

Author: Wiktor Markowicz

Source: <https://reqres.in>

## Testing API in Postman

I chose three endpoints:

- /api/users
- /api/login
- /api/register

with different HTTP method like GET, POST, PUT, DELETE.

I defined in variables of my collection one variable:

*baseUrl* = <https://reqres.in>

To get a valid response from the server, every request requires an API key in the header:

*Header: "API-Key: reqres-free-v1"*

# Test cases

## Test 1: Get all users

*Method:* GET

*Purpose of the test:* Retrieve the list of all users in the system

*Endpoint:* /api/users (see Figure 1)

*Input:* none

*Expected result:* HTTP Status Code: 200 - Success

*Actual result:* as expected

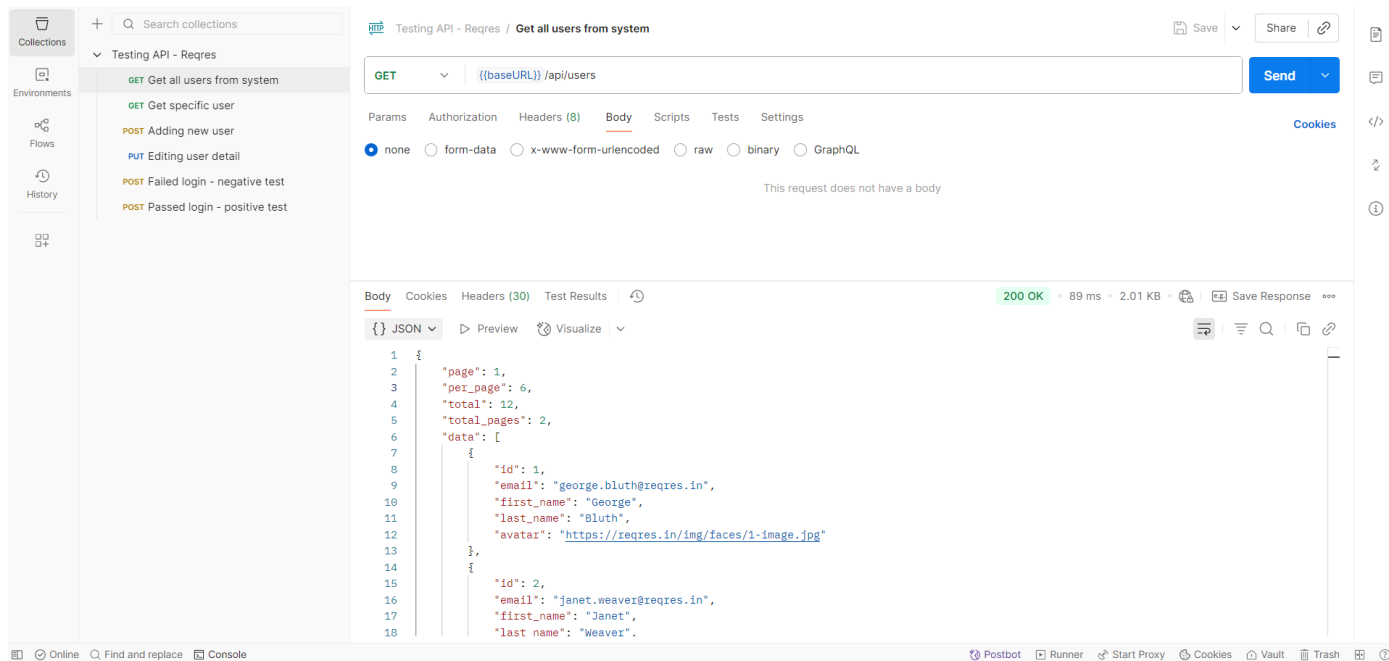


Figure 1 Get all users from the system

*Description:* This endpoint returns a list of users available in the system, including fields such as: id, email, first\_name, last\_name, and avatar.

