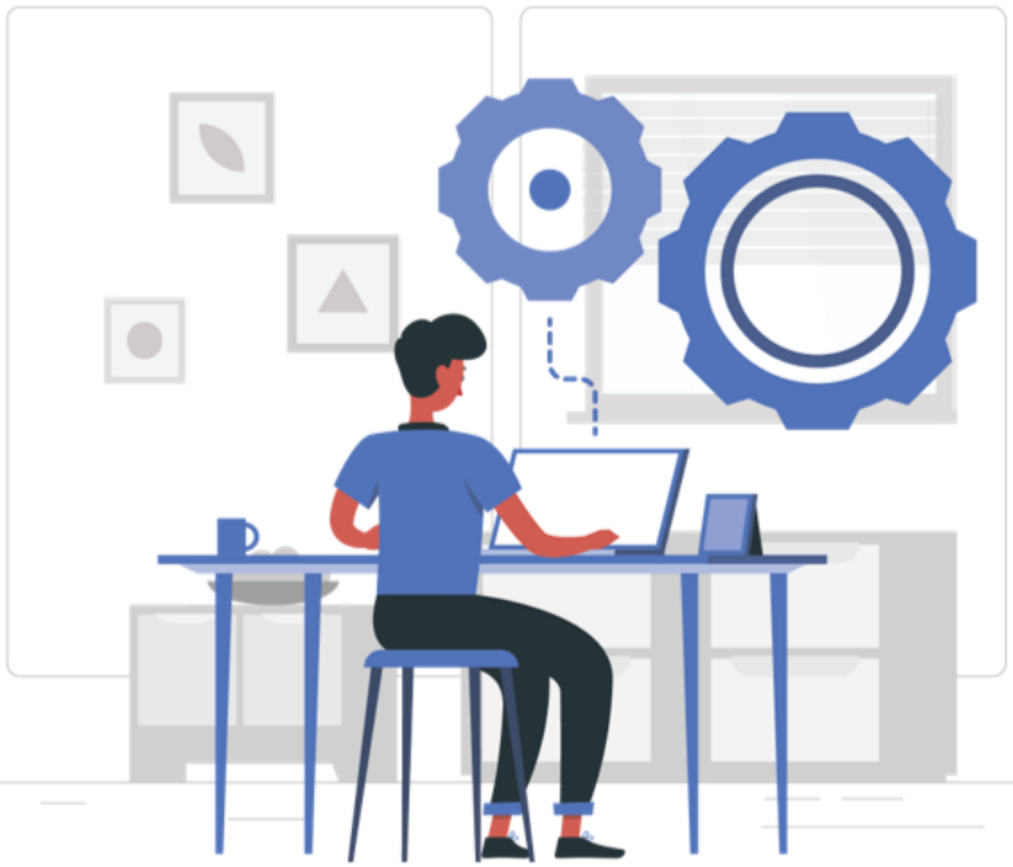


Challenge

Create a Single-entry Point to Route the Request Coming for Different Microservices Using Spring Cloud

Music Streaming Application

Implement a music streaming application that enables users to stream music on any smart device. The application should provide multiple features for all its registered users like adding a track to a playlist, deleting a track from the playlist, updating a specific track, displaying all the tracks in the playlist, etc.
The user will need to register with the application to access some of its features.
Use the microservices approach to implement the application.
Also implement Spring Cloud Gateway.

CHALLENGE

API Gateway

Clients

/user

/register

/login

The clients send request for a service.

