**Spring Boot REST**: Spring Data with Hibernate Case study Problem

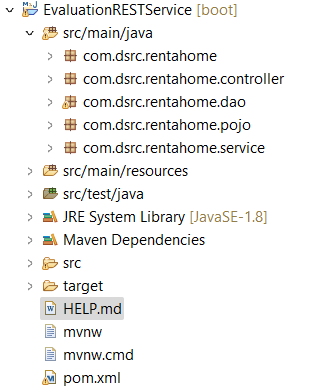
**Task**: You are working on a House Renting Application like 99Acres.com. You are assigned the task of creating back-end REST service for this application. You need to complete the Rented House Details Module. Use MySQL as database server.

Use POSTMAN to test the Application.

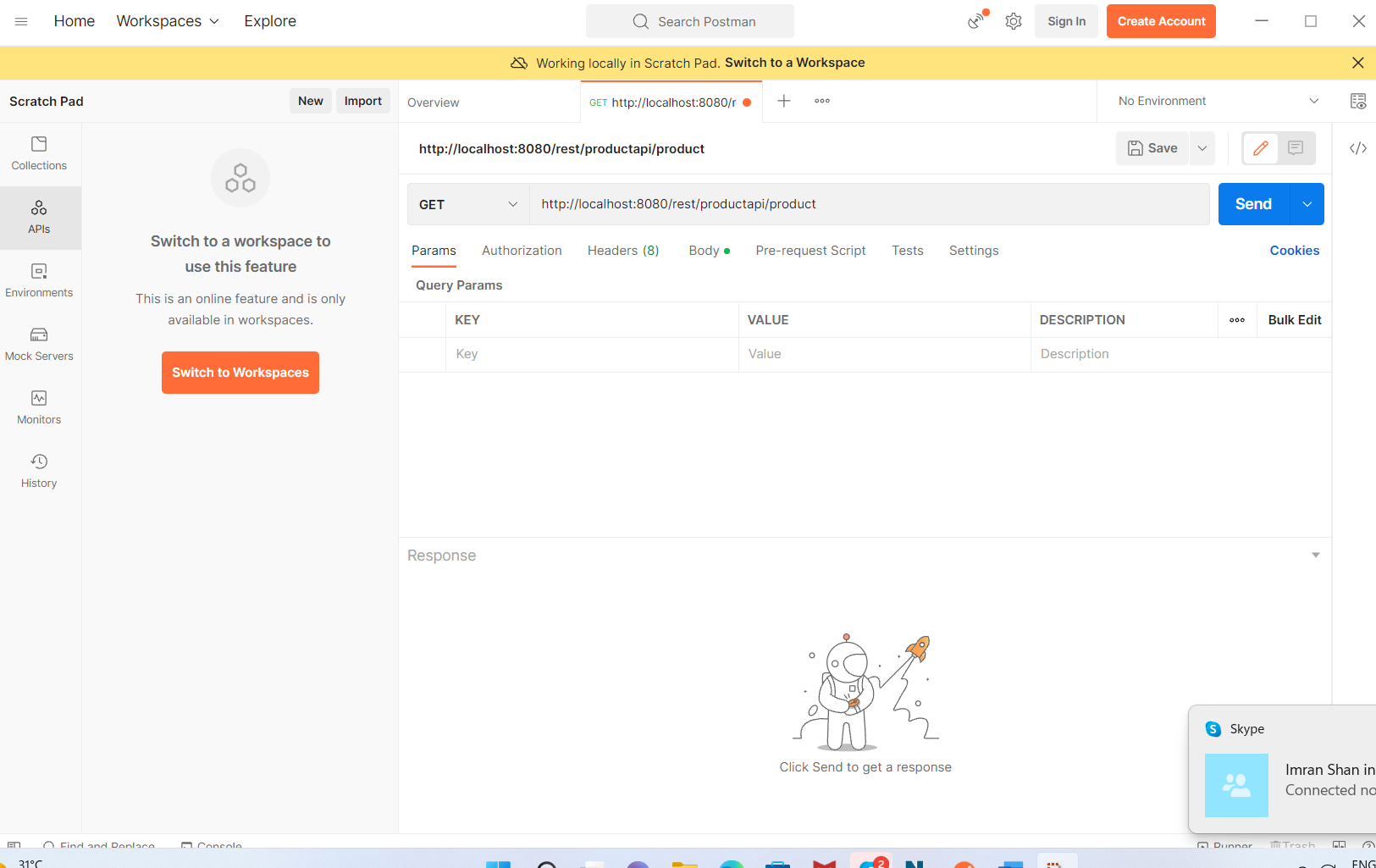
1. Create a database named “rentahome”
2. Create a table named “HouseData” as follows.