## Test 2: Get user by ID (positive)

*Method:* GET

*Purpose of the test:* Retrieve information about a specific user by ID

**Endpoint:** /api/users/{id} (see Figure 2)

**Input:** none

**Expected result:** HTTP Status Code: 200 - Success

**Actual result:** as expected

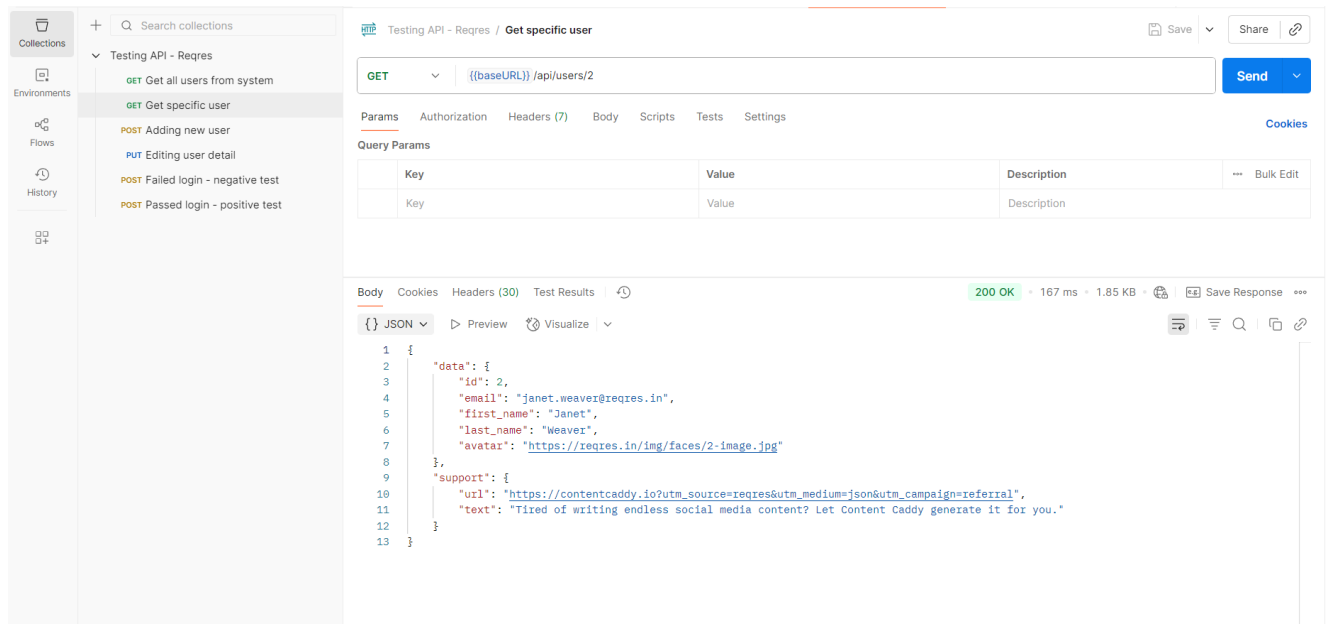


Figure 2 Get particular user

**Description:** It works similar to previous endpoint, but the difference is that the user must provide the ID as a path parameter at the end of the URL. It returns information such as id, email, first\_name, last\_name, avatar.

