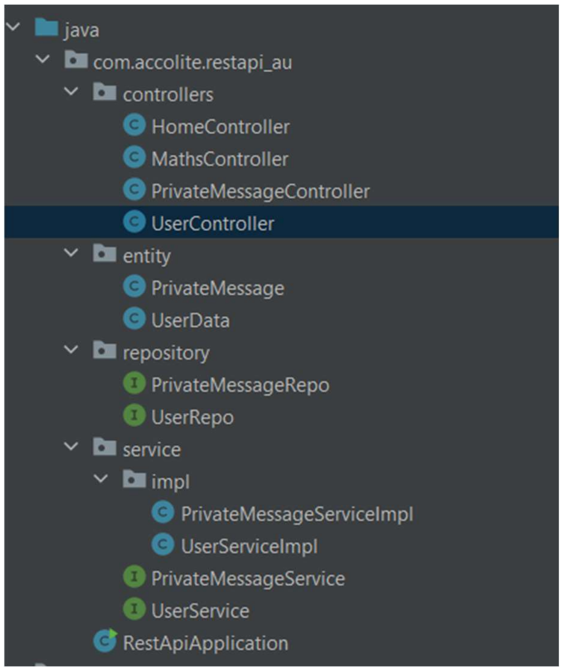
Spring REST Services Assignment

By Raj Vignesh Karunakaran

1. Created a Spring boot application with REST services. The application is a simple webapp that allows user registration and send messages.

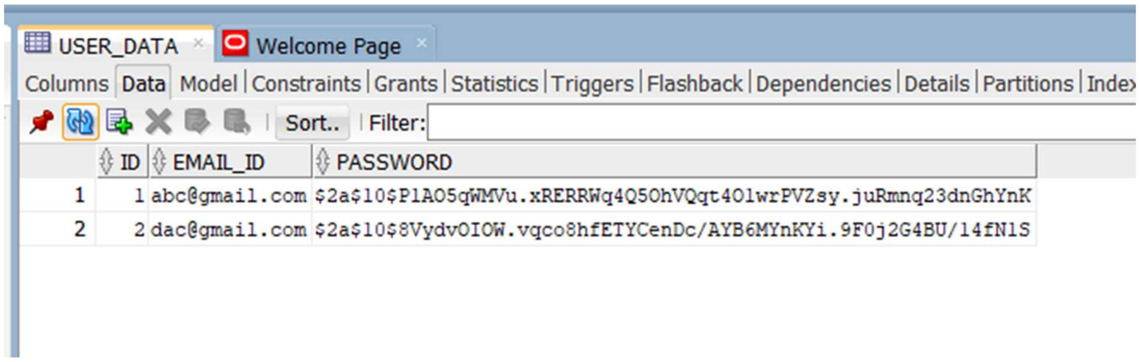
   Check GitHub for Sources codes



```
java
  com.accolite.restapi_au
    controllers
      HomeController
      MathsController
      PrivateMessageController
      UserController
    entity
      PrivateMessage
      UserData
    repository
      PrivateMessageRepo
      UserRepo
    service
      impl
        PrivateMessageServiceImpl
        UserServiceImpl
      PrivateMessageService
      UserService
    RestApiApplication
```

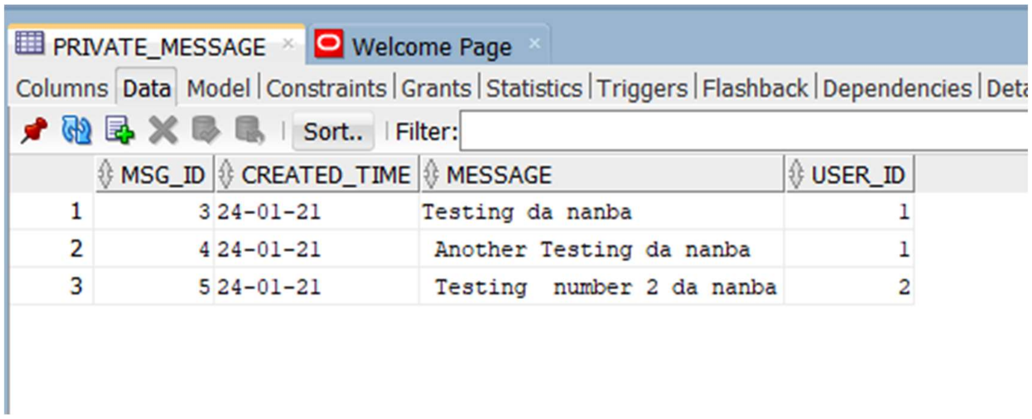
2. Used Oracle DB to store all the user data and private messages.

