

User Management API – Quick Guide

Download and compile source code

1. Open application Git Bash.
2. Navigate to the folder where you want to save the source code.
3. Run the command "git clone <https://github.com/Joakiin/users-amangement-api.git>".
4. Open the cloned source code using your preferred code editor; the application was developed using Visual Studio Code.
5. The application can be compile using Maven with the command "mvn compile" or, in the scenario where you want to edit the code, use the command "mvn clean compile" after saving the change.
6. To run the application, use the command "mvn spring-boot:run".
7. For very quick tests, it is recommended to use Swagger to run the endpoints using the following URL: <http://localhost:7020/api/swagger-ui/index.html>

Run with Docker

1. Build the JAR with command "mvn clean package -DskipTests".
2. Create Docker image with command "docker build -t users-api".
3. Run with command "docker run -p 7020:7020 users-api".

Import Postman Collection

1. Download the Postman collection file: Api Users Management.postman_collection.json.
2. Open Postman Desktop application.
3. Click "Import" button (top left).
4. Select the downloaded JSON file.
5. All API endpoints will be automatically configured.

Test Endpoints

1. GET All Users
 - a. Endpoint: <http://localhost:7020/api/users>
 - b. Expected: JSON format with at least 3 pre-configured records.
 - c. Method: GET
2. GET Single User By Id
 - a. Endpoint: <http://localhost:7020/api/users/{id}>
 - b. Expected: JSON format with the record whose ID is being searched.
 - c. Method: GET

- d. Note: The ID is generated randomly, so it is necessary to have previously executed endpoint 1 to obtain an ID to use for testing.
- 3. POST Create New User
 - a. Endpoint: <http://localhost:7020/api/users>

Body example: {

```
"name": "Daniel Padierna Jimenez",
"email": "test@mail.com",
"phone": "5512345678",
"password": "password123",
"taxId": "TEST990101XXX",
"addresses": [
  {
    "name": "Home",
    "street": "Street 123",
    "countryCode": "MX"
  }
]
```

- b. Method: POST
- 4. Patch Update Existing User
 - a. Endpoint: <http://localhost:7020/api/users/{id}>

Body example: {

```
"email": "trejoanaisabel@mail.com",
"name": "Ana",
"phone": "5555887412",
"password": "Password789",
"taxId": "CCRR990101XXX",
"addresses": [
  {
    "id": 5,
    "name": "string",
    "street": "string",
    "countryCode": "string"
  }
]
```

- b. Method: PATCH
- 5. DELETE Users
 - a. <http://localhost:7020/api/users/{id}>
 - b. Method: DELETE
- 6. GET Users with Filter

- a. Endpoint (Postman example):
<http://localhost:7020/api/users?filter=email%2Bco%2Btrejo>
- b. Filter in swagger: parameter+filter(co,eq,ew,sw,)+value
- c. Method: GET