### Test 3: Get user by id (negative)

**Method:** GET

**Purpose of the test:** Attempt to retrieve information about a user that does not exist

**Endpoint:** /api/users/{id} (see Figure 3)

**Input:** none

**Expected result:** HTTP Status Code: 404 – Not Found

**Actual result:** as expected

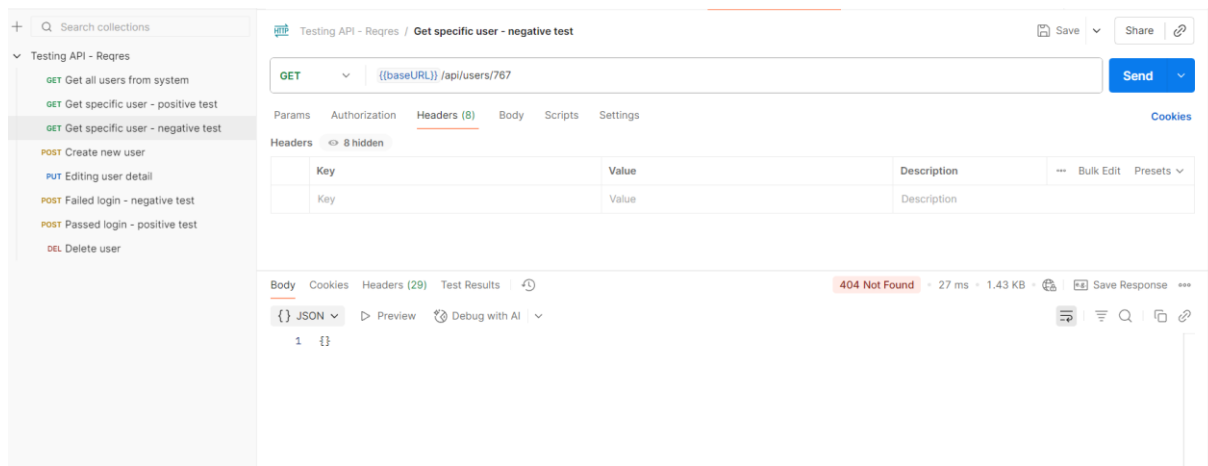


Figure 3 Get particular user, which isn't in the system

*Description:* If the ID provided in the path does not exist, the API responds with HTTP 404 and no content.

#### **Test 4: Register user**

*Method:* POST

*Purpose of the test:* Creating a user account

*Endpoint:* /api/register (see Figure 4)

*Input:*

```
{
  "email": "eve.holt@reqres.in",
  "password": "pistol"
}
```

*Expected result:* HTTP Status Code: 200 – Success. The user get an id and generated token in the response.

*Actual result:* as expected

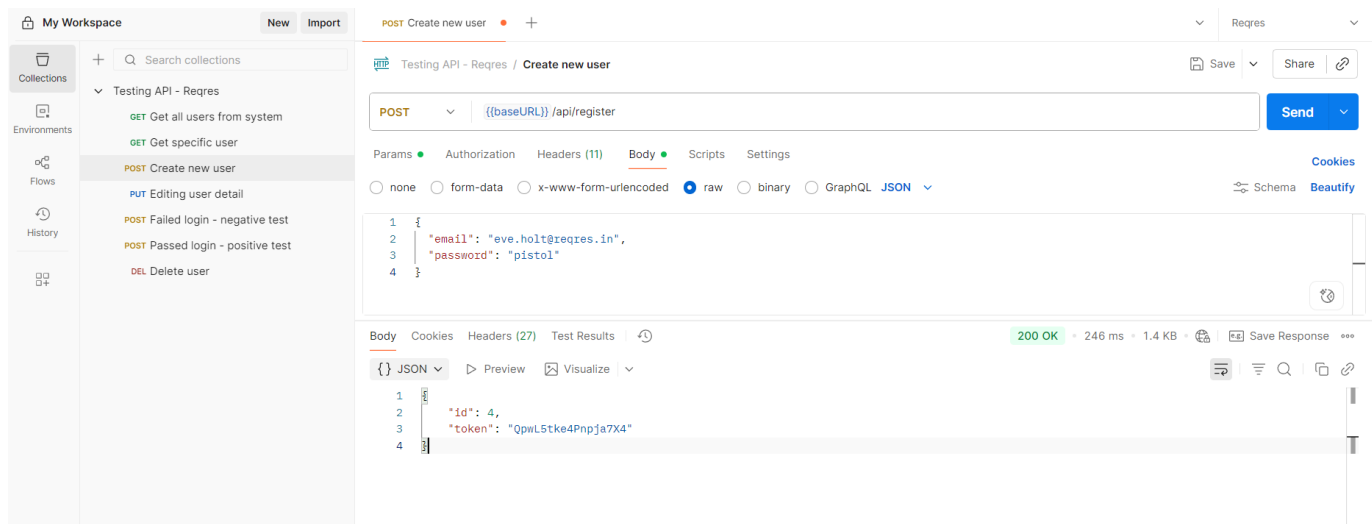


Figure 4 Relevant data that will enable you to create an account

**Description:** The endpoint works only for a few predefined accounts, which are defined in the system. You can't enter any email address. If you try give the data, which are not in the system, Postman throws status code: 400 (Bad request)



