DOCUMENT

1. Get detail user

Type: GET

URL: https://reqres.in/api/users/{id}

Request:

Header

Field Name	Description	Type	Required	Note
x-api-key	**API key**	**String**	**Y**	**reqres-free-v1**

Body

Field Name	Description	Type	Required	Note

Response

```
{
  "data": {
    "id": 2,
    "email": "janet.weaver@reqres.in",
    "first_name": "Janet",
    "last_name": "Weaver",
    "avatar": "https://reqres.in/img/faces/2-image.jpg"
  }
}
```

2. Create user

Type: POST

URL: https://reqres.in/api/user

Request:

Header

Field Name	Description	Type	Required	Note
x-api-key	**API key**	**String**	**Y**	**reqres-free-v1**

Body

Field Name	Description	Type	Required	Note
email	**Email**	**String**	**Y**	janet.weaver@reqres.in
first_name	**First Name**	**String**	**Y**	Janet
last_name	**Last Name**	**String**	**Y**	Weaver
avatar	**Avatar**	**String**	**N**	https://reqres.in/img/.jpg

Response

```
{
  "email": "janet.weaver@reqres.in",
  "first_name": "Janet",
  "last_name": "Weaver",
  "avatar": "https://reqres.in/img/faces/2-image.jpg",
  "id": "704",
  "createdAt": "2025-08-30T07:12:31.119Z"
}
```

3. Update user

Type: PUT

URL: https://reqres.in/api/users/{id}

Request:

Header

Field Name	Description	Type	Required	Note
x-api-key	**API key**	**String**	**Y**	**reqres-free-v1**

Body

Field Name	Description	Type	Required	Note
email	**Email**	**String**	**Y**	janet.weaver@reqres.in
first_name	**First Name**	**String**	**Y**	Janet
last_name	**Last Name**	**String**	**Y**	Weaver
avatar	**Avatar**	**String**	**N**	https://reqres.in/img/faces/2-image.jpg

Response

```
{
      "email": "janet.weaver@reqres.in",
      "first_name": "Janet",
      "last_name": "Weaver",
      "avatar": https://reqres.in/img/faces/2-image.jpg,
      "updatedAt": "2025-08-30T07:15:02.969Z"
}
```

4. Delete user

Type: DELETE

URL: https://reqres.in/api/users/{id}

Request:

Header

Field Name	Description	Type	Required	Note
x-api-key	API key	String	Y	reqres-free-v1

Body

Field Name	Description	Type	Required	Note

Response

```
{
}
```

Test Plan

I. Get detail user: https://reqres.in/api/users/{id}

- **Validate URL:**

- Wrong endpoint (https://reqres.in/api/user/1) ← (missing “s”) Expect **404 Not Found**

➢ **Successful scenario**

- Fetch a old user (id=1) https://reqres.in/api/users/1 → Expect **200 OK** + data matches schema.
- Fetch a new user created https://reqres.in/api/users/2 → Expect **200 OK** + data matches schema.
- Fetch a no id https://reqres.in/api/users → Expect **200 OK** + list data matches schema.

➢ **Failure scenario**

- Fetch a non-existent user (id=23) → Expect **404 Not Found**
- Fetch a deleted-user (id=22) → Expect **404 Not Found**
- Fetch a wrong format (id=@#$) → Expect **400 Bad Request**

- **Server timeout:**

- Get user valid id but server time out → Expect **408 Request Timeout.**

II. Create user: https://reqres.in/api/user

- **Validate URL:**

- Wrong endpoint (https://reqres.in/api/users) ← (extra “s”) Expect **404 Not Found**
- Wrong endpoint with ID (https://reqres.in/api/users/2) ← (extra “id”) Expect **405 Method Not Allowed → "POST not supported on /users/{id}"**

➢ **Successful scenario**

- Create user successfully: Valid body → Expect **201 Created** + new ID unique + createdAt + data matches
- Re-create user after delete: Delete user → Create the same user again → Expect **201 Created** + new ID unique + new createdAt + data matches

➢ **Failure scenario**

- **Invalid Request:**

- Missing field required: Send only {"email":"John@gmail.com"} → Expect **400 Bad Request** → “Missing required field”

- Empty field required: Empty field {"email":""} → Expect **400 Bad Request** → “Empty required field”
- Empty body: Empty body {} → Expect **400 Bad Request** → “Empty required field”
- Wrong format field: Send field {"email":"Johngmail "} → Expect **400 Bad Request** → “Invalid type

- **Invalid Logic:**
- Create a user include field “id” on Request body → Expect **400 Bad Request** → Field 'id' is not allowed
- Create a user that **already exists** (email exist) → Expect **409 Conflict** → “User already exists” Or **400 Bad Request** with error message

- **Server timeout:**
- Create user valid body but server time out → Expect **408 Request Timeout**

III. Update user: https://reqres.in/api/users/{id}

- **Validate URL:**
- Wrong endpoint (https://reqres.in/api/user) ← (missing “id”) → Expect **400 Bad Request → “Missing user ID in path”**
- Wrong endpoint (https://reqres.in/api/user/{id}) ← (missing “s”) → Expect **404 Not Found**

- **Successful scenario**

- Update user successfully: Valid body → Expect **200 OK** + old ID + old createdAt + new updatedAt + data matches

- **Failure scenario**

- **Invalid Request:**

- Missing field required: Send only {"email":"John@gmail.com"} → Expect **400 Bad Request** → “Missing required field” → Retain the existing data
- Empty field required: Empty field {"email":""} → Expect **400 Bad Request** → “Empty required field” → Retain the existing data
- Empty body: Empty body {} → Expect **400 Bad Request** → “Request body cannot be empty”
- Wrong format field: Send field {"email":"Johngmail"} → Expect **400 Bad Request** → “Invalid type → Retain the existing data

- **Invalid Logic:**
- Update user include field “id” on Request body → Expect **400 Bad Request** → Field 'id' is not allowed → Retain the existing data
- Update user that **email already exists**: Update email exist → Expect **409 Conflict** → “User already exists” → Retain the existing data

- Update user with invalid role escalation → Expect **403 Forbidden** → "You are not authorized to assign this role" → Retain the existing data
- Update user with ID not exist (https://reqres.in/api/user/abczxc) → Expect **404 Not Found**
- Update user with ID deleted (https://reqres.in/api/user/1) → Expect **404 Not Found**

- **Server timeout:**

- Update user valid body but server time out → Expect **408 Request Timeout**

IV. Delete user: https://reqres.in/api/users/{id}

- **Validate URL:**
- Wrong endpoint (https://reqres.in/api/user) ← (missing "id") → Expect **400 Bad Request → "Missing user ID in path"**
- Wrong endpoint (https://reqres.in/api/user/{id}) ← (missing "s") → Expect **404 Not Found**

➢ **Successful scenario**

- Delete user successfully with valid "id": → Expect **204 No Content** + no data on database or Status = "Deleted"

➢ **Failure scenario**

- Delete user with valid "id" & include request body → Expect **400 Bad Request** → User still exist on database or status is still "**Actived**"
- Delete with ID not exist (https://reqres.in/api/user/abczxc) → Expect **404 Not Found**
- Delete with ID deleted (https://reqres.in/api/user/1) → Expect **404 Not Found**
- Delete with invalid role escalation → Expect **403 Forbidden** → "You are not authorized to assign this role" → User still exist on database or status is still "**Actived**"

- **Server timeout:**

- Delete user valid header but server time out → Expect **408 Request Timeout**