Practice

Create Microservices by Using Spring Boot

Exercise

- Shopping Application

PRACTICE

Shopping Application

Consider a shopping application that enables customers to shop for products on any smart device. It provides multiple features to all its registered customers.
The customers can add products and delete products from the shopping list. Additionally, they can view all the products on the shopping list. Build microservices for authentication and product service.

Steps to Access the Features of Application

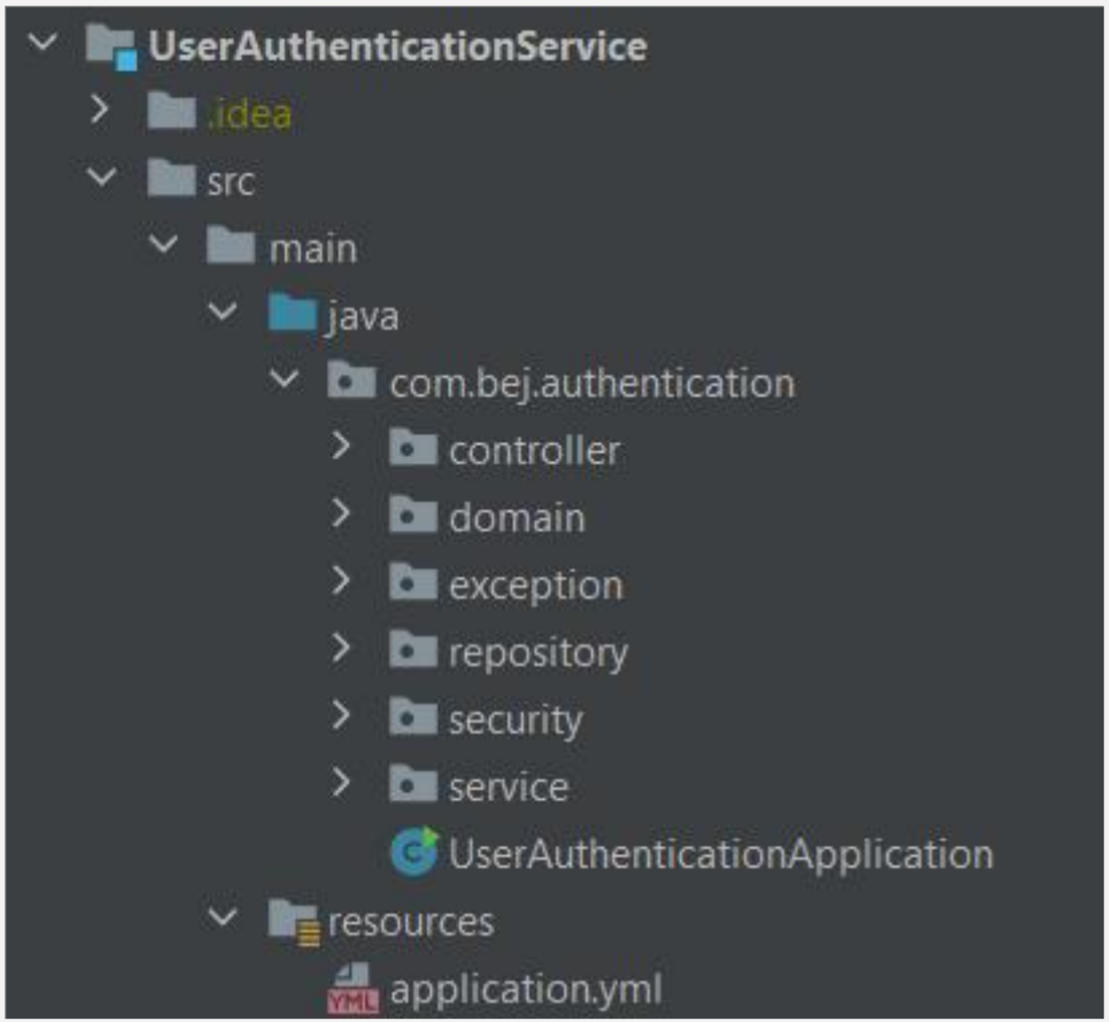
- The user must first register with the application
- The user must login with valid credentials
- After login, a JWT is generated that provides access to the features of the application
- The user can access the features provided:
  - Add products to the shopping list
  - Delete a product from the shopping list
  - View all the products in the shopping list

Instructions for the Practice

- Click on the boilerplate.
- Fork the boilerplate using the fork button
- Select your namespace to fork the project.
- Clone the project into your local system.
- Open the project in the IntelliJ IDE.

