

Screenshots of the Results

I. Screenshots of the APIs:

1. POST Method:

VS Code Thunder Client API Testing:

The screenshot shows the VS Code Thunder Client interface. The request is a POST to `http://localhost:8080/api/v1/users`. The response is a JSON object with the following details:

```
{  "userId": 52,  "userName": "Gopal",  "userPhoneNumber": "132456789",  "userEmailAddress": "gopal@gmail.com"}
```

Console Log:

```
2024-02-17T19:03:51.270+05:30 WARN 14872 --- [nio-8080-exec-1] .w.s.m.s.DefaultHandlerExceptionHandlerResolver : Resolved [org.springframework
2024-02-17T19:03:55.891+05:30 INFO 14872 --- [nio-8080-exec-3] c.e.0.services.UserServiceImpl :
Log: User Details Saved!
```

H2 Console:

The screenshot shows the H2 Console interface. The SQL statement `Select * from user_table;` has been executed, resulting in the following table:

USER_ID	USER_EMAIL_ADDRESS	USER_NAME	USER_PHONE_NUMBER
1	rohaan@gmail.com	Roha	123456789
2	midhun@gmail.com	Midhun	123456798
52	gopal@gmail.com	Gopal	132456789

(3 rows, 2 ms)

2. GET Method:

VS Code Thunder Client API Testing:

The screenshot shows the VS Code Thunder Client interface. The request is a GET method to the URL `http://localhost:8080/api/v1/users/52`. The response status is 200 OK, with a size of 99 Bytes and a time of 6 ms. The response body is a JSON object containing user details for user ID 52.

```
1 {
2   "userId": 52,
3   "userName": "Gopal",
4   "userPhoneNumber": "132456789",
5   "userEmailAddress": "gopal@gmail.com"
6 }
```

Console Log:

The console log shows a timestamped INFO message from the application and a log message indicating that user details were displayed.

```
2024-02-17T19:15:07.274+05:30 INFO 14872 --- [nio-8080-exec-4] c.e.o.services.UserServiceImpl

Log: User Details Displayed!
```

H2 Console:

The screenshot shows the H2 Console interface. The SQL statement `Select * from user_table;` has been executed, resulting in a table with 3 rows and 4 columns: USER_ID, USER_EMAIL_ADDRESS, USER_NAME, and USER_PHONE_NUMBER.

USER_ID	USER_EMAIL_ADDRESS	USER_NAME	USER_PHONE_NUMBER
1	rohaan@gmail.com	Rohaam	123456789
2	midhun@gmail.com	Midhun	123456798
52	gopal@gmail.com	Gopal	132456789

(3 rows, 2 ms)

3. PUT Method:

VS Code Thunder Client API Testing:

PUT ⌵ http://localhost:8080/api/v1/users/52 Send

Query Headers² Auth Body¹ Tests Pre Run

JSON XML Text Form Form-encode GraphQL Binary

JSON Content Format

```
1 {
2   "userName": "Gopal",
3   "userPhoneNumber": "987654321",
4   "userEmailAddress": "gopal@gmail.com"
5 }
```

Status: 200 OK Size: 99 Bytes Time: 21 ms

Response Headers⁴ Cookies Results Docs {} ≡

```
1 {
2   "userId": 52,
3   "userName": "Gopal",
4   "userPhoneNumber": "987654321",
5   "userEmailAddress": "gopal@gmail.com"
6 }
```

Copy

Console Log:

```
2024-02-17T19:18:34.406+05:30 INFO 14872 --- [nio-8080-exec-6] c.e.0.services.UserServiceImpl
Log: User Details Updated!
```

H2 Console:

Run Run Selected Auto complete Clear SQL statement:

Select * from user_table;

Select * from user_table;

USER_ID	USER_EMAIL_ADDRESS	USER_NAME	USER_PHONE_NUMBER
1	rohaan@gmail.com	Rohaam	123456789
2	midhun@gmail.com	Midhun	123456798
52	gopal@gmail.com	Gopal	987654321

(3 rows, 0 ms)

4. DELETE Method:

VS Code Thunder Client API Testing:

