

DOCUMENTATION

Github link: <https://github.com/vstephne/Employee>

POSTMan API link: <https://www.getpostman.com/collections/8008e8b6b112a47a2556>

ASSUMPTION:

1. Assuming username for every user is unique for a user
2. Assuming minimum annualSalary is 12000\$ and set it as default in case not entered
3. A user can have different address but their names must be unique for a given user

SYSTEM DESIGN:

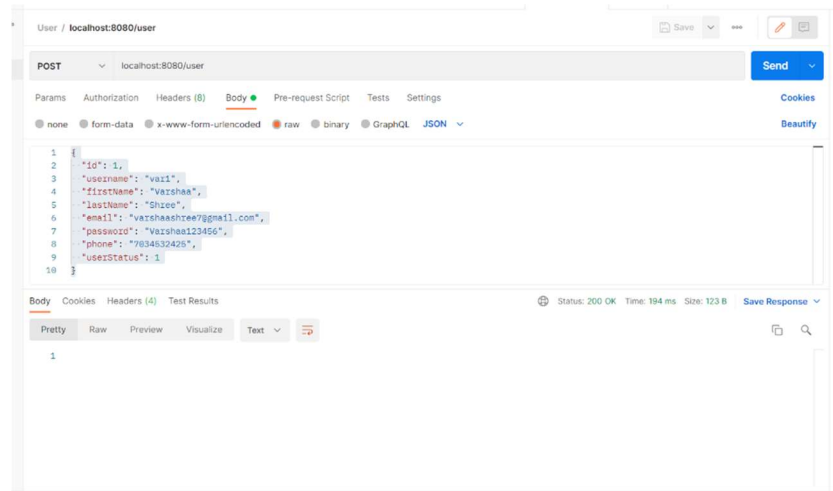
1. DATABASE

- a. The database name is USER_DETAILS
- b. The Database has two tables
 - i. USER_PROFILE_INFO which contains all the information about Users
 - ii. The data also includes username and password this could be put into a separate table as they are for a specific purpose login and logout but as they are singular for a given user at a given time I have added to same USER_PROFILE_INFO table
 - iii. The next table is USER_ADDRESS it has addresses of users as a user can have multiple addresses but their names must be unique.
 - iv. The combination of both username and address is made a primary key.

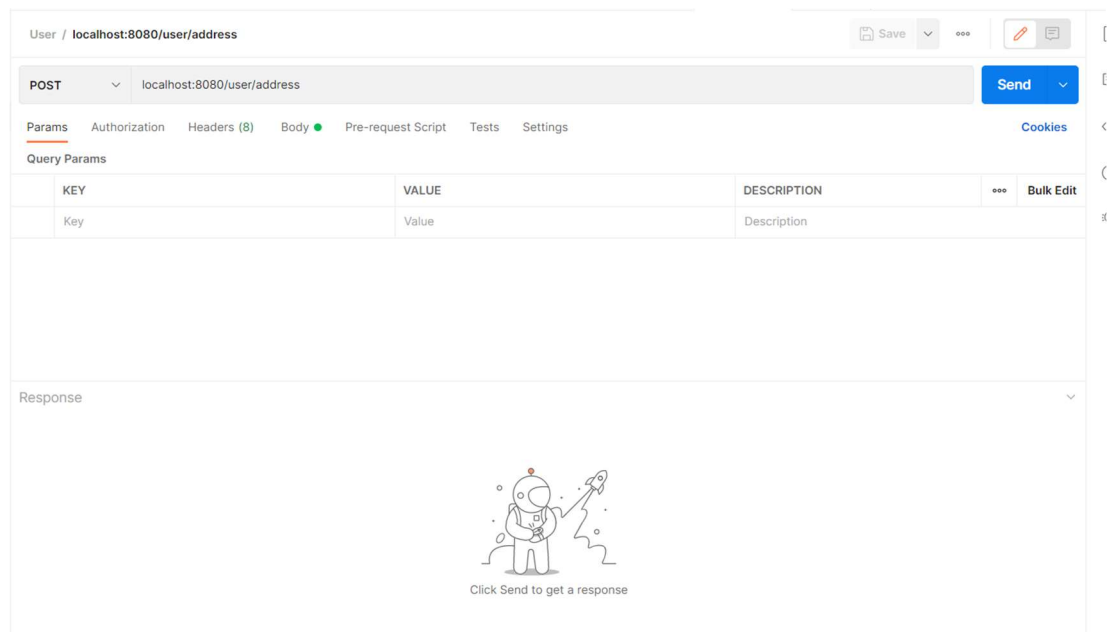
2. SPRING BOOT APPLICATION:

- a. I have divided the code into multiple sections
 - i. Entity: This package mainly has information about the table the columns their type and their getter-setter methods.
 - ii. Dao: This is the data layer. It has functions that fetch data or perform any SQL related function.
 - iii. Service: It is one layer above dao layer here we call the functions in Dao layer and if there are any extra functionalities we add it here
 - iv. Controller: It is the entry point for any API the mappings of different API to their respective functions are done at this layer.
- b. These are some of the API's and their outputs:

i. localhost:8080/user



ii. localhost:8080/user/address



iii. localhost:8080/user/createWithArray

User / localhost:8080/user/createWithArray


POST localhost:8080/user/createWithArray

Params Authorization Headers (8) Body Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary GraphQL JSON

```
1 {
2   "id": 3,
3   "username": "bhuv",
4   "firstName": "Bhuvanendar",
5   "lastName": "Sivashankar",
6   "email": "boonbemlxine@gmail.com",
7   "password": "bhuvbhuv",
8   "phone": "7896798989",
9   "userStatus": 2
10 }
11
```

