

Java III FAQs

So what's Java III going to cover?

We will dive deeply into building full stack web applications using portions of Java EE (Java Enterprise Edition). You will be introduced to several technologies over the course of this section. The applications we will build will be organized in an MVC way. Our Git workflow will expand to encompass the use of tags.

What the heck is MVC?

The Model View Controller pattern is an architectural style for web applications where a web application's logic / files are separated into three main concerns...

1. interacting with persistent data / databases (The Model code)
2. generating a view / HTML & CSS & JS for the client/user (The View code)
3. listening and responding to requests, often HTTP requests, from a client/user (The Controller code) MVC is not a specific technology, but rather a best practice of how to organize the logic of a web application. It is widely used but there are other architectural styles / approaches in web development.

What are Git tags?

Git tags are simply named commits. They are typically used to document a specific commit of an application and often are used to designate a specific version of a project. You can checkout to them like any other commit. In order to make any additional commits / changes, you must first checkout the tag, then create a new branch off of the tag. You will not need to make your own tags for the exercises, only branches off of tags. Please note that *you should have a clean working directory (i.e. everything is committed) before switching tags.*

How are we going to use Git tags in Java III?

During the second Java III exercise (for the lesson on Servlets), you will fork and clone a repo of a partially completed Craigslist clone into IdeaProjects. We refer to this as the Adlister project. This repo contains two tags for every remaining exercise in the Java III section. For each lesson, one tag exists where the Adlister is in a state to begin the exercise; another tag exists with a completed solution of the exercise for you to see how your code compares. You can simply copy

and paste the git commands in the curriculum exercises to checkout to the correct tag for each exercise.

So, I thought we were going to build our own Craigslist clone/adlister.

Once we complete the Java III lessons, we will likely have several days to continue building much more functionality into the Adlister project in teams of 2-3. Your team will add additional features and styling.

How do I use Tags correctly to complete the exercises?

1. Checkout to the correct exercise tag for the current lesson (not the solution tag).
2. Create your own branch (name it according to exercise instructions).
3. Complete the exercise.
4. Add and commit your work on the branch.
5. Push up the branch to your forked repo (MAKE SURE YOU PUSH TO YOUR FORKED VERSION!).
6. Checkout to the solution branch to compare to your own solution and to follow along with instructor walk-throughs.

What will be on the Java III assessment?

You will be given a partially built application and you must finish building the features specified in the assessment instructions. You will have a way to run tests to see what features pass and fail as you work through the assessment.

Why are we working with so many files when building web applications?

Why are books broken into chapters? Why are chapters broken into pages? Paragraphs? Sentences? Words? Letters?

Separating our code across many files may make it harder to understand how everything works initially, but separation greatly organizes the process of maintaining and building an application. Also consider that as an application scales upward in size and complexity, potentially hundreds of developers may need to work on the application at the same time. Think of the merge conflicts if everyone worked on the same file!

What is the point of Maven?

Maven solves a very similar problem for Java that we used npm to solve for in the JavaScript II section. By editing a special file, called pom.xml, we are able to list any extra

libraries/dependencies our project needs (much like the package.json file does for npm). Maven is more than just a dependency manager, however, and controls the entire build lifecycle of our application. You can read the documentation to better understand how this lifecycle works.

What is Tomcat?

Apache Tomcat is a wrapper around our application and functions as both a web server and web (servlet) container. In other words, Tomcat is responsible for receiving and sending HTTP requests (the web server part) and creates the required instances of our Java code, including servlets, for our application logic to do something with those requests (the web container part).

How can I maximize my success in the Java III section?

Always keep in mind that the only bad question is the one you don't ask.

In addition to completing the exercises, being engaged with lectures, asking questions and seeking assistance, you will find the following tips very beneficial:

1. Skim through the slides at least the day prior to when we cover them in class.
2. Experiment (play) with your existing code as you develop a solution and after completing the exercise.
 - What happens if you change a method call?
 - Can you refactor it to make it easier to read?
 - If you break it, do you understand how and why you broke it?
3. Watch additional videos, read tutorials, read documentation, and do extra research to more deeply understand the topics we cover.
4. If you are working ahead, work on your own separate application along with the curriculum Adlister project
 - Limit the tables to keep it simple at first.
 - Having this side project will mean you get to practice everything at least twice.

What is Spring and how does it relate to all of this?

Spring is a Java web framework that has become a leading technology in building Java web applications. Many parts of back end development in Java III are streamlined or abstracted away in Spring. Understanding how the main parts of Java EE work without a framework will give you a deeper perspective full stack development and of how a web framework simplifies the process.

