Challenge

Create Microservices by Using Spring Boot

Music Streaming Application

Implement a music streaming application that enables users to stream music on any smart device. It should provide multiple features to all its registered users like adding and deleting a track from the playlist, updating a specific track, displaying all the tracks, etc.

The user will need to register with the application to access some of its features. Use the Microservices approach to implement the application.

CHALLENGE

Steps to Access Features of Application

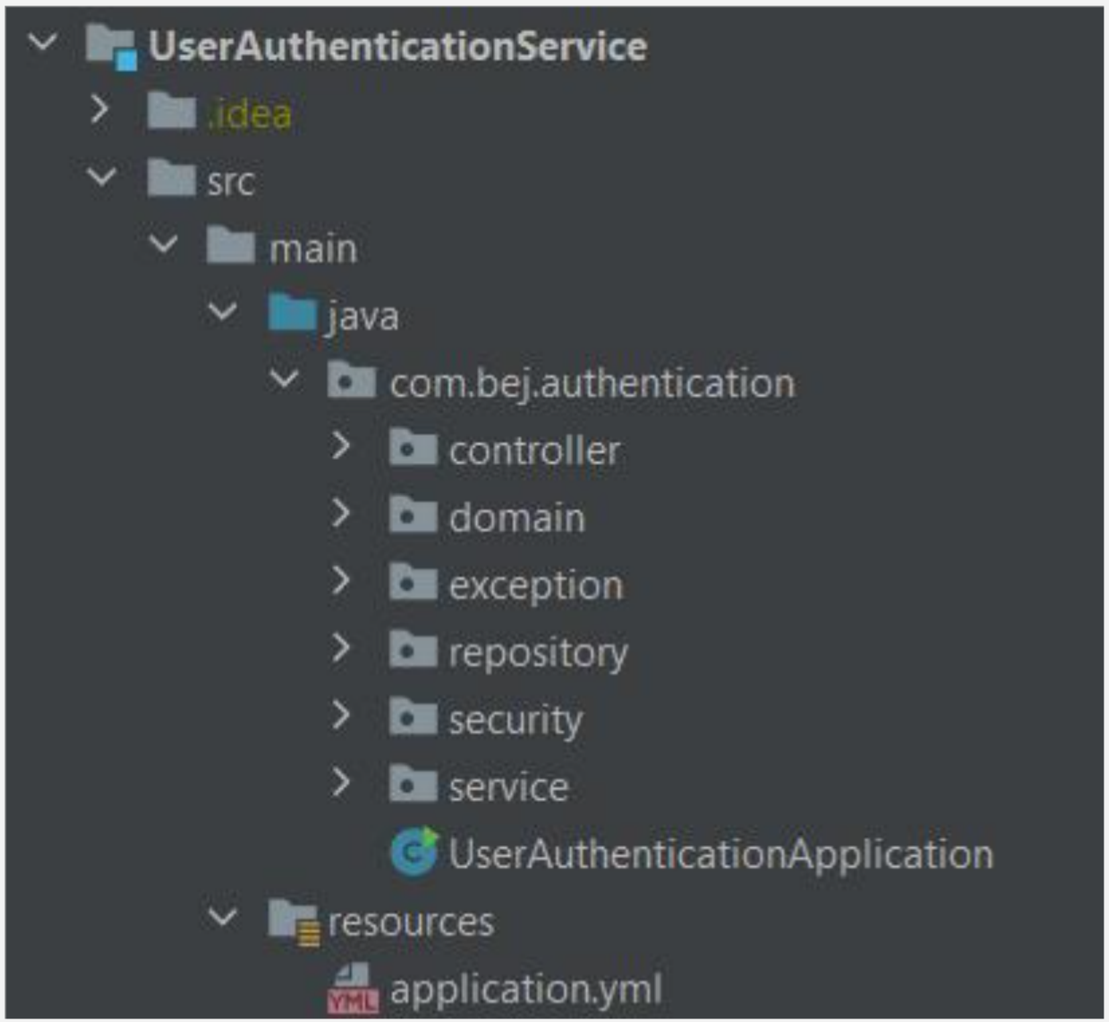
- The user must first register with the application.
- The user must login with valid credentials.
- After login, a JWT is generated that provides access to the features of the application.
- The user can access the features provided to:
  - Add tracks to the playlist
  - Delete a track from the playlist
  - Update a track in the playlist
  - View all the tracks in the playlist