Figure 5 Response of API, if user uses data which aren't in the system

### Test 5: Edit user detail – positive test

**Method:** PUT

**Purpose of the test:** Modification of user's detail

**Endpoint:** `/api/users/{id}` (see Figure 6)

**Input:**

```
{
  "job": "QA Manual Engineer"
}
```

**Expected result:** HTTP Status Code: 200 – Success. User data have been updated.

**Actual result:** as expected

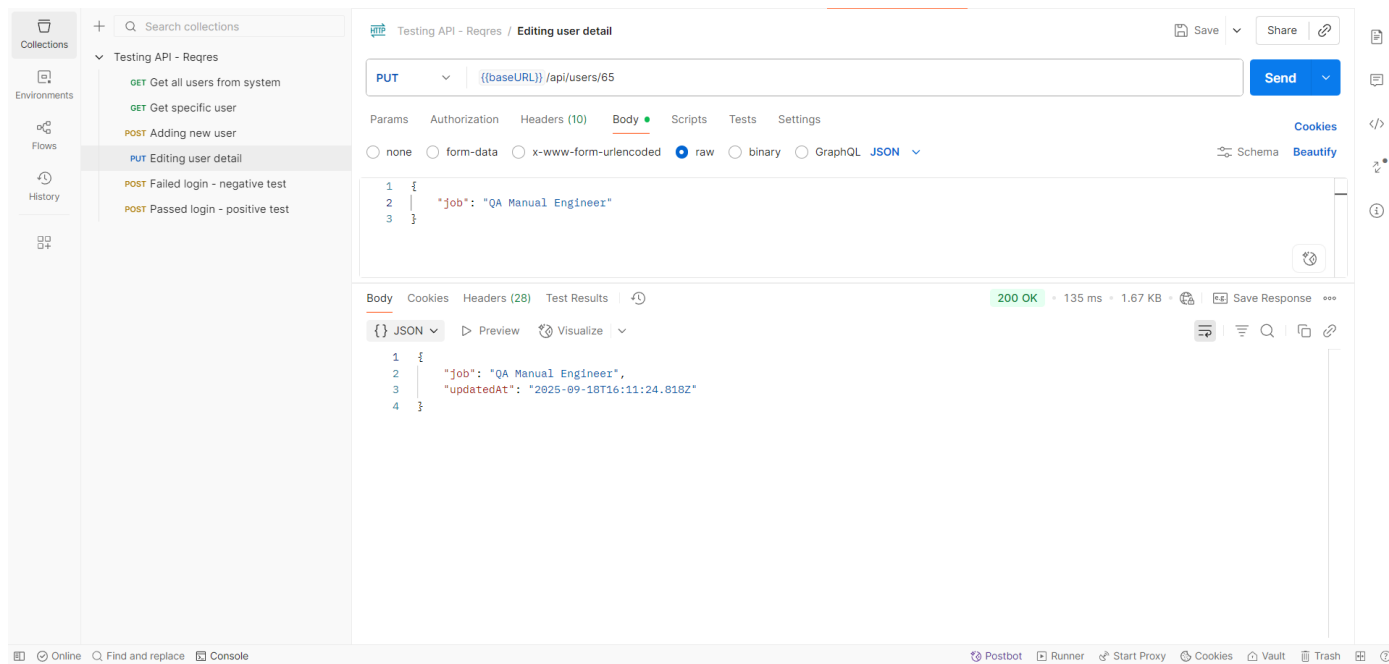


Figure 6 Editing user detail

**Description:** In RAW, in JSON format, specify the occupation for which the user wishes to update the relevant field. When the update is successful, the response will show the job title and the time of the update.

