**1. Get All Users**

* **Request Type**: GET
* **Endpoint**: http://localhost:8081/users
* **Description**: Retrieve a list of all users.

**Expected Response**:

* **Status**: 200 OK
* **Response Body (example)**:

json

[

{

"id": 1,

"username": "user1",

"email": "user1@example.com"

},

{

"id": 2,

"username": "user2",

"email": "user2@example.com"

}

]

**2. Get User by ID**

* **Request Type**: GET
* **Endpoint**: http://localhost:8081/users/{id}
* **Description**: Retrieve details of a specific user by ID.
* **Example ID**: 1

**Expected Response**:

* **Status**: 200 OK
* **Response Body (example)**:

json

{

"id": 1,

"username": "user1",

"email": "user1@example.com"

}

**3. Insert User into Database**

* **Request Type**: POST
* **Endpoint**: http://localhost:8081/users
* **Description**: Insert a new user into the database.
* **Request Body**:

json

{

"username": "user",

"email": "user@example.com",

"password": "password123"

}

**Expected Response**:

* **Status**: 200 Ok
* **Response Body (example)**:

json

{

"id": 1,

"username": "user",

"email": "user@example.com"

}

**4. Update Existing User**

* **Request Type**: PUT
* **Endpoint**: http://localhost:8081/users/{id}
* **Description**: Update the details of an existing user.
* **Example ID**: 1
* **Request Body**:

json

{

"username": "updateduser",

"email": "updateduser@example.com",

"password": "updatedpassword123"

}

**Expected Response**:

* **Status**: 200 OK
* **Response Body (example)**:

json

{

"id": 1,

"username": "updateduser",

"email": "updateduser@example.com"

}

**5. Delete Existing User**

* **Request Type**: DELETE
* **Endpoint**: http://localhost:8081/users/{id}
* **Description**: Delete a user from the database.
* **Example ID**: 1

**Expected Response**:

* **Status**: 204 No Content
* **No response body**

**Negative Test Cases**

**6. Get User by Invalid ID**

* **Request Type:** GET
* **Endpoint:** http://localhost:8081/users/invalid-id
* **Test Case:**
  + **Description:** Attempt to retrieve a user with an invalid or non-existent ID.
  + **Example ID:** invalid-id
  + **Expected Response:**
    - **Status:** 404 Not Found
    - **Response Body (example):**

json

{

    "timestamp": "2024-09-12T12:35:26.226+00:00",

    "status": 404,

    "error": "Not Found",

    "trace": "com.ecommerce.usermicroservice.exception.ResourceNotFoundException: User not found with id: 6767 java.base/java.lang.Thread.run(Thread.java:1583)\r\n",

    "message": "User not found with id: 6767",

    "path": "/users/6767"

}

**7. Delete User with Invalid ID**

* **Request Type:** DELETE
* **Endpoint:** http://localhost:8081/users/invalid-id
* **Test Case:**
  + **Description:** Attempt to delete a user with an invalid or non-existent ID.
  + **Example ID:** invalid-id
  + **Expected Response:**
    - **Status:** 404 Not Found
    - **Response Body (example):**

json

{

    "timestamp": "2024-09-12T12:36:27.210+00:00",

    "status": 404,

    "error": "Not Found",

    "trace": "com.ecommerce.usermicroservice.exception.ResourceNotFoundException: User not found with id: 6767\r\n\tat java.base/java.lang.Thread.run(Thread.java:1583)\r\n",

    "message": "User not found with id: 6767",

    "path": "/users/6767"

}