Response



Click Send to get a response

iv. localhost:8080/user/createWithList

User / localhost:8080/user/createWithArray Copy


POST localhost:8080/user/createWithList

Params Authorization Headers (8) Body Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary GraphQL JSON

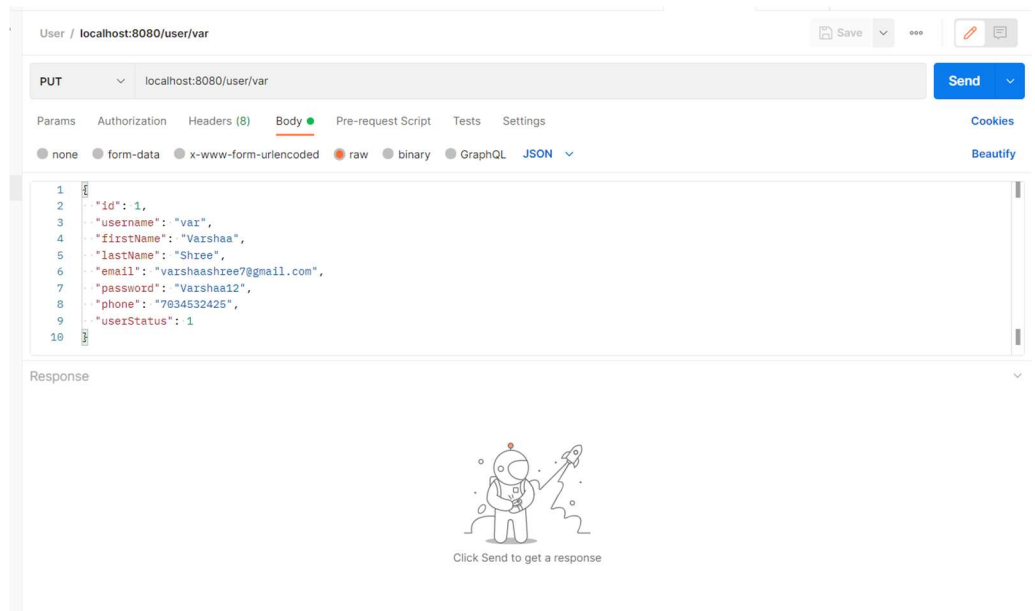
```
1 {
2   "id": 1,
3   "username": "var",
4   "firstName": "Vaishaa",
5   "lastName": "Shree",
6   "email": "varshaashree7@gmail.com",
7   "password": "Varshaa12",
8   "phone": "7834532425",
9   "userStatus": 1
10 }
11
```

Response

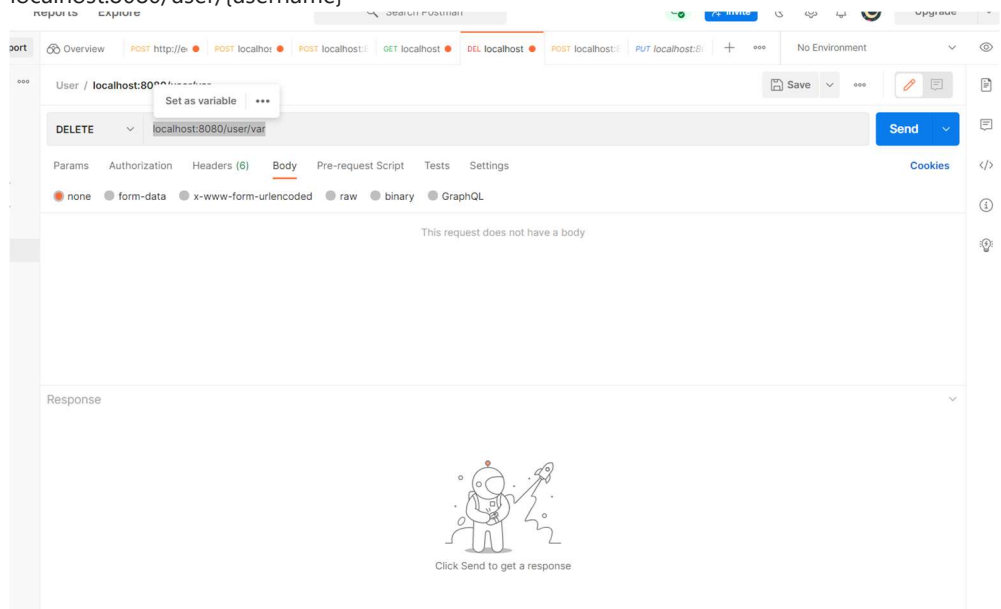


Click Send to get a response

v. localhost:8080/user/{username}



vi. `localhost:8080/user/{username}`



This is POSTMAN Collection with all the API:

<https://www.getpostman.com/collections/8008e8b6b112a47a2556>

STEPS to run the application:

1. Install MYSQL in the system
2. Open the .sql file attached in email or found on repository under SQL folder
3. Clone the repository
4. Run it as a springboot application
5. Run various API dfound on post man which can be accessed by above link

SYSTEM DESIGN QUESTION:

1. If we want to add extra information to address table like walkScore, TransitScore, crime rate for neighborhood we can make these into a separate table. Requirement being if we have two address in same location factors like crime rate and TransitScore will repeat which is bad. In order to normalize we can put them into a different table with the foreign key of the table referencing address table primary key.
2. GPS coordinates for a given address is always same so we can add it to address table directly.
3. From the side of API to get GPS coordinates it is a direct SELECT query based on username and address name on the address table.
4. For TransitScore, walkScore and crime rate we must do SELECT query based on username and address name on the new extra table which would be created.

Things which could not be completed:

Error handling has not been done due to lack of time.