### Test 6: Edit user detail – Sending empty body in PUT request.

**Method:** PUT

**Purpose of the test:** Modification of user's detail when the request body is empty

**Endpoint:** `/api/users/{id}` (see Figure 7)

**Input:** none

**Expected result:** API should still return 200 (success) and provide an `updatedAt` field.

**Actual result:** as expected

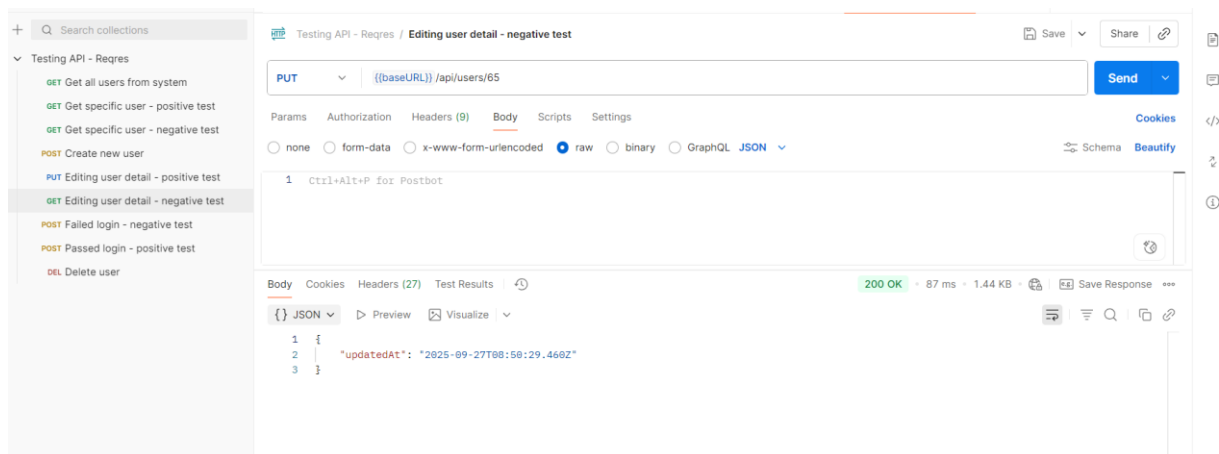


Figure 7 Editing user detail – empty body

*Description:* Unlike many APIs, reqres.in accepts empty body for PUT requests and still responds with 200. This is a predefined mock behavior of the demo API.

### Test 7: Login – positive test

*Method:* POST

*Purpose of the test:* Verify successful login with valid credentials

*Endpoint:* `/api/login` (see Figure 8)

*Input:*

```
{
  "email": "eve.holt@reqres.in",
  "password": "cityslicka"
}
```

*Expected result:* HTTP Status Code: 200 – Success. Token has been created.

