

Project Instructions

1. Build the JAR File

To build the JAR file for your Spring Boot project, follow these steps:

1. **Navigate to the Root Directory of the Project:** Open a terminal or command prompt and navigate to the root folder of your project where the `pom.xml` file is located.
2. **Run the Maven Command to Build the JAR:** In the root directory of your project, execute the following command to clean, install, and skip tests during the build process:

```
mvn clean install -DskipTests=true
```

You should see a `.jar` file with the name of your project.

2. Build the Docker Image

To build the Docker image for your Spring Boot project, follow these steps:

1. **Navigate to the Folder Containing the Dockerfile:** Ensure you are in the root directory of your project or the directory where the `Dockerfile` is located.
2. **Build the Docker Image:** Run the following Docker command to build the Docker image:

```
docker build -t lms-image:latest .
```

3. Run PostgreSQL & Spring Boot Containers

To run PostgreSQL and Spring Boot containers, use Docker Compose to manage the containers. Follow these steps:

1. **Navigate to the Folder Containing `env.yml`:** Open a terminal where `env.yml` (Docker Compose configuration file) is located.
2. **Run Docker Compose to Start Containers:** Execute the following command to start PostgreSQL and Spring Boot containers in detached mode:

```
docker-compose -f env.yml up -d
```

This command will:

- Start the containers based on the configuration in `env.yml`.
- Run the containers in the background .

4. Send the Register and Login Requests

After starting the containers, interact with the API to perform user registration and login to retrieve the token.

1. **Send a Register Request:** Use **Postman** or another API client to send a `POST` request to the registration endpoint.

POST

url: <http://localhost:8080/users/register>

Body raw (json)

```
json {  
  
  "email":"marwa4@gmail.com",  
  
  "password":"12345678",  
  
  "name":"marwa"  
  
}
```

2. **Send a Login Request:** Once registration is successful, send a `POST` request to the login endpoint to obtain a JWT token.

POST

url: <http://localhost:8080/login>

Body raw (json)

```
json {  
  
  "email":"marwa3@gmail.com",  
  
  "password":"12345678"  
  
}
```

3. **Retrieve the Token:** The login response will contain a JWT token. Copy this token as it will be required in the next steps.
-

5. Install Python Libraries

Before running the Python script, ensure that all the required libraries are installed. Follow these steps:

1. **Install the Required Libraries:** Install the necessary libraries by running the following command:

```
pip install requests faker pandas
```

6. Run the Python Script

With the environment set up and libraries installed, you can now run the Python script. Here's how you can do it:

1. **Modify the Python Script:** Insert the token you received from the login response into the Python script. Replace `your_jwt_token` with the actual token in the script:

```
headers = {  
  
    "Authorization": "Bearer your_jwt_token",  
  
    "Content-Type": "application/json"  
  
}
```

Then Run the script