

Exercises: Servlet, JSP, and MVC Review

There are *definitely* more exercises here than there is time available to do them. I listed several exercises so that you could pick and choose depending on your level of servlet/JSP experience.

1. Open Eclipse. Start Tomcat (R-click on the Tomcat entry at the bottom, “Start”) and make sure you can view the home page at `http://localhost/`. Deploy the “intro” project (R-click on the Tomcat entry at bottom, Add and Remove Projects, choose “intro”, R-click on Tomcat, restart). Make sure you can run the HTML pages, JSP pages, and servlets.
2. Make a new Dynamic Web project in Eclipse. Go into the “intro” Eclipse project and copy `Hello.html` and `Hello.jsp` to the `WebContent` folder of your new project. Deploy the project and make sure you can run `Hello.html` and `Hello.jsp`. Remember that URLs are case sensitive, so unless you renamed the file, it is `http://localhost/projectName/Hello.html`, not `http://localhost/projectName/hello.html`.
3. Copy `HelloServlet2.java` from my “intro” project and put it in the `src/coreservlets` folder of your new Eclipse project. Give it a custom address in `web.xml` and run it.
4. Make a very simple online banking application using MVC that collects a customer ID and outputs the first name, last name, and bank account balance of the customer with that ID. It should give an error page for an unknown ID. To simplify this, there are two simple classes from the `coreservlets` package in the `review-exercises` project: `Customer.java` and `CustomerUtils.java`. The first represents a customer and the second makes a simple `HashMap` of customers and lets you retrieve a `Customer` based on an id (`CustomerUtils.getCustomer`) or a `firstname/lastname` pair (`CustomerUtils.getNamedCustomer`). Just grab these two classes and drop them in your project. Don’t peek at the `ShowBalance` class until you finish this problem: it is the main part of the problem solution.
5. Make a JSP page that randomly selects a background color for each request. Just choose at random among a small set of predefined colors. Be sure you do not use a style sheet, since it overrides the colors given by `<BODY BGCOLOR="...">`.

Note: in Eclipse, *all* Java code goes in `src/subdirectory-matching-packageName`, regardless of whether the Java code is a servlet or some other kind of Java class. So, if you have a class called `ColorUtils` with a static method called `randomColor` and it is in a package called `utils`, you put `ColorUtils` in `src/utils`, and you call it with `utils.ColorUtils.randomColor()`. Also, if you want to test some Java code separately from a JSP page, and your class has a “main” method, you can right-click anywhere in the file, choose `Run As --> Java Application`, and Eclipse will run `main` and put the output in the console at the bottom.