Documentation.js Tutorial

Alrighty Alrighty here goes the tutorial on installing this thing and also getting it working with travis so its all automated

Step 1.

https://github.com/documentationjs/documentation/blob/master/docs/GETTING_STARTED.md

follow this guide but instead of doing

```
npm install -g documentation
```

do

```
npm install documentation
```

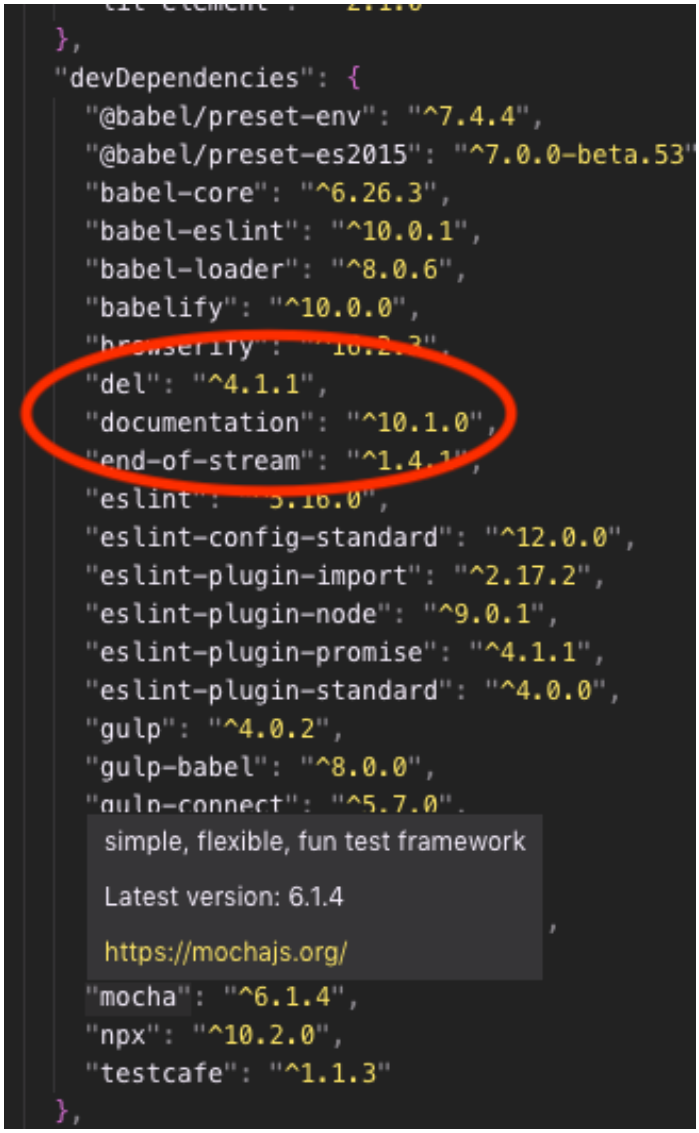
unless of course you would like it to be installed globally

Step2.

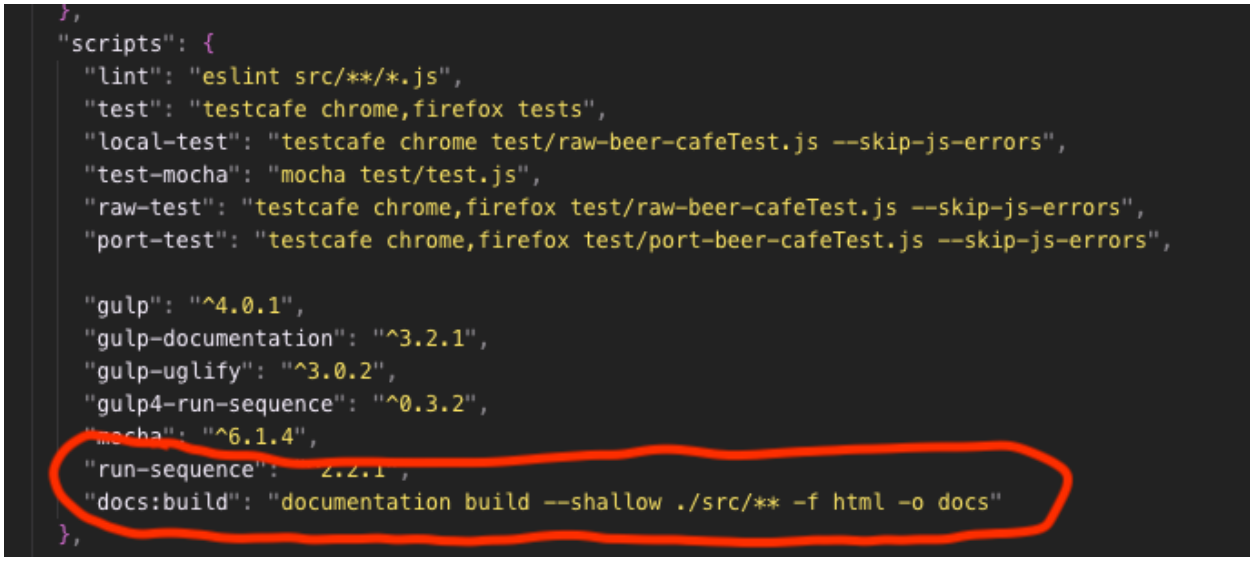
Alrighty alrighty cool you got it installed now its time to add it to the project so that you can do something like

```
npm run blah blah
```

