React Integration / Team Notes

Flask

Demo code

- https://github.com/instructorlee/react_integration_flask_app

Create base project

- dB
- Github

Integration flow

- config
- server.py
    - CORS (**need to install flask_cors**)
    - app.secret_key
    - +/- 7 lines related to loading the configuration.
- decorators
- controllers
    - user_congroller.py
- models
    - base_bodels.py
    - user_model.py

Spring Boot

Create your project base

- pom
- schema
- github

Implement Spring Security

- Work through Authorization
    - https://login.codingdojo.com/m/325/9951/67401

Implement JWT Code

- Read through the tutorial below. As you read through it, and understand, move sample code into your project.
    - https://www.bezkoder.com/spring-boot-jwt-authentication/
- Demo code
    - https://github.com/instructorlee/react_integration_spring_boot
- Notes:
    - The User class has some changes from the Spring Security updates.
    - A lot of what needs to be done just involves C&P'ing code from the sample. This gives you more time to read and understand.
    - You will be moving code in bits and pieces. So you will nearly constantly see error messages. Ignore them, unless you just happen to know the immediate issue. As you build out more code, the error messages should disappear.
    - Take time to read the tutorial and gain some understanding of what each file does. No need to deep-dive into new code. Getting a basic understanding will be enough to help when you are debugging.
    - My code has been modified from the tutorial. So it should fit pretty easily with little adjustment needed.

Postman

Download: https://www.postman.com/

React

Demo code

- https://github.com/instructorlee/react_ingegration_react_app

Create your base app

- React-create
- Routing

Integration Flow

- Services
- Store
- routing
  - Just use <></> in the <Routes> tag while you get Login working
- cofig
  - update ports
- Login

notes

- use fake data to test if the backend is not ready with live data

Team Notes

- Decide how you will organize your build.
  - 1 team for the backend, 1 team for the frontend
  - each team member builds out 1 route or set of routes at a time.
  - ???
- Use Github starting early.
  - Check out the video from Session 3 on this link to learn how to work together on Github

        - https://sites.google.com/codingdojo.com/part-time-bb/blogs/git
- Communicate!