   a. UserDataDB



	ID	EMAIL_ID	PASSWORD
1	1	abc@gmail.com	$2a$10$P1AO5qWMVu.xRERRWq4Q5OhVQqt4O1wrPVZsy.juRmnq23dnGhYnK
2	2	dac@gmail.com	$2a$10$8VydvOIOW.vqco8hfETYCenDc/AYB6MYnKYi.9F0j2G4BU/14fN1S

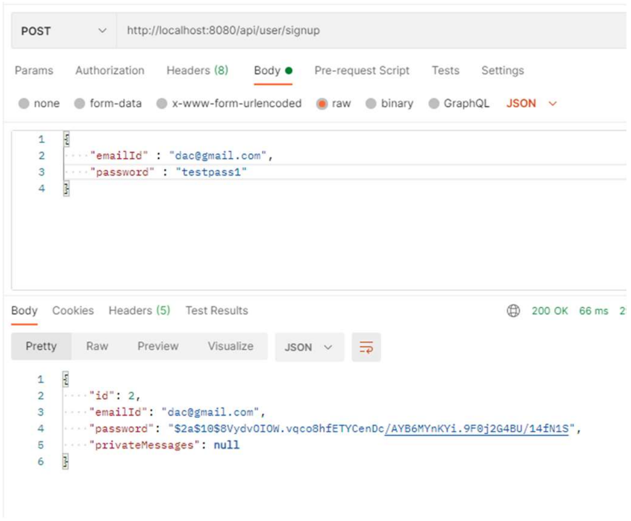
b. PrivateMessageDB


PRIVATE_MESSAGE
Welcome Page
Columns | Data | Model | Constraints | Grants | Statistics | Triggers | Flashback | Dependencies
Sort..
Filter:

	MSG_ID	CREATED_TIME	MESSAGE	USER_ID
1	3	24-01-21	Testing da nanba	1
2	4	24-01-21	Another Testing da nanba	1
3	5	24-01-21	Testing number 2 da nanba	2

3. Created REST API services to demonstrate POST, GET, PUT, DELETE operations

a. POST for registering User



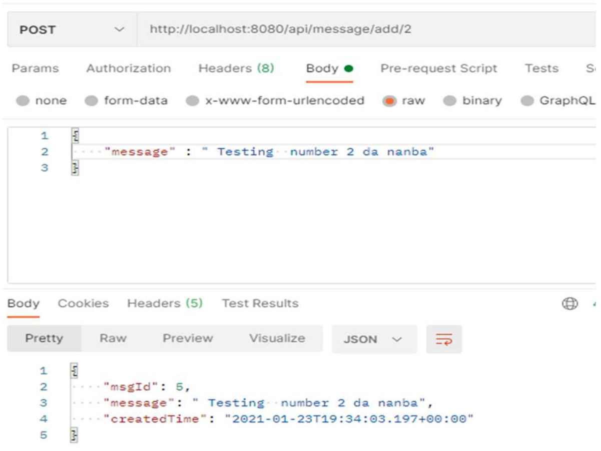
```
POST  http://localhost:8080/api/user/signup
```

```
{
    "emailId" : "dac@gmail.com",
    "password" : "testpass1"
}
```

```
200 OK  66 ms
```

```
{
    "id": 2,
    "emailId": "dac@gmail.com",
    "password": "$2a$10$8VydvOIOW.vqco8hfETYCenDc/AYB6MYnKYi.9F0j2G4BU/14fN1S",
    "privateMessages": null
}
```

