Using TestCafe

Installing TestCafe

I basically just followed this guide

- https://devexpress.github.io/testcafe/documentation/getting-started/#creating-a-test

Fixture

```
fixture`Getting Started`
.page`http://devexpress.github.io/testcafe/example`
```

JavaScript

- Im not really sure what a fixutre it it appears to be some thing that allows things to be organized into catagories.
- .page
  - This part is just the webpage that you want to test

My first wee baby test

```
.typeText('#developer-name', 'John Smith')
```

JavaScript

whats happening here

- so from what i can tell here it is searching for an id called developer-name and then entering John Smith into it
    - we shall test this theory by attempting to populate the comment box
        - couldn't populate the comment box because it is disabled
- I don't know how it knows that it can enter John smith here I expect its something fancy on their end thats checking through a bunch of elements or something

```
test('My first wee baby test', async t => {
```

JavaScript

- what I've discovered about this line
    - test must be test if you change it to test2 or anything else it will bug out
- Alrighty I FOUND OUT HOW TO MAKE MULTIPLE TESTS
    - easy peasy you simply perform

```
test('whatever you want', asynch t = > {
```

JavaScript

THE MIGHTY ASSERTIONS

```
.expect(Selector('#article-header').innerText).eql('Thank you, John
Smith!')
```

JavaScript

- So it appears that we use the Selector object to select html attributes and then we can investigate them to see if they are what we expect them to be here the assertion is looking for an ID called article-header and then grabbing its inner text which as well as i can tell is just some text thrown into the tag and then seeing if it is equal to Thank you, John Smith!
    - John Smith was a piece of information that was entered previous to this

Selectors

- '.name' Searches for a css element
- 'name' searches for html element
- '#name" searches for an id

Using Selectors & Shadow-DOM

- Use document.querySelector().shadowRoot to select shadowRoot
  - From there use querySelector again to select element you want to test

```
await Selector('core-hello'); const p = await Selector(() =>
document.querySelector('core-
hello').shadowRoot.querySelector('#hello'))
```

JavaScript

In the above code querySelector('core-hello') is grabbing the element that is part of the light-DOM (the part thats visible to everything and the .shadowRoot.querySelector('#hello') is looking for an element within the Shadow-DOM that has id hello

Useful Links

- https://devexpress.github.io/testcafe/documentation/test-api/test-code-structure.html#fixtures
- Links that were help full with solving Shadow-Dom issue
  - https://gist.github.com/AlexanderMoskovkin/897073929442db031d518e1e6db4ec9e
  - https://github.com/DevExpress/testcafe/issues/2172