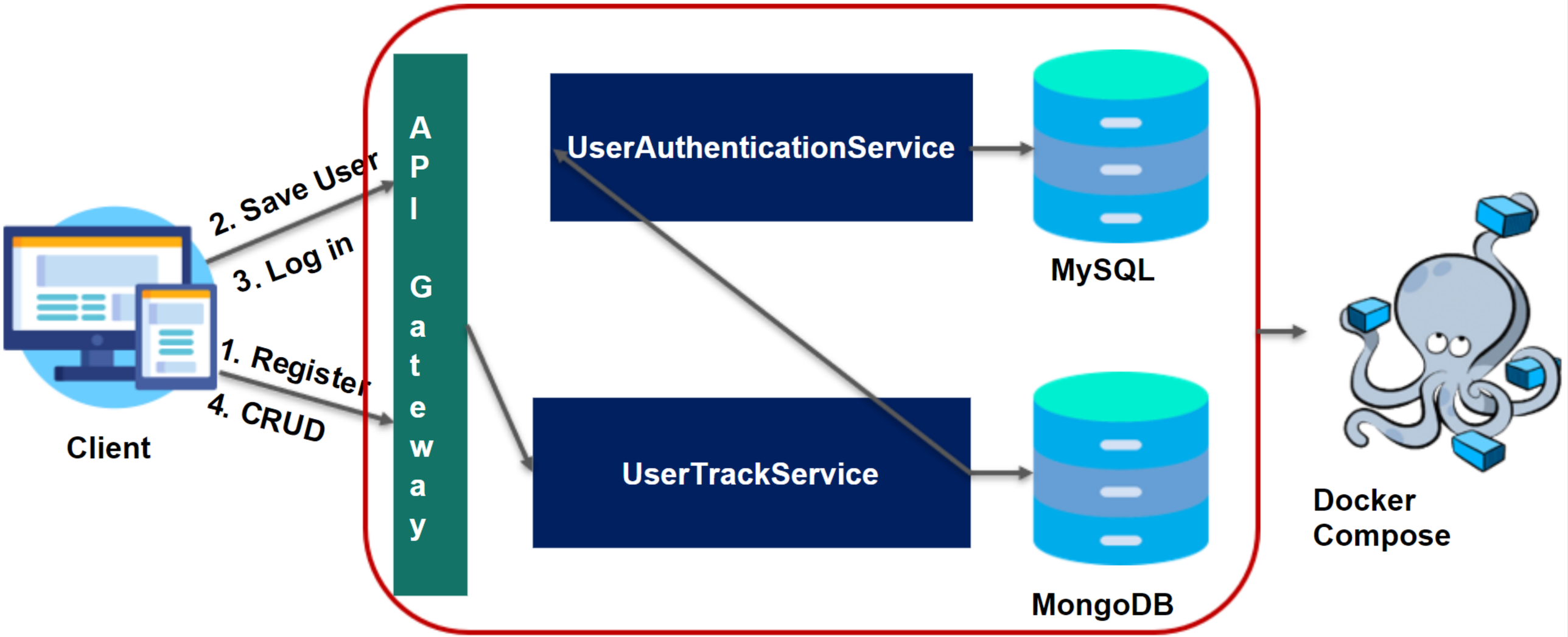
A P I G a t e w a y

The API Gateway routes the request to the service requested.

User Service

Track Service

How Does the Application Work?



Instructions for the challenge

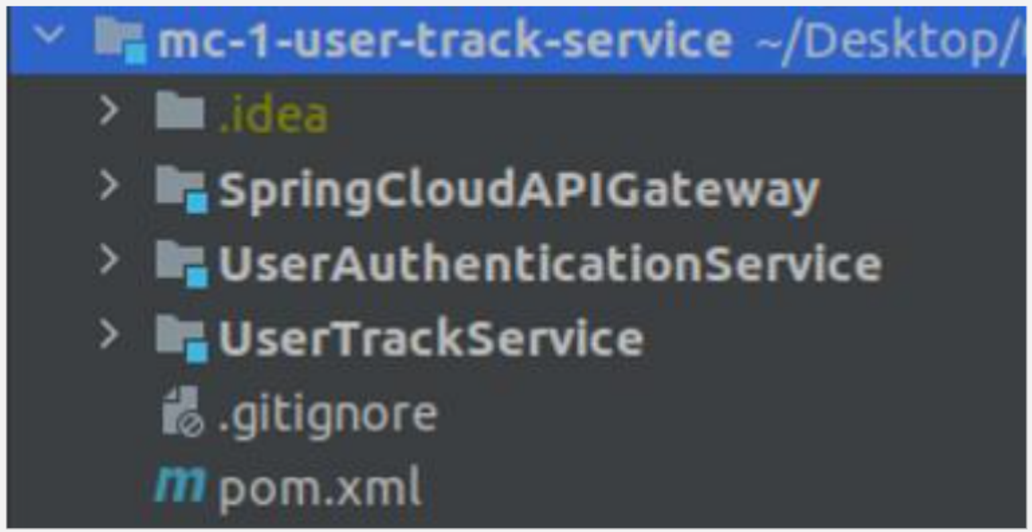
- Click on the boilerplate.
- Fork the boilerplate using the fork button
- Select your namespace to fork the project.
- Clone the project into your local system.
- Open the project in the IntelliJ IDE.

Task: Challenge 1

- Create a new project using the Spring Initializer to build the Spring Cloud API Gateway.
- Add the dependencies for the Spring Cloud Gateway in the pom.xml.
- Add the Spring Cloud API Gateway in the module of the parent pom.
- Configure the routes in the API Gateway.
- All services must be routed through the API Gateway.
- The API gateway must send the request to the corresponding service.
- Test the output in Postman.

Shopping Application – How to Access Features

- The user must first register with the application
- The user must log in 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



The structure of the Application

Task: Challenge 2

SpringCloudAPIGateway is the service that will act as API Gateway.

UserAuthenticationService is the service that will have the login and save the user functionalities.

UserTrackService is the service that will have all the CRUD functionalities related to Track object.

pom.xml is the parent pom.

Submission Instructions

- Before pushing the solution to the repository,
  - In the `application.properties` file there are two configurations to execute the application, one for local execution and other for Hobbes execution.
  - When executing the application on your local machine, comment the hobbes configuration and uncomment the local configuration and change username and password to connect to database as per your local config.
  - Before pushing the solution to the repository comment the local configuration and uncomment the hobbes configuration.
- Push the solution to git.

Submission Instructions (contd..)

- 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-spring-cloud-api-gateway-c4-s3-mc-1-music-streaming-application` against which your submission will be evaluated, under **Assignment Repository**
- Select your solution repository `s3-mc-1-music-streaming-application` under **Search Submission Repo**
- Click on **Submit**.
- The results can be viewed in the **Past Submissions** screen.