b. POST for adding messages to user

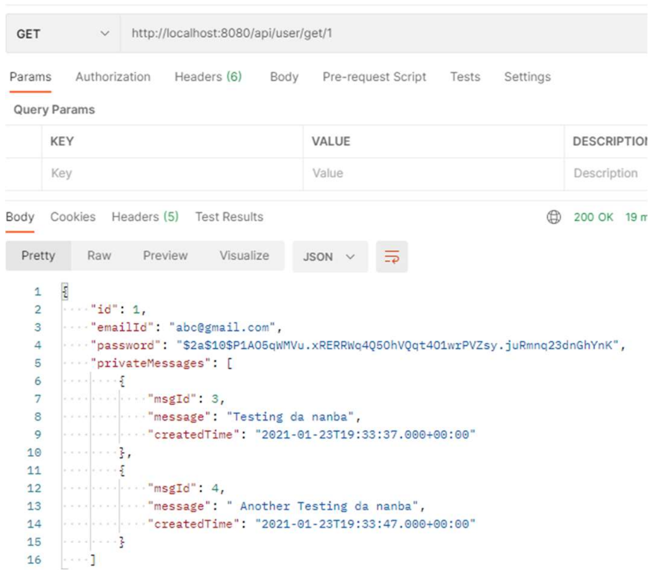


```
POST  http://localhost:8080/api/message/add/2
```

```
{
    "message" : " Testing  number 2 da nanba"
}
```

```
{
    "msgId": 5,
    "message": " Testing  number 2 da nanba",
    "createdTime": "2021-01-23T19:34:03.197+00:00"
}
```

c. GET to retrieve user data along with his messages



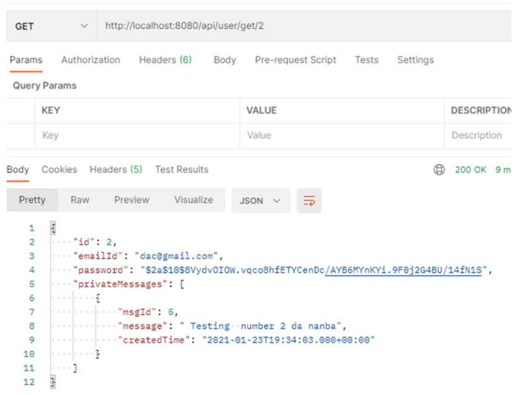
GET http://localhost:8080/api/user/get/1

200 OK 19 m

```
{
    "id": 1,
    "emailId": "abc@gmail.com",
    "password": "$2a$10$P1A05qWMVu.xRERRWq4Q50hVQqt401wrPVZsy.juRmnq23dnGhYnK",
    "privateMessages": [
        {
            "msgId": 3,
            "message": "Testing da nanba",
            "createdTime": "2021-01-23T19:33:37.000+00:00"
        },
        {
            "msgId": 4,
            "message": " Another Testing da nanba",
            "createdTime": "2021-01-23T19:33:47.000+00:00"
        }
    ]
```

d. PUT to update user details (Can be updated only if the provided password in the header matches with existing password)

Before PUT:

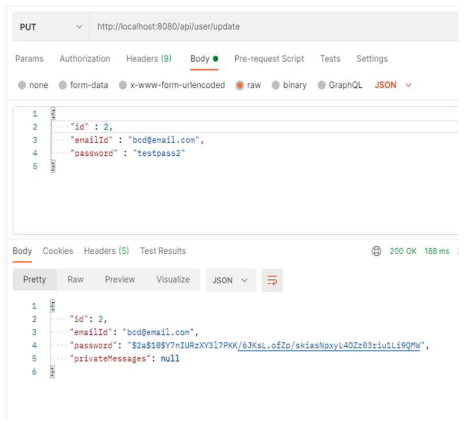


GET http://localhost:8080/api/user/get/2

200 OK 9 m

```
{
    "id": 2,
    "emailId": "dac@gmail.com",
    "password": "$2a$10$8VydvOIOW.vqco8hfETYCenDc/AYB6MYnKYi.9F0j2G4BU/14fN1S",
    "privateMessages": [
        {
            "msgId": 5,
            "message": " Testing  number 2 da nanba",
            "createdTime": "2021-01-23T19:34:03.000+00:00"
        }
    ]
}
```

PUT:



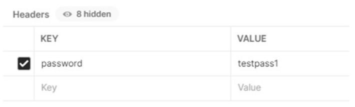
PUT http://localhost:8080/api/user/update

Body: ● raw JSON

```
{
    "id" : 2,
    "emailId" : "bcd@email.com",
    "password" : "testpass2"
}
```

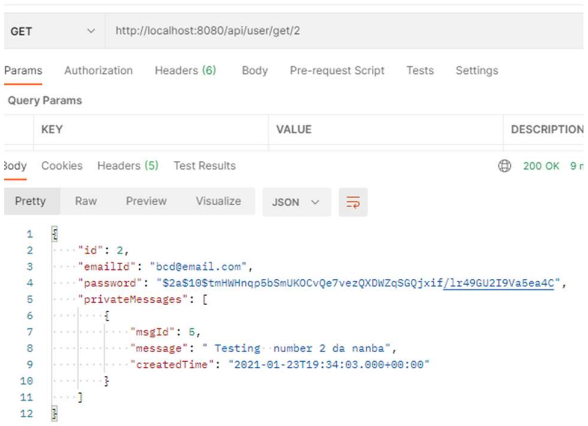
200 OK 188 ms

```
{
    "id": 2,
    "emailId": "bcd@email.com",
    "password": "$2a$10$Y7nIURzXY3l7PKK/6JKsL.ofZp/skiasNpxyL4OZz03riu1Li9QMW",
    "privateMessages": null
}
```