DELETE

http://localhost:8080/api/v1/users/52

Send

Query

Headers²

Auth

Body

Tests

Pre Run

JSON

XML

Text

Form

Form-encode

GraphQL

Binary

JSON Content

Format

1

Status: 200 OK

Size: 0 Bytes

Time: 19 ms

Response

Headers³

Cookies

Results

Docs

{}

≡

1

Copy

Console Log:

```
2024-02-17T19:21:13.600+05:30 INFO 14872 --- [io-8080-exec-10] c.e.o.services.UserServiceImpl

Log: User Details Deleted!
```

H2 Console:

Run

Run Selected

Auto complete

Clear

SQL statement:

Select * from user_table;

Select * from user_table;

USER_ID	USER_EMAIL_ADDRESS	USER_NAME	USER_PHONE_NUMBER
1	rohaan@gmail.com	Rohaam	123456789
2	midhun@gmail.com	Midhun	123456798

(2 rows, 0 ms)

II. Screenshots of the Test Runs:

1. POST Method:

Test Code :

```
@Test
public void testCreateUser() throws Exception {
    mockMvc.perform(post( urlTemplate: "http://localhost:8080/api/v1/users")
        .contentType(MediaType.APPLICATION_JSON)
        .content("{ \"userName\": \"John\", \"userPhoneNumber\": \"8976\" }")
        .andExpect(status().isOk()));
}
```

Console Log:

```
✓ UserControllerTes 755 ms ✓ Tests passed: 4 of 4 tests - 755 ms
✓ testUpdateUser 718 ms 2024-02-18T16:59:42.254+05:30 INFO 12088 --- [main] c.e.0.services.UserServiceImpl
✓ testCreateUser 22 ms
✓ testDeleteUser 9 ms
✓ testDisplayUser 6 ms
Log: User Details Saved!

Hibernate:
select
next value for user_table_seq
```

2. GET Method:

Test Code:

```
@Test
public void testDisplayUser() throws Exception {
    mockMvc.perform(get( urlTemplate: "http://localhost:8080/api/v1/users/1")
        .contentType(MediaType.APPLICATION_JSON)
        .content(""))
        .andExpect(status().isOk());
}
```

Console Log:

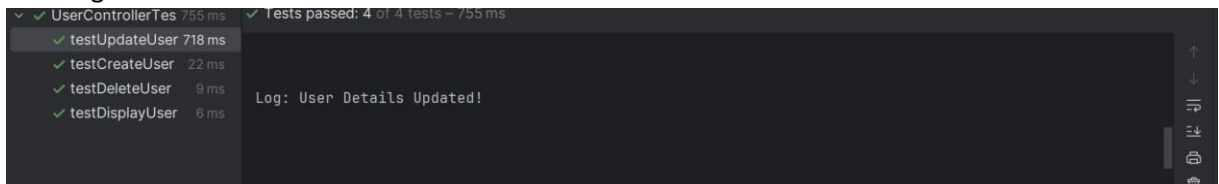
```
✓ UserControllerTes 755 ms ✓ Tests passed: 4 of 4 tests - 755 ms
✓ testUpdateUser 718 ms 2024-02-18T16:59:42.288+05:30 INFO 12088 --- [main] c.e.0.services.UserServiceImpl
✓ testCreateUser 22 ms
✓ testDeleteUser 9 ms
✓ testDisplayUser 6 ms
Log: User Details Displayed!
```

3. PUT Method:

Test Code:

```
@Test
public void testUpdateUser() throws Exception {
    mockMvc.perform(put(urlTemplate: "http://localhost:8080/api/v1/users/2")
        .contentType(MediaType.APPLICATION_JSON)
        .content("{\"userName\": \"Tom\", \"userPhoneNumber\": \"0897\"}"))
        .andExpect(status().isOk());
}
```

Console Log:



4. DELETE Method:

Test Code:

```
@Test
public void testDeleteUser() throws Exception {
    mockMvc.perform(delete(urlTemplate: "http://localhost:8080/api/v1/users/152")
        .contentType(MediaType.APPLICATION_JSON)
        .content(""))
        .andExpect(status().isOk());
}
```

Console Log:

