

Studio: MVC Design

This is a shorter-than-usual, design-only studio that will be included in class at the discretion of your instructor.

Designing and MVC Application

As you move from **TechJobs (Console Edition)** (`../../assignments/techjobs-console/`) to **TechJobs (MVC Edition)** (`../../assignments/techjobs-mvc/`), you'll begin to use your newfound knowledge of controllers and views in Spring Boot. One of the things that can be tricky about dipping your toes into the MVC waters is figuring out how to structure your applications. While there are some broad best-practices, there are few rules that will specifically direct you at each turn.

This exercise is a chance for you to think about how to structure an MVC application -- well, the V and C portions, at least.

In IntelliJ, open up your `techjobs-console-java` project. We want to consider how we might provide the same functionality in the view and controller portions of a Spring Boot app.

Your Task

With pen and paper, a whiteboard, or your favorite notes app, sketch and list out the following:

- Which controllers would you build? What would each be responsible for? A controller should contain like functionality, and not be a random collection of unrelated handler methods.
- What would the handler methods be for each controller?
- What are the screens that the user would interact with?
- For these screens, what are the specific view files you would need to create?
- Which handler methods in which controllers would use each view?
- What would the routes / URL paths be for your application?

While there isn't one "correct" answer to these questions, there are some approaches that will be better than others. Think about the user experience, avoiding code repetition, and avoiding putting unrelated functionality in the same files.

