Users Management System - Documentation

Table of Contents

---

1. Overview

The **Users Management System** is a RESTful web service for managing user accounts, designed with Jakarta EE and deployed on a WildFly server. It supports user creation, updates, deletion, role management, and password resets, with a strong emphasis on clean architecture, exception handling, and scalability.

---

2. Technology Stack

- **Framework:** Jakarta EE
- **Server:** WildFly
- **Language:** Java
- **Dependencies:**
  - Jakarta EE API
  - JAX-RS API
  - Jakarta Persistence (JPA)
  - H2 Database
  - Jakarta CDI (Context and Dependency Injection)
  - JUnit Jupiter API
  - Mockito Core
  - Mockito JUnit Jupiter
  - Jakarta Commons Logging

---

3. API Endpoints

Below is a concise table of available endpoints:

Note: During user creation, the id field should not be included as it is generated automatically.

Method	Endpoint	Description
GET	/api/v1.0/users/{page}/{size}	Fetch all users (paginated)
GET	/api/v1.0/users/{id}	Get user by ID
POST	/api/v1.0/users	Create a new user
PUT	/api/v1.0/users/update	Update an existing user
DELETE	/api/v1.0/users/{id}	Delete a user by ID
PUT	/api/v1.0/users/{id}/ reset-password	Reset a user’s password
PUT	/api/v1.0/users/ update-role	Update user role

---

4. Sample Payloads

Sample Create User Payload

```
{
    "firstname": "David",
    "lastname": "Gati",
    "email": "gatidavid2012@gmail.com",
    "birthday": "1990-01-01",
    "password": "password123",
    "phoneNumber": "1234567890",
    "address": "123 Main St Michel Camp",
    "username": "dgati",
    "profilePictureURL": "http://aws.com/image.jpg",
    "role": "USER",
    "status": "ACTIVE"
}
```

Sample Update User Payload

```
{
    "address": "123 Main St Michel Camp",
    "createdOn": "2024-11-29T11:52:44.504833",
    "email": "gatidavid2012@gmail.com",
    "firstname": "David Updated",
    "id": 1,
    "lastname": "Gati",
    "phoneNumber": "1234567890",
    "profilePictureURL": "http://aws.com/image.jpg",
    "role": "USER",
    "status": "ACTIVE",
    "updatedOn": "2024-11-29T11:52:44.504833",
    "username": "dgati"
}
```

Sample Change Password Payload

```
{
    "id": 1,
    "oldPassword": "password123",
    "newPassword": "password@123"
}
```

Update Role Payload

```
{
    "id": 1,
    "userRole": "ADMIN"

}
```

5. Recommendation for Improvement

1. **Password Security**: Encrypt passwords using secure hashing (e.g., BCrypt). Avoid storing plain text passwords.
2. **Future Enhancements**: Introduce soft deletes to preserve historical records

6. How to Run the Project

1. **Prerequisites:**
   - Install and configure WildFly server (compatible with Jakarta EE 9+).
   - Ensure Maven is installed for project builds.
2. **Steps to Deploy:**
   - Build the project using Maven:

     ```
     mvn clean install
     ```

   - Copy the generated .war file to WildFly's deployment directory.

Conclusion

I am confident that the technical skills and problem-solving approach I have applied to this project make me well-suited for any future challenges. I look forward to discussing how my experience can contribute to your team's success.
