**MUSELY APPLICATION DEMO**

**Github link for code:**

[**https://github.com/aakash0456/Musely-demo-Application**](https://github.com/aakash0456/Musely-demo-Application)

Requirement:

- Support persistence layer

- Updating UserService

All the business I wrote in UserService layer

- Store User into database of your choice

I have used MySQL DB

- You may add your preferred persistence layer framework into the project

I have added layers according to the requirement. Some are userDao layer, layer for exception handling. Please check github link or zip file.

- Add another API

- To support creating multiple users in one call.

I have created API for which can accept multiple user and save in the database.

- Modify existing API

- To support exception handling

- Such as not able to find such a user with “Get User”

All the exceptions in for every call have handled.

- If you have more time, what else would you add/improve?

There is many things I could have added .

* Exception handling message – I haven’t added proper the messages. I could have added custom messages according to the exception error.
* Regarding the dependency documentation. Most of the part I struggled with compatibility of dependencies. I really wanted to document it, I tried many times but somehow I got it resolved using stack overflow. It was the area I needed to document it for future use.

**Jetty Server Successfully Started…..**

**To RUN project please change the Database, I have used mysql DB, Please change the dependency in pom.xml and persistence.xml according to the database. Below I have all the results for all the API calls.**

**A screenshot of a computer

Description automatically generated with medium confidence**

**A picture containing text, screenshot, software, multimedia software

Description automatically generated**

**All PostMan call**

**Get call - To get 1 Entity.**

**A screenshot of a computer

Description automatically generated with medium confidence**

**POST – To Save 1 Entity**

**A screenshot of a computer

Description automatically generated with medium confidence**

**GET ALL - To Get All Entity**

**A screenshot of a computer

Description automatically generated with medium confidence**

**POST ALL – To Insert All Entity**

**A screenshot of a chat

Description automatically generated with medium confidence**

**DELETE – To delete 1 Entity**

**A screenshot of a computer

Description automatically generated with medium confidence**

**PUT - To update existing Entity**

**A screenshot of a computer

Description automatically generated with medium confidence**

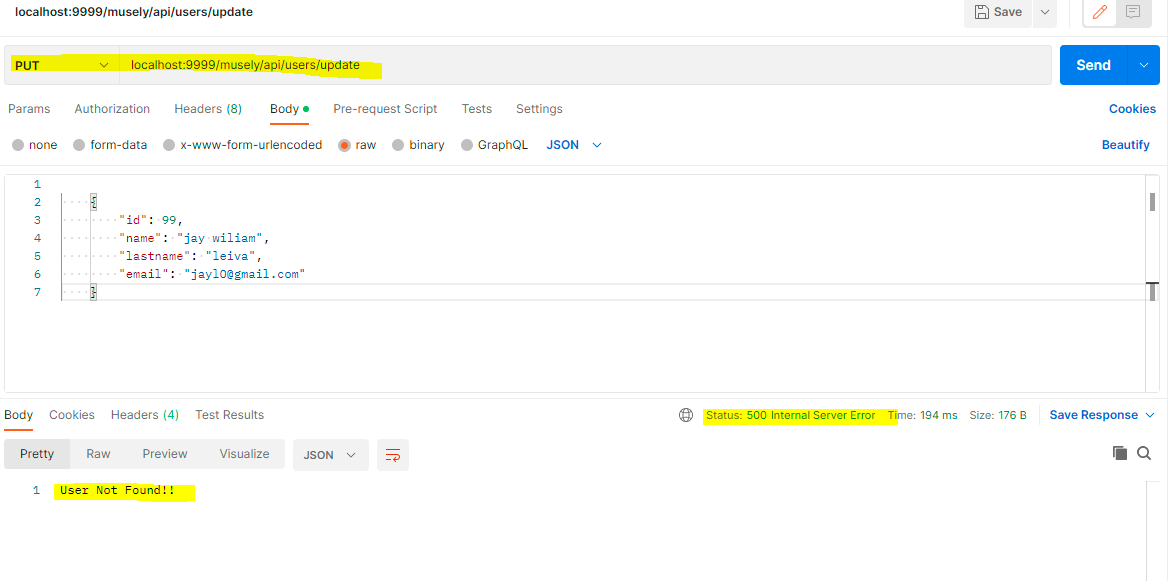
**Exception Handling**

**If user not found in Database**

**A screenshot of a computer

Description automatically generated with medium confidence**

**If User not in database and update wrong user/Entity**

****

**DELETE – if Delete wrong user.**

A screenshot of a computer

Description automatically generated with medium confidence