### **APIs in UserServiceApp**

### **1. Register User**

* **Endpoint:** /api/user/register
* **HTTP Method:** POST
* **Description:** Registers a new user.
* **Authorization Level:** Public (No authentication required).
* **Request Body:**
  + JSON object representing the user details to be registered (e.g., username, email, password, etc.).
* **Response:** The newly registered User object.

**Example Request**  
POST /api/user/register

{

"username": "johnDoe",

"role": "ADMIN",

"password": "securePassword"

}

**Example Response:**  
{

"id": 1,

"username": "johnDoe",

"role": "ADMIN",

"password": "securePassword"

}

### **2. Get User by ID**

* **Endpoint:** /api/user/{id}
* **HTTP Method:** GET
* **Description:** Retrieves the details of a specific user by their ID.
* **Authorization Level:** Requires ADMIN or USER authority.
* **Path Parameter:**
  + id (Long) - The ID of the user to be retrieved.
* **Response:** The User object for the given ID.

**Example Request**  
GET /api/user/1

**Example Response:**  
{

"id": 1,

"username": "johnDoe",

"role": "ADMIN",

"password": "securePassword"

}

### **3. Get All Users**

* **Endpoint:** /api/user/getAllUser
* **HTTP Method:** GET
* **Description:** Retrieves all registered users.
* **Authorization Level:** Requires ADMIN authority.
* **Response:** A list of all User objects.

**Example Request:**  
GET /api/user/getAllUser

**Example Response:**json

When data not found

{

"responseMessage": "No Data Found",

"responseCode": "-1",

"responseObject": []

}

When successful data fetched

{

"responseMessage": "Data fetched successfuly",

"responseCode": "1",

"responseObject": [

{

"id": 1,

"username": "Jhon Wick",

"password": "$2a$10$8B60A02A.aY.a1t4yyeAi.ALGirBAYMdoL9zV/AYzOjz/nzvQhtQ.",

"role": "ADMIN"

}

]

}

### **4. Update User**

* **Endpoint:** /api/user/{id}
* **HTTP Method:** PUT
* **Description:** Updates an existing user's details.
* **Authorization Level:** Requires ADMIN or USER authority.
* **Path Parameter:**
  + id (Long) - The ID of the user to be updated.
* **Request Body:**
  + JSON object containing the updated user details.
* **Response:** The updated User object.

**Example Request:**  
PUT /api/user/1

{

"id": 1,

"username": "johnUpdated",

"role": "ADMIN",

"password": "securePassword"

}

**Example Response:**  
{

"id": 1,

"username": "johnUpdated",

"role": "ADMIN",

"password": "securePassword"

}

### **5. Delete User**

* **Endpoint:** /api/user/{id}
* **HTTP Method:** DELETE
* **Description:** Deletes a user by their ID.
* **Authorization Level:** Requires ADMIN or USER authority.
* **Path Parameter:**
  + id (Long) - The ID of the user to be deleted.

**Example Request:**  
/api/user/1