first things first open up your package.json file and then make sure under devDependencies you got something there that says "documentation" followed by a version number



then add the following under scripts "docs:build": "documentation build ./src/** -f html -o docs”



alrighty now head over to your terminal and perform

```
npm run docs:build
```

a docs folder should be crated and you should be able to load html file in there with your documentation, but lets break down the above command so that you know you didn't just put some nonsense into this package.json file

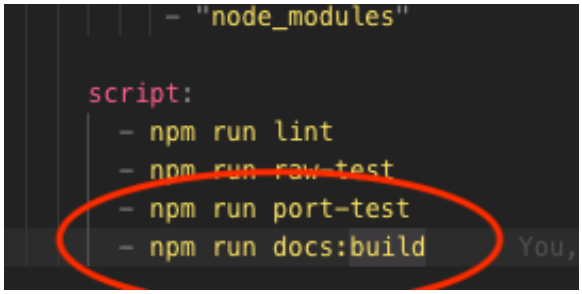
./src/** is the file path to the files you want to be documented

-o docs this is the folder that will hold the output when documentation.js finishes doing its magical stuff

-f html is used so that the output comes out as an html documentation.js supports other outputs as well head over to the above link to checkout the rest

Step3.

Alrighty so you got yourself some documentation but you want to automate this creation with travis CI, easy enough open up that god forsaken .travis.yml file and head on over to where it says script: right under it add the command you created previously in your package.json here we used npm run docs:build so we simply add - npm run docs:build under scripts and we are done



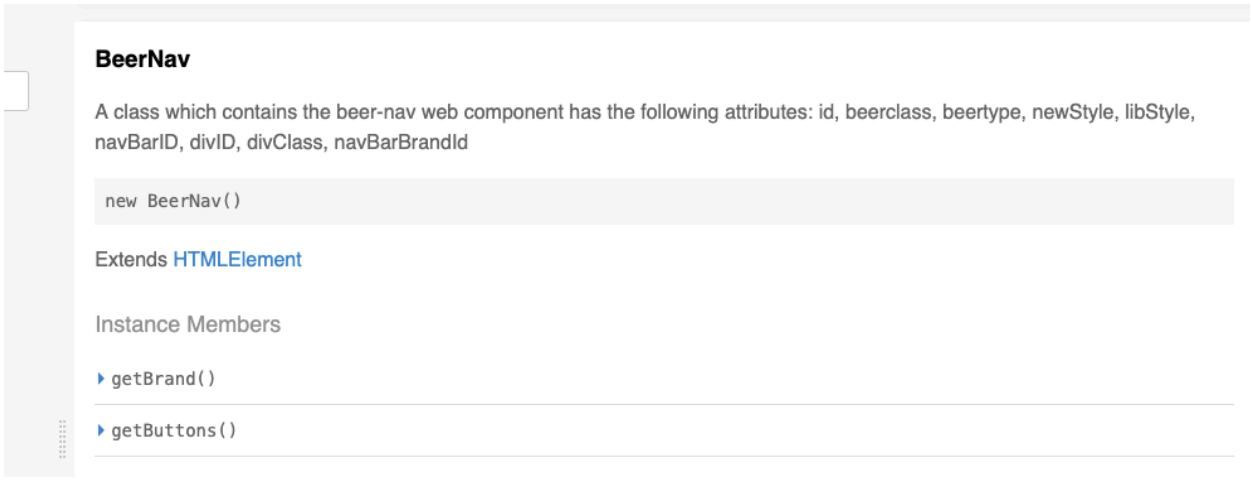
Using Documentation.js

alright alrighty this is quick and easy first things first you got to use /** / *this first* /** is how documentation.js parses the file anything you put in this block will pretty much appear you can even use HTML but be warned at least from my experience if you try and use HTML in combination with their @tags the @tags will take priority and your HTML stuff will vanish

Here's an example of what a header should look like using the @ tags

```
/**
* @description A class which contains the beer-nav web component has the
following attributes: id,
* beerclass, beertype, newStyle, libStyle, navBarID, divID, divClass,
navBarBrandId
* <beer-navbar> </beer-navbar>
*/
```

And here's what documentation.js will create



EXTRA EXTRA CREDIT NOTES!!!!!

So we have a function which makes use of that really cool dynamic importing stuff and every documentation generator and its grand pappy was crying about it, it appears that these things also take a look at the code and sort of following the imports in the same way a web crawler follows links at least this is my theory and I very well could be extremely wrong, so if you were paying close attention to when I added the command in the package.json you would of seen a - -shallow this basally tells it yo your not a web crawler stop crawling and gets around the dynamic import issue.

HAVE FUN!!!