|  |  |  |
| --- | --- | --- |
| Column Name | Data type and Size | Constraints |
| AdNo | int | Primary key |
| HouseType | Varchar , 20 | Not null |
| Location | Varchar ,20 |  |
| BuildingAge | int |  |
| CarParkingAvailable | Varchar,5 |  |
| BikeParkingaAvailable | Varchar,5 |  |
| AskingRent | int |  |

1. Create a spring boot application of category Spring Web and spring-data support
2. Add the configurations in application props
3. Project Architecture as follows.



1. The following are the functionalities to be implemented in application
2. Service Layer must be implemented in an Interface -> Implementation Style
3. Service Layer methods
   1. Boolean addProperty(House house)
   2. Boolean editProperty(House house)
   3. Boolean deleteProperty(int adno)
   4. List<House> searchByLocation(String location)
   5. List< House > searchByHouseType(String type)
   6. List< House > searchByRentAsking(int fromprice, int toprice)
   7. List< House > searchByCarParking()
   8. List< House > searchByBikeParking()
4. DAO Layer must be implemented in an Interface -> Implementation Style
5. All the above service layer methods have to be implemented in the Dao layer.
6. The REST URLs and description is given below.
   1. [http://localhost:<<port>>/rentahomerest](http://localhost:%3c%3cport%3e%3e/rentahomerest)/home
      1. Should accept house details and save to database
      2. POSTMAPPING to be Used
   2. [http://localhost:<<port>>/rentahomerest](http://localhost:%3c%3cport%3e%3e/rentahomerest)/home
      1. Should update the house details based on ad no
      2. PUTMAPPING to be Used
   3. [http://localhost:<<port>>/rentahomerest](http://localhost:%3c%3cport%3e%3e/carsellrest/)/home/{adno}
      1. Should accept ad no and delete home from db.
      2. DELETEMAPPING to be Used
   4. [http://localhost:<<port>>/rentahomerest/](http://localhost:%3c%3cport%3e%3e/carsellrest/)searchbyhometye/{type}
      1. Return all homes based on the type passed (eg . 1BHK ,2BKH etc)
      2. GETMAPPING to be Used
   5. [http://localhost:<<port>>/rentahomerest/](http://localhost:%3c%3cport%3e%3e/carsellrest/)searchbylocation/{location}
      1. Return all homes based on the location
      2. GETMAPPING to be Used
   6. [http://localhost:<<port>>/rentahomerest/](http://localhost:%3c%3cport%3e%3e/carsellrest/)searchbyrent/{from}/{to}
      1. Return all homes based on the rent range
      2. GETMAPPING to be Used
   7. [http://localhost:<<port>>/rentahomerest/](http://localhost:%3c%3cport%3e%3e/carsellrest/)searchbycarparking
      1. Return all homes with car parking available
      2. GETMAPPING to be Used
   8. [http://localhost:<<port>>/rentahomerest/](http://localhost:%3c%3cport%3e%3e/carsellrest/)searchbybikeparking
      1. Return all homes with bike parking available
      2. GETMAPPING to be Used
7. Use POST MAN to test the application and test the above REST URLs.



Note:

1. Make the POJO with perfectly matching with database table, map the columns
2. Don’t add ddlauto in application. Properties since the table is created manually
3. Do the dependency injection properly between Controller, Service and DAO layers
4. Use the proper annotations on each layer
5. Your application should use hibernate for backend connectivity. Not Spring jdbc.
   1. Use EntityManager instead of jdbctemplate