*Actual result:* as expected

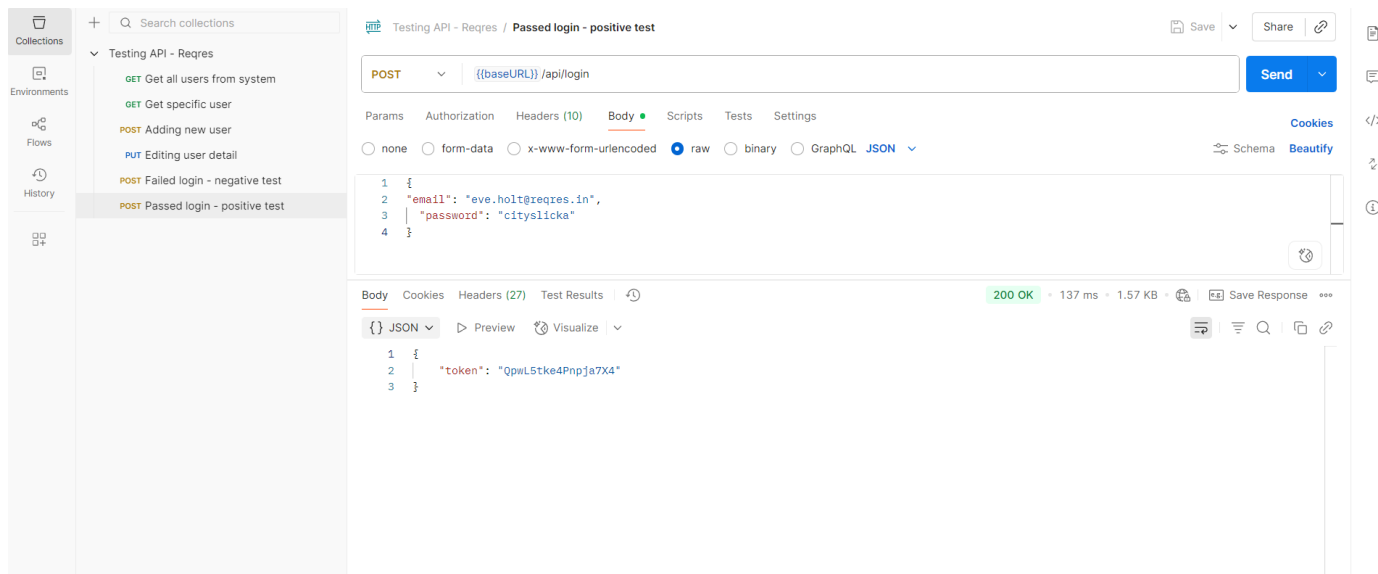


Figure 8 Logging - positive test

*Description:* The login endpoint only works for predefined test accounts. Other email/password combinations are rejected.

### Test 8: Login – negative test

*Method:* POST

*Purpose of the test:* Verify login fails when only the email is provided without a password

*Endpoint:* /api/login (see Figure 9)

*Input:*

```
{  
  "email": "eve.holt@reqres.in"  
}
```

*Expected result:* HTTP Status Code: 400 – Bad Request. Token isn't generated. The user receives an appropriate message about the missing password.

