**BotPress Readme:**

This directory contains the data directory of a development BotPress installation. There is also an export file, bot\_211sonoma\_1607298261301.tgz

The chat bot, 211chatbot, accesses an SQL database found in the web\_scrape directory in this repository. (Note, the words in “key\_words” have been converted to all lower case to make matching easier.)

**Update 12/2/20: Please use the new sql files in the web\_scrape directory.**

Key parts of the repository:

1. The main flow of the bot is in bots/211sonoma/flows/main.flow.json
2. The javascript “action” that calls the database to get the links is: global/actions/callResourceApi.js (**Note**: this has been updated to call a webservice on another server, not your local machine.)
3. There is a directory (assets/modules/qna) which sounds like it should have the “Q&A” from the studio, but I haven’t found where they’re stored yet…

How to install the bot and get it working;

1. Install botpress here: <https://botpress.com/download>  
   **Note**: Steps 2, 3 & 4 below are not needed with the current javascript file as it references these files on another server, not your machine.
2. There needs to be a database on a server. The SQL files are in the web\_scrape directory. Create a database, I used the name “211Sonoma”. Import these 4 SQL table files into that database: key\_words.sql, Links.sql, other\_words.sql, WordLinks.sql. You probably want to also create a database user on that database.
3. In the server directory copy the 2 php files from the php directory. I used Xampp as a local server on my machine, so the files went into the htdocs directory. Start the server if localhost.
4. You will need to create the file dbconvars.php in this directory with the following information:   
   *<?php $dbhost = "127.0.0.1"; $dbuser = "username"; $dbpwd = "yourpassword"; $dbname = "211Sonoma"; ?>*
5. Start BotPress (./bp in it’s installed directory if using Linux).
6. Open localhost:3000 to start the bot.
7. On the main page, click “Create Bot” and “Import Existing” to import the bot.
8. If you’re not using the localhost database with the same name as I did, then you’ll need to update the following line in the callResourceApi.js file. You can find this in the “Code Editor” tab.  
   *const { data } = await axios.get('http://localhost/request.php?command=getResource&resource=' + rsrc)*
9. Select “211sonoma” as the bot and it should open the BotPress studio. Launch the emulator and you should have the functioning bot.

The bot is now on Heroku at https://young-oasis-43466.herokuapp.com/

You can access this bot from a web interface at [http://cfsj211chat.tk](http://cfsj211chat.tk/) The index file used is in this directory – index.html.

**Heroku installation instructions:**

The botpress instructions are here: [https://botpress.com/docs/advanced/hosting#heroku](https://botpress.com/docs/advanced/hosting" \l "heroku)

First you need a Heroku account and the Heroku CLI, see here: Install the Heroku CLI by following [these instructions](https://devcenter.heroku.com/articles/heroku-cli).

Note that the version of botpress used to create your ‘data’ directory must match the version installed in Heroku. This is version v12\_14\_1 in the Dockerfile in this directory.

You’ll need Docker running on your computer. Installation instructions for your OS can be found here: https://docs.docker.com/engine/install/

As you see in this directory, you need the following:

* A Docker file
* The data directory from the Botpress instance.

The steps for creating your Heroku Botpress instance the first time are:

1. heroku login
2. heroku create ## Note, save the instance name!
3. heroku container:login
4. heroku container:push web --(your instance name)
5. heroku container:release web --(your instance name)

After you’ve successfully pushed the application to Heroku, login to your instance:

https://yourinstancename).herokuapp.com/admin/login/default

The first time you login you can create your admin account / password. This is where you can modify your bot.

If you need to push your bot again, these are the commands:

1. heroku login
2. heroku container:login
3. heroku container:push web --(your instance name)
4. heroku container:release web --(your instance name)