

MSSE670_Week4_Assignment

Thursday, September 1, 2022 3:57 PM

Week 4 Assignment, Services Layer Part 2

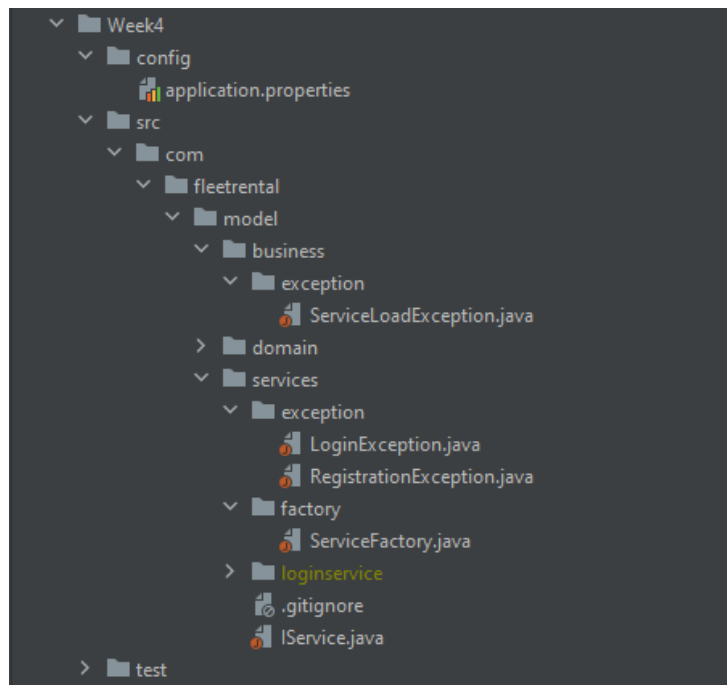
Important Notes

- This week, a project must run as viewed from both the perspective of the tests and the Driver Class to receive a passing grade. In your Driver class, you should instantiate and configure the Composite object, pass it to the services, and print returned output from the methods.
- Be sure to include screen shots of sample runs and passed tests in your docs folder.

Plan

- Refer to the **Fleet Rental Sample Code** for details about how to program your project.
- You will deliver the next iteration of your course project. This week, the Services Layer will be enhanced.
- This week you will:
 - Implement at least two services in addition to the Login Service. (If you implemented more than one last week, you do not have to do anymore right now).
 - Implement exceptions for all services in your application.
 - Modify your service layer unit tests to catch exceptions.
 - Update your domain objects so that they can be serialized and de-serialized.
 - Update your service layer interfaces using a Marker Interface design pattern.
 - Refactor your service factory to decouple interfaces to your services from their implementations.
 - In a config directory, at the same level as your source directory, you will create an application.properties file which will allow you to configure service implementation that will be loaded at runtime.
 - The application.properties file must be loaded at runtime using the following (spaces are important).
-Dprop_location=<pathname>
- You will still be working in the services layer of the existing structure, however, you will add a business layer, that will include a ServiceLoadException:

Code View of Structure



What to turn in

You will implement your Services Layer following the details above. The should include:

- Appropriate naming conventions and style (see sample code).
- Packages in the Services Layer, including the refactored Factory and Marker Interface.

- At least two fully implemented services in addition to the Login Service using Interfaces and Implementations (must include at least the 4 CRUD methods).
- Exceptions in the Business Layer and Services Layer.
- An **application.properties** file that is used to load the implementation of the services at runtime.
- A Driver Class with a main method that runs the program. This goes above the Services Layer. You can put it in the Business Layer for now, but eventually it will should be in the Presentation Layer.
- Tests for the Services Layer including tests that handle exceptions.
- A docs (see location in the Sample Code) folder with screen shots of completed runs using the Driver as well as successful test runs. This week, you must manually trigger exceptions and catch them in your testing.
- Code uploaded to a source control repo and link submitted in slack. Code for Week 4 must be clearly labeled in a folder or branch.