

Trellinator Hello World

 manual.docs.theprocedurepeople.com/index.php

1. This is a pretty complicated process, so I would recommend watching the [Video Walkthrough](#) first
2. You will need [npm](#) and [clasp](#) installed
3. Download the [Trellinator template from github](#) as a ZIP file then unzip it
4. Make a copy of [this file](#) from your own Google account and add it to your own Drive
5. Login to a Trello account, either your primary account or one you create for testing purposes, in the same web browser session where you have your Google account open
6. In the Configuration tab of the spreadsheet you copied go to the link in cell C2 then copy your API key back to cell B2 and hit enter
7. Now go to the link in cell C3, allow access then copy the token back to cell B3 and hit enter
8. Now in a terminal (linux/unix/macOS) go to the directory where you unzipped the Trellinator template repo you downloaded from github
9. Run the command

```
mv trellinator-template-master hello-world
```

10. Now run

```
cd hello-world && ./init
```

to initialise the project (you will be asked for your sudo password in order to install the trellinator script in /usr/local/bin/)

11. If you haven't already installed them globally, install the `murphytest` and `md5` npm packages by running

```
npm i md5 murphytest -g
```

12. From your spreadsheet go to Tools -> Script editor and then choose File -> Project properties
13. You will be asked to name the file, so call it something like "Hello World Trellinator by The Procedure People"
14. Once the project properties dialog displays, copy the Script ID which will look something like this: `17Qfpk-1SIAynmG_OabWLBBYHzuKOstT646Nn6M2DCI8vPfbfcGyohAYT`
15. Choose the correct timezone for your script and then click Save
16. Now from your git working directory run

```
clasp clone SCRIPTID
```

where SCRIPTID is the ID you copied, for example

```
clasp clone 17Qfpk-1SIAynmG_OabWLBBYHzuKOstT646Nn6M2DCI8vPfbfcGyohAYT
```

17. Delete the file "Code.js" and leave the file appsscript.json

18. Now run

```
trellinator push
```

in order to push all the code to your Google Apps Script

19. From the Script Editor refresh the page and you should see all the code appear

20. Choose Publish -> Deploy as Web App

21. Leave the app set to Execute as "me" and change "Who has access to the app" to "Anyone, even anonymous" then click "Deploy"

22. Now flip back to your spreadsheet and reload the page, and you should see the "Trellinator Utilities" menu appear

23. Choose "Initialize" from the menu and wait for the script to finish, one spreadsheet tab should be created for each board you're a member of as well as some other configuration boards

24. We will now create a new Hello World function and associated Murphy test to be executed on one of the boards you're a member of

25. Pick the board that is "left most" in the tabs created when you initialised. For the sake of this exercise we'll assume this board is called "Iain Dooley Sandbox"

26. Back in your working directory you'll see that when we initialised the project, a directory called "boards" was created so now run

```
cd boards && mkdir iain_dooley_sandbox && cd iain_dooley_sandbox
```

replace iain_dooley_sandbox with whatever your board name is (for example make My Board into my_board)

27. Now run

```
trellinator new HelloWorld && cd HelloWorld.js.murphy
```

28. In Trello, go to the board you chose (eg. Iain Dooley Sandbox) and create a list called "Sample"

29. Go back to your spreadsheet and click into the tab for the board you chose (should be the "left most" board tab, eg. "Iain Dooley Sandbox")

30. Now into the "Function Call" column in the first row put "sampleNotification"

31. In your spreadsheet go to the "Info Log" tab and clear the rows that you can see in there, to make it easier to see new notifications

32. Go back to Trello and create a card called "New Card" in the "Sample" list

33. Flip back to the spreadsheet and you should see a new log message appear for the createCard notification caused by your having created a card

34. Copy the contents of the cell, then go to <http://jsonviewer.stack.hu/> or another online JSON viewer, paste in the contents of the cell you copied and click "Format" up the top

35. The log format is an array of messages, and you should see that below the log entry "executing realtime function: sampleNotification" that there is an entry that contains a JSON object, it will start with something like "{ \"model\": { \"id\": \"

36. Copy this entire JSON object and then, back in your terminal open the file

```
HelloWorld.js.murphy/notifications/default.run.js/card_created.js
```

then paste in the JSON object you copied and wrap it in the function `JSON.parse()`;
then assign this to a variable called `card_created`

37. When done, your file should look something like this

38. Save the file and then open the file `HelloWorld.js.murphy/default.run.js` and make the following changes to the template content:

1. Edit line 23 to change `SystemUnderTest.js` to `HelloWorld.js`
2. Edit line 24 to change `some_event.js` to `card_created.js`
3. Edit line 27 to set the `Trellinator.username` variable to the username of the Trello account you connected your spreadsheet to
4. Edit line 39 to change your function to `helloWorld` and change `some_event` to `card_created`

39. When done, your file should look like this

40. Now copy the trello API key and token from your spreadsheet's configuration tab and run

```
node default.run.js APIKEY APITOKEN
```

replacing `APIKEY` with your actual `APIKEY` and `APITOKEN` with your actual API token

41. You should see an error trace that starts with

```
ReferenceError: helloWorld is not defined
```

42. Open up the `HelloWorld.js` file in the parent directory and add this content to it

43. Now run your test again and you should see no output, but when you look in the card you created in Trello you should see your comment appear

44. In your terminal, change back to the root of your working directory and run

```
git add .
```

45. Now run `git status` and you should see there are some new files. Copy the full path of the top file, that is the path to `HelloWorld.js`

46. Now open the file `claspignore-template` and add the line

```
!boards/iain_dooley_sandbox/HelloWorld.js
```

47. Now run

```
git add . && git commit -m "Added HelloWorld function and modified claspignore template"
```

48. Now run

```
trellinator push
```

49. Back in the Script Editor of the spreadsheet you're working in, refresh the page and you should see your new `boards/boardname/HelloWorld.gs` file

50. Choose `Publish -> Deploy as web app` again and update Project version to "New" then ensure that "Who has access to the app" is still set to "Anyone, even anonymous"

51. Now in the spreadsheet tab for the board you're working on (the "left most" board tab in your spreadsheet) replace the "sampleNotification" entry with "helloWorld"

52. Now, whenever you create a card in the Sample list, you will see a comment posted.
If you add another list and create cards in it, you will not see the comment posted
53. This Hello World process has shown you how to set up your project and create well tested Trellinator code. The following sections of our documentation will take you through how Notifications work and how to create functions using the API reference