Social Media Application

Introduction

This document will provide a step-by-step implementation of making the social media application. We will then explore the implementation of dev tools and actuators in the upcoming lectures.

1. Step 1

- Create an entity class **Connection** with the following attributes.

    1. *id (Integer)*
    2. *name (String)*
    3. *emailId (String)*
    4. *company (String)*
    5. *username (String)*
    6. *level (String)*

- Add properly required annotations to make this class an entity and generate the getters and setters for the attributes.

2. Step 2

Create a DTO class **ConnectionResponseDto** with the following attributes and with their getters and setters.

- name (String)
- emailId (String)
- company (String)
- username (String)
- level (String)

3. Step 3

1. Create a **UserController** class with the following APIs:

    - GET *localhost:8080/ninjas/connections* - retrieve a list of a user's connections.

    - POST *localhost:8080/ninjas/add* - add a connection to the user.

    - GET *localhost:8080/ninjas/connections/{company} (path-variable - String company)* - Fetch a list of all connections of a user by company.

2. Autowire the required service layer objects

3. Add proper annotations for methods according to the method parameters and mapping.

4. Step 4

Create a service class named **ConnectionService**

- Add proper annotations for the class.
- Write methods in this class corresponding to the methods in the **UserController** class.
- Autowire the required dal layer object.

5. Step 5

- Create a repository class to interact with the database named **ConnectionDal.**
- Extend the class with the required **JpaRepository** interface with the correct parameters.
- Create a custom method *findByCompany* in the **ConnectionDal,** which will fetch a list of all connections by a company by taking a company in a string as an argument.
- Add the required annotation for this repository class.

6. Step 6

Configure the application.yml file as shown below.

```
spring:
   application:
       name: social_media
   datasource:
       password://your password for my sql
       url:jdbc:mysql://localhost:3306/socialmediadb
       username: //your sql username
   jpa:
       generate-ddl: true
       hibernate:
           ddl-auto: update
       properties:
           hibernate:
               dialect: org.hibernate.dialect.MySQL8Dialect
       show-sql: true
```

Refer to the template below for more clarity:
[Template Link]