*Actual result:* as expected



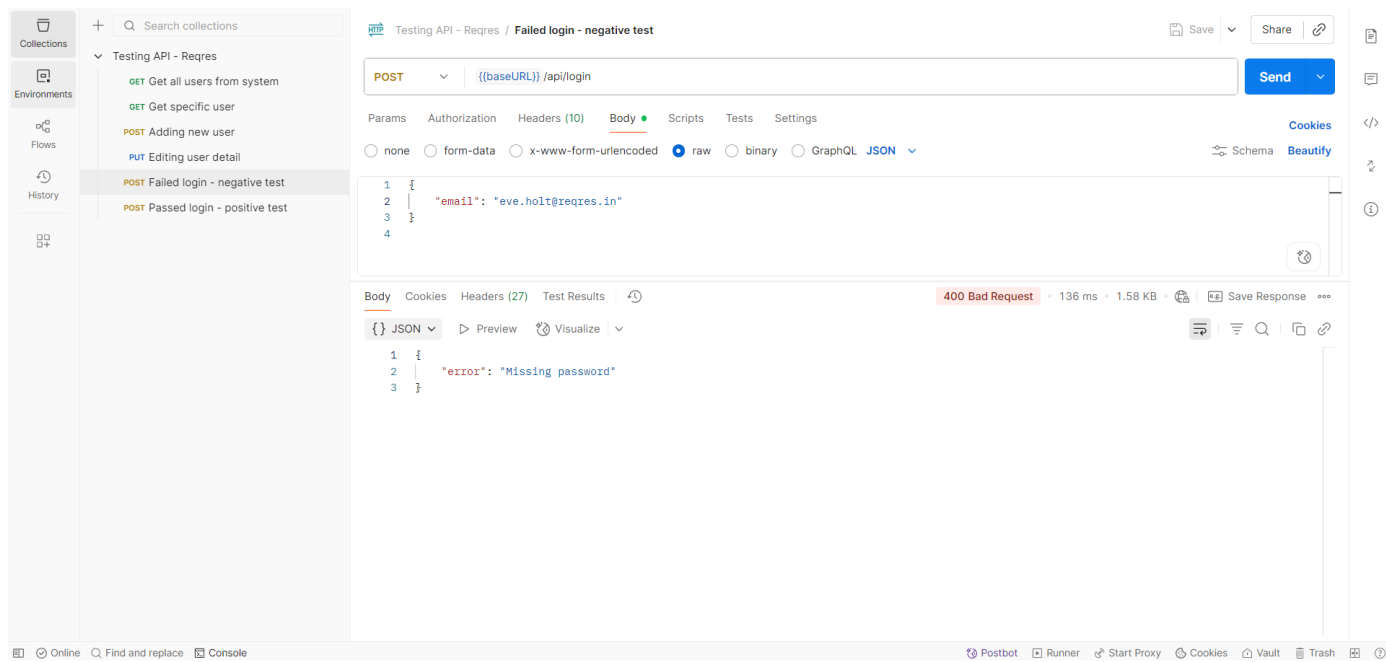


Figure 9 Logging - negative test

## Test 9: Delete user from system

*Method:* DELETE

*Purpose of the test:* Delete a user account by ID

*Endpoint:* /api/users/{id} (see Figure 10)

*Input:* none

*Expected result:* HTTP Status Code: 204 – Success.

*Actual result:* as expected

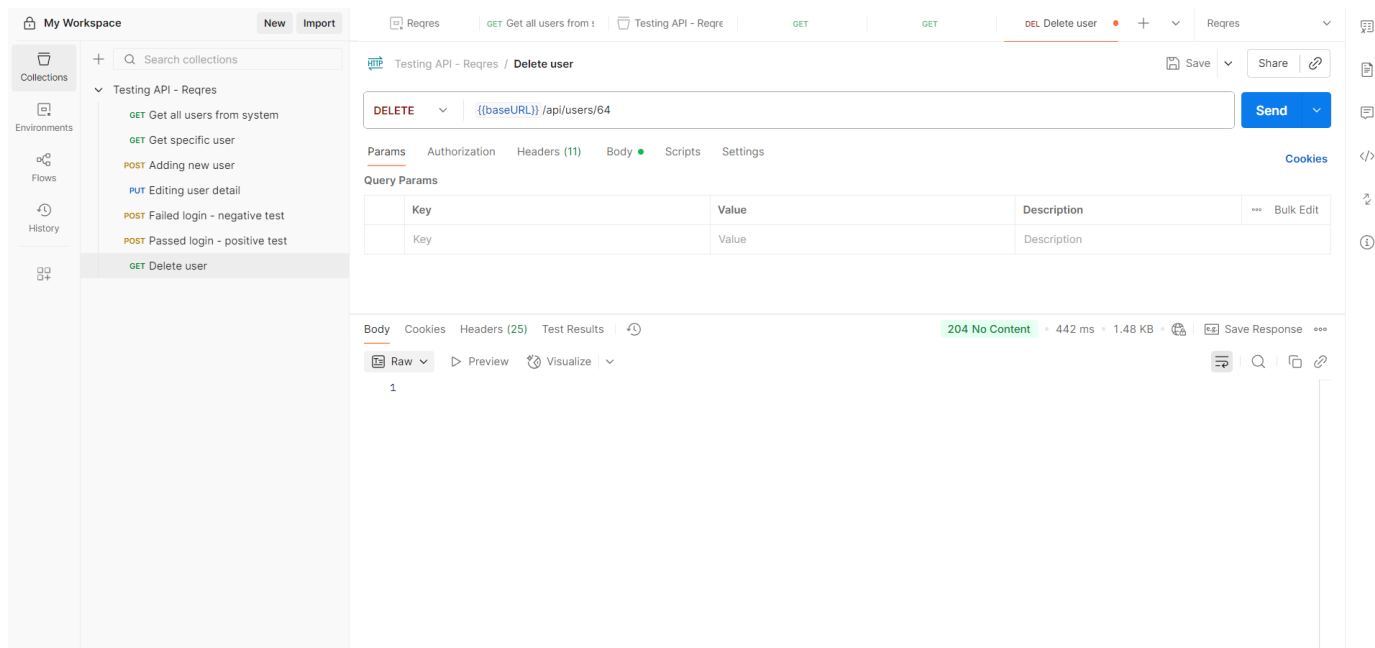


Figure 10 Delete user

**Description:** This endpoint deletes the specified user. Even if the user does not exist, the API still returns HTTP 204 with no content.