Headers 8 hidden

	KEY	VALUE
✓	password	testpass1
	Key	Value

After PUT:


GET http://localhost:8080/api/user/get/2

Params | Authorization | Headers (6) | Body | Pre-request Script | Tests | Settings

Query Params

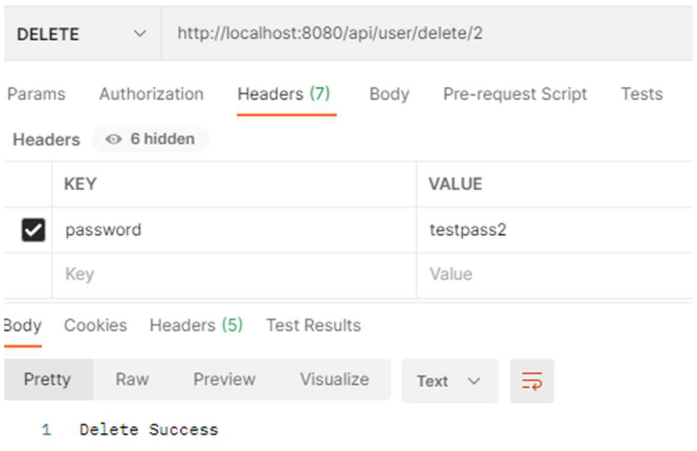
KEY	VALUE	DESCRIPTION

Body | Cookies | Headers (5) | Test Results — 200 OK

Pretty | Raw | Preview | Visualize | JSON

```
{
    "id": 2,
    "emailId": "bcd@email.com",
    "password": "$2a$10$tmHWHnqp5bSmUKOCvQe7vezQXDWZqSGQjxif/lr49GU2I9Va5ea4C",
    "privateMessages": [
        {
            "msgId": 5,
            "message": " Testing  number 2 da nanba",
            "createdTime": "2021-01-23T19:34:03.000+00:00"
        }
    ]
}
```

e. DELETE to delete an user (Can be deleted only if the provided password in the header matches with existing password)


DELETE http://localhost:8080/api/user/delete/2

Params | Authorization | Headers (7) | Body | Pre-request Script | Tests

Headers 6 hidden

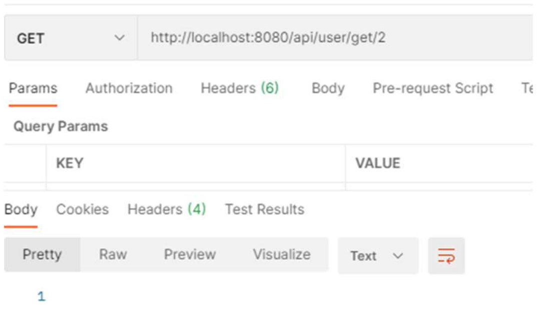
	KEY	VALUE
✓	password	testpass2
	Key	Value

Body | Cookies | Headers (5) | Test Results

Pretty | Raw | Preview | Visualize | Text

```
Delete Success
```

After DELETE:


GET http://localhost:8080/api/user/get/2

Params | Authorization | Headers (6) | Body | Pre-request Script

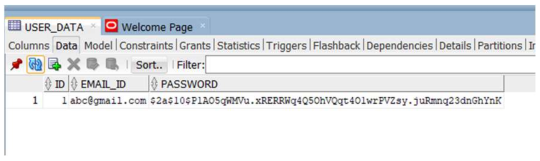
Query Params

KEY	VALUE

Body | Cookies | Headers (4) | Test Results

Pretty | Raw | Preview | Visualize | Text

```
1
```


USER_DATA | Welcome Page

Columns | Data | Model | Constraints | Grants | Statistics | Triggers | Flashback | Dependencies | Details | Partitions

Sort.. Filter:

	ID	EMAIL_ID	PASSWORD
1	1	abc@gmail.com	$2a$10$P1AO5qWMVu.xRERRWq4Q5OhVQqt4OlwrPVZsy.juRmnq23dnGhYnK