Testing Notes

1. Django Tests

Django Tests is based on unittest. Shares the same syntax but extends the functionality into as below

1. Allows you to simulate a request to see what response is returned by a particular endpoint. Also allows you to check (a) status code of endpoint (b) context (c) what content is actually being returned

Note: When you are creating tests don’t need to be exhaustive. The idea is that when bugs are discovered, you will add an extra test to make sure that that bug will never turn up again

1. Selenium

Django Tests does not have the ability to test code that occurs on the client side. I.E. Any event handler that is attached to HTML elements cannot be checked whether that interactivity is correct.

Common way to conduct browser testing is via Selenium. Selenium can simulate a web browser and a user interacting with that browser via a web driver. A web driver allows a developer to programmatically control what goes on to a browser.

1. Selenium Set-Up

Each browser requires a special web driver to be installed. Please download these from the Selenium download page.

1. Selenium Commands

**File\_uri(uriName)**

Selenium needs a page’s URI in order to test it. Simple use the above method to get it.

**Driver.get(uri)**

This would tell the driver to actually go to that page

**driver.<htmlElement> and driver.page\_source**

Once you have the uri attached, then you can access any part of the DOM simply by just dot notation.

**Driver.find\_element\_by\_id(“increase)**

This works too to find HTML elements. You can assign this element to a python variable. If it is a button you can use increase.click().

1. CI/CD

Continuous Integration

* Frequent merges to main branch
  + Catch and resolve conflicts quickly
* Automated unit testing
  + Make sure each component – like function by function – works well.

Continuous Delivery

* Short release schedule
  + Incremental changes means faster time to market plus catching bugs earlier

1. Github Actions

Allows for certain actions to automatically trigger once something has been merged to a repo. I.E. A style check for code or running unit tests and having an email sent if they fail

Github actions is written in YAML. YAMlL has the following:

1. Key-value pairs
2. Arrays
3. Nesting between key and values i.e. within 1 key you can have 1 array which was values
4. Docker

Docker helps to standardize environments across servers and development machines to ensure that application actually works.

Docker creates a container where each container contains its own configurations. Applications are ran inside these containers instead on separate developer machines.

Docker is similar to a VM however a VM runs an entirely new OS while docker containers run on Host OS but has a layer in between applications and OS. Hence, they are lighter weight than VMs

Docker files describe instructions to create a docker image. Docker image contains all the configurations you want the container to have.

A common problem docker fixes is the issue of having a separate database. Databases in production are hosted on their own server such as in POSTGRSQL or MYSQL. However, your development database might be SQLite. Docker fixes this by having separate containers for your web application and your database.

Docker compose allows us to compose various service. I.E. allow 2 containers one running Postgres another running the web application to talk to one another