Task – Practice 1

- In the parent project called `ShoppingApplication`. change the parent `pom.xml` and add the necessary dependencies
- The `UserAuthenticationService` uses the MySQL database and stores the user credentials like `userID` and `password`.
- The `UserProductService` uses MongoDB and stores all details of registered customers, the Product information like `productID`, `name` etc.,
- Add `UserAuthenticationService` and `UserProductService` to the parent pom of the `ShoppingApplication` as modules

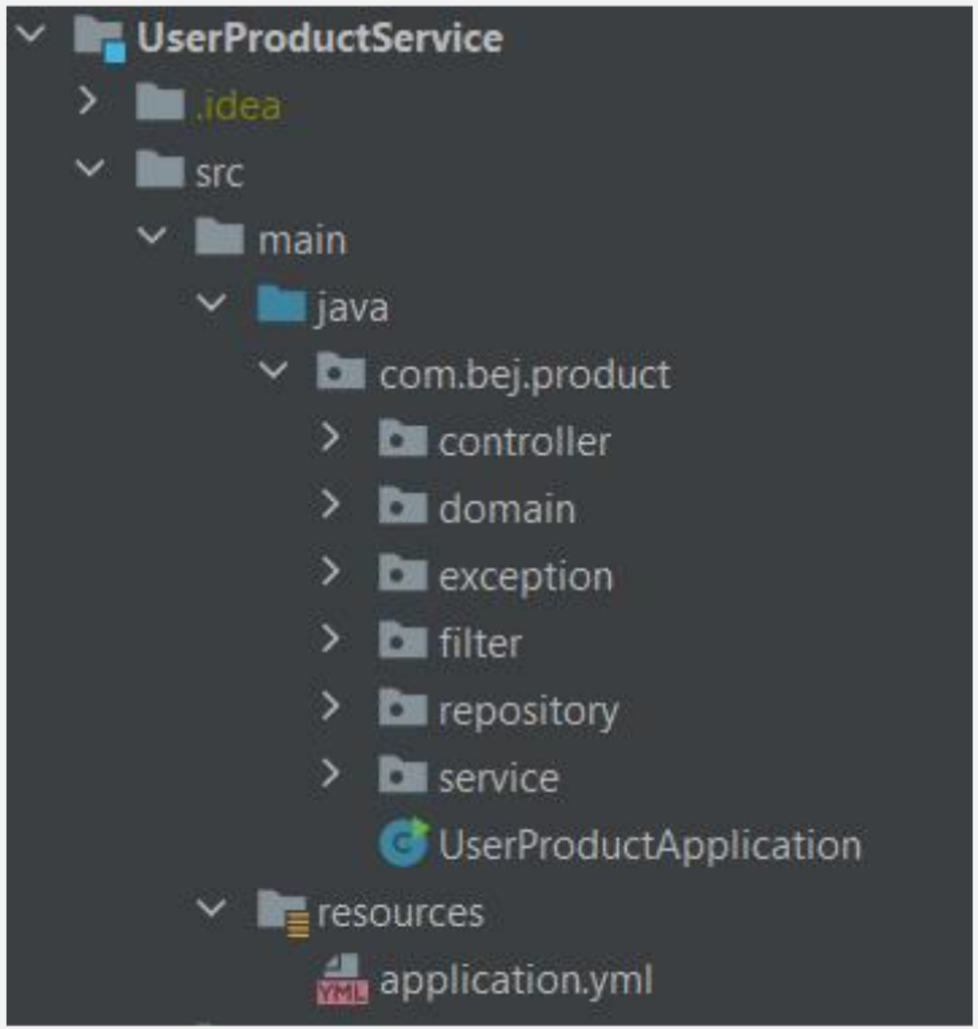


Create UserAuthenticationService REST API

- The layers of the REST application are defined .
- The User domain class that will hold the user credentials for login.
- Handle all exceptional situations.
- Secure the application by generating a JWT at login.
- Write implementation for the methods for save user and get user by userId and password to validate user credentials at login in the service layer.

Create UserProductService REST API

- The layers of the REST application are defined .
- Declare a JWT filter to filter and allow users to access the features only on login and with a JWT token in the authorization header.
- Set the `userId` in the JWT claims, so that it can be accessed for all the requests.
- Define all methods for registering a user, save a product to the user product list, delete a product from the product list and get the product list of the user in the service layer.
- In the main class specify the filter condition with `FilterRegistrationBean.`



Executing the Application

- Execute the test cases provided in the test folder.
- Test the output in Postman.
- Register the user in the `UserProductService`.
- The user credentials must be saved before a user can access the features of the application.
- Hence save the basic credentials like `userId` and `password`.
- Save the `userId` and `password` of the user in `UserAuthenticationService`.
- Login to the `UserProductService` to add, delete and view products.
- At log in the jwt token must be generated.

Submission Instructions

- Submit the practice or challenge on hobbes.
- Login to hobbes using your credentials.
- Click on **Submission** in the left navigation bar.
- The **Submit for evaluation** page is opened.
- Select the solution repository `bej-create-microservices-pc-1-shopping-application` against which your submission will be evaluated, under **Assignment Repository**
- Select your solution repository s2-pc-1-shopping-application under **Search Submission Repo**
- Click on **Submit**.
- The results can be viewed in the **Past Submissions** screen.