

Module 3 Coding Assessment

This assessment helps you validate your understanding of the module objectives:

- Controllers
- Views
- Passing data from controllers to views
- Handling Request data
- View logic
- Dynamic link creation
- Query parameter access

It also gives you practice with coding assessments you may encounter during the job interview process.

Overview

This is a time-boxed, individual coding assessment. You have one hour to complete as much of your assigned problem as you can. You are not expected to complete all features of the problem in the time given.

You may use any resource available in the classroom except another Human Being. This is an individual effort.

Whatever you complete should not have compile or run-time errors. Any features you can complete should run successfully.

If you are nearing the end of the time-box and you have compile/run-time errors consider commenting out or removing the code causing the errors before you push your final effort.

At the end of the time-box, you should push your final effort to your class repository.

You are expected to remain in the classroom until you have completed the coding assessment.

If you finish before the end of the time-box, please be respectful of those still working. Please leave the classroom if you would like to talk with others that have completed the coding assignment.

Instructions

1. Pull from your class repository like you do for code every day. The assessment is in the `module-3` folder of your class repo folder.
2. Open a terminal session and navigate to the folder: `~/workspace/yourname-java/module-3/00-Assessment/java/database`
3. Run the command: `sh import-database.sh`. This will create the database and table you need for the assessment as well as populate the table and display it's contents.
4. Go to Eclipse and import the project in the `00-Assessment` folder.
5. The `m3-java-assessment` project contains:
 - Starter code for all required controllers
 - Complete DAO and POJO code for accessing the database
 - An exception class used by the DAO code
 - Common header and footer JSP files.
 - Starter code for all required views (JSP files)
6. Tasks to complete:
 - Create a controller action in `PuppiesController` mapped to the root `/` that calls the `getAllPuppies()` method in the DAO and passes that information to the `puppyList.jsp` view.
 - Change the `puppyList.jsp` view to show all the entries from the controller in the provided table. The weight should have `lbs` after it. The `paperTrained` value should show as 'Yes' when true and 'No' when false.
 - Create a controller action in `PuppiesController` that:
 - Takes information from the form on `puppyList.jsp`
 - Saves that information as a new entry in the database by passing it to `savePuppy(Puppy)` in the DAO.
 - Redirect back to the controller action defined in the first task.
 - Create a controller action in `PuppiesController` mapped to `/puppy` that takes an id as a query string parameter. It calls `getPuppy(int)` in the DAO and passes that information to the `puppyDetail.jsp` view.
 - Change the `puppyList.jsp` view to make the name of the puppy a link to the controller action/path created in the previous step.