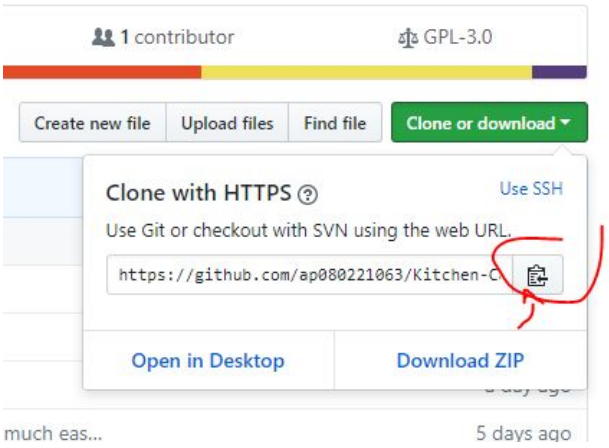
# RASPBERRY PI CONFIGURATION GUIDE

for KCA , Kitchen Cabinet Assistant

1. You will need a Raspberry Pi;
   *(the one used in this configuration is a Pi 3 Model B)*
2. Get a copy of a Raspbian distribution here, of which I highly recommend the LITE version (RASPBIAN STRETCH LITE);
   *(it has no graphical interface, command line only, making it lighter on the Pi)*
3. Flash an ssd card with the distribution;
   *(used this application)*
4. Power up the pi!;
5. You will need to connect your pi to a monitor just this once for an initial configuration;
6. Login with the default user **Pi** and password **Raspberry**;
7. Voilá, your in! Use the terminal to:
   a. type `raspi-config`
   b. when the splash screen appears, change the pi's password;
   c. enable ssh;
   d. change the locale and keyboard configuration if you desire;
   e. give a hostname to your pi;
8. You can now disconnect your pi from your monitor, reboot, and store it in a place you can forget about it;
9. Access your pi using ssh with tools like Putty and WinSCP;
10. Install node.js
    a. `curl -sL https://deb.nodesource.com/setup_7.x | sudo -E bash -`
    b. `sudo apt-get install -y nodejs`
11. Install git
    a. `sudo apt-get update`
    b. `sudo apt-get install git`
12. Clone project from Github
    a. `git clone to the project's destination folder (one of the below):`
       i. `cd to project destination folder`
       ii. `cd to a directory and then`
           1. `sudo mkdir 'Folder_Name'`

b. Go to KCA's project on github (here) and copy the link as shown in the image below:




i. type:
   1. sudo git clone 'project's path copied'

c. type:
   i. cd to sub folders server/api-node
   ii. npm update

13. Install Angular CLI
    a. cd back and to Kitchen-Cabinet-Assistant/KCAssistant
    b. npm install -g @angular/cli
    c. npm update
    d. ng build
14. Put KCA's back-end running
    a. cd your way to Kitchen-Cabinet-Assistant/Server/api-node
    b. type:
       i. node app.js (your backend web api is now running and listening to any type of GET and POST requests)
15. Put KCA's front-end running
    NGinx is a light-weight web server and it will allow to serve web pages.
    a. type:
       i. sudo apt-get install nginx
       ii. sudo /etc/init.d/nginx start (starts the server)
16. Copy files to webserver
    a. In step 13, after ng build, a dist folder was created and files created within it.
    b. Copy all files within that folder to NGinx var/www/html/ folder with the following:
       i. cp -r path to project/dist/. var/www/html/
       ii. On another device, go to a browser and type the raspberrypi's hostname or ip: http://192.168.1.66

This isn't 100%, however the important pointers are here.
Good luck and enjoy.