Instructions for the Challenge

- 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 `MusicApplication`. change the parent `pom.xml` and add necessary dependencies
- The `UserAuthenticationService` uses the MySQL database and stores the user credentials like `userID` and `password`.
- The `UserTrackService` uses MongoDB and stores all details of registered users, the Track information like `trackID`, `name` etc., The Track must contain an `Artist` object, the artist will have an id and name.
- Add `UserAuthenticationService` and `UserTrackService` to the parent pom of the `MusicApplication` as modules

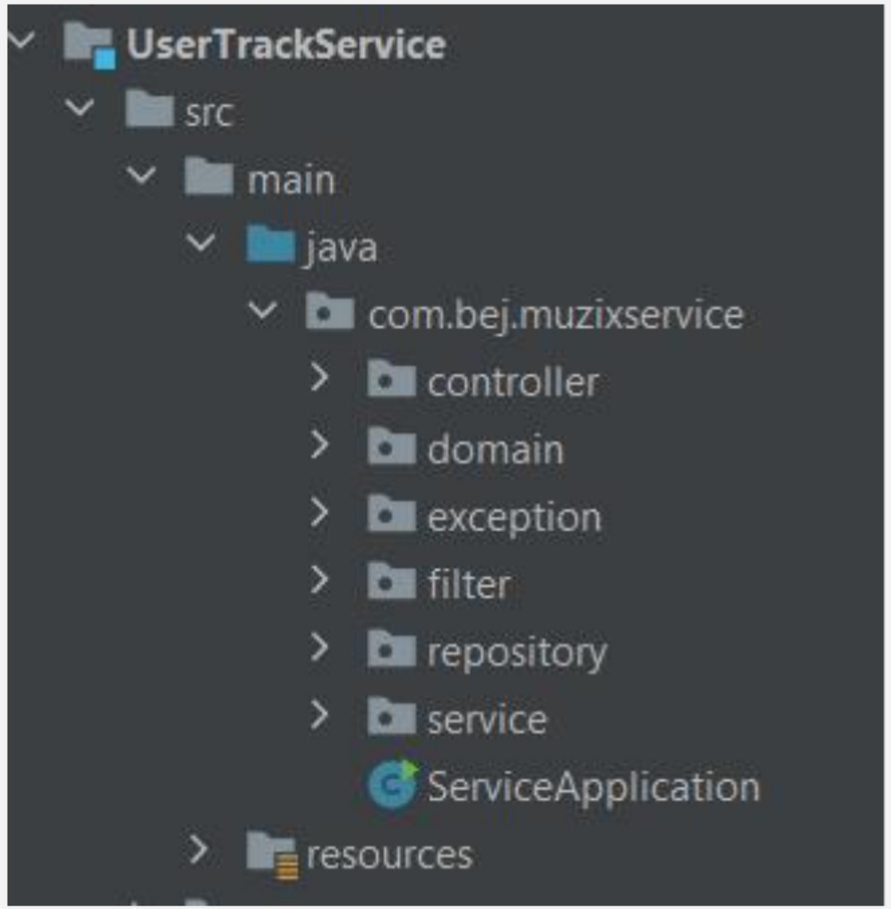


Create UserAuthenticationService REST API

- Define the layers of the REST application.
- Create 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 methods to save the user and get the user by userId and password to validate user credentials at login in the service layer.
- **Note :** The `UserAuthenticationService` created during the practice can be used here. The service need not be created from scratch.

Create UserTrackService REST API

- Define the layers of the REST application
- 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, saving a track to the user playlist, deleting a track from the playlist, updating details of a specific track in the playlist, and getting the tracklist of the user in the service layer.
- In the main class specify the filter condition with `FilterRegistrationBean`



Executing the Application

- Register the user in the `UserTrackService.`
- 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 `UserTrackService` to add, delete, update and view tracks.
- 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-mc-1-user-track-service` against which your submission will be evaluated, under **Assignment Repository**
- Select your solution repository `s2-mc-1-user-track-service` under **Search Submission Repo**
- Click on **Submit**.
- The results can be viewed in the **Past Submissions** screen.