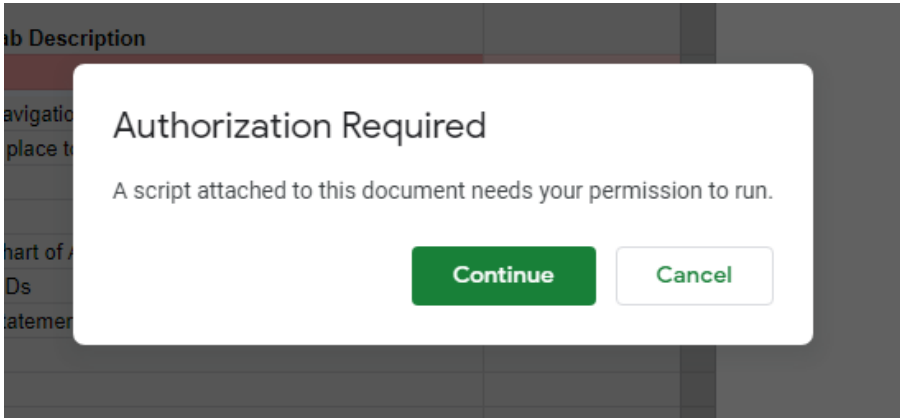
Authenticate the External/Unbound Google Script Project (intro only)

Go back to the Original File Template

“Do something” that calls any script in the External Google Script Project. There is no way to know this intuitively. You would need to be told what to do to trigger this or look in the External Google Script Project and identify a way to call a function.



In this case you are going to click on the “Refresh” button on the “Nav1” worksheet, which is the first worksheet in the work book.



Understand that the authentication process is in fact designed to protect you. It does not discern as much as it should who created and who owns the file now. It's confusing and can be scary unless you've built your own from scratch and see the same illogical process.

Select “Cancel”

See next document for that.

