

### 3. API tests- “reqres.in” using “Postman”

Test case 1: User list			
Steps	Action	Expected Result	Actual Results
<b>1.1 Functionality</b>			
<b>1.1.1 Get user list from page 2 only</b>			
Pre-requisites:			
1. Open Postman and create an account 2. Adding new request in the “Collection” 3. Adding “Get” request <a href="https://reqres.in/api/users?page=2">https://reqres.in/api/users?page=2</a> 4. Adding api key: “x-api-key”, and value: “reqres-free-v1”			
1.	Press on “Send”	1. Status response: 200 2. Since there are 12 users that are split into 2 page results, here the Jason should contain data of six users 7-12 display from result page 2 with correct Jason schema contains their id (=number), email (=string), first name (=string), last name (=string), and link to their avatar image (=string). <pre>{     "page": 2,     "per_page": 6,     "total": 12,     "total_pages": 2,     "data": [       {         "id": 7,         "email": "michael.lawson@reqres.in",         "first_name": "Michael",         "last_name": "Lawson",         "avatar": "https://reqres.in/img/faces/7-image.jpg"       },       {         "id": 8,         "email": "lindsay.ferguson@reqres.in",         "first_name": "Lindsay",         "last_name": "Ferguson",         "avatar": "https://reqres.in/img/faces/8-image.jpg"       },       {         "id": 9,         "email": "tobias.funke@reqres.in",         "first_name": "Tobias",         "last_name": "Funke",         "avatar": "https://reqres.in/img/faces/9-image.jpg"       },       {         "id": 10,         "email": "byron.fields@reqres.in",         "first_name": "Byron",         "last_name": "Fields",         "avatar": "https://reqres.in/img/faces/10-image.jpg"       },       {         "id": 11,         "email": "george.edwards@reqres.in",         "first_name": "George",         "last_name": "Edwards",         "avatar": "https://reqres.in/img/faces/11-image.jpg"       },       {         "id": 12,         "email": "rachel.howell@reqres.in",         "first_name": "Rachel"       }     ]   }</pre>	Pass

		<pre>"last_name": "Howell", "avatar": "https://reqres.in/img/faces/12-image.jpg" }</pre>	
<b>1.1.2 Get the full user list with all existing users</b>			
Pre-requisites:			
1	Press "send"	1. Response status: 200 2. Getting a list with all the 12 existing users	<p>Fail. Display only the first page results, of users 1-6 instead of all users 1-12:</p> <pre>{   "page": 1,   "per_page": 6,   "total": 12,   "total_pages": 2,   "data": [     {       "id": 1,       "email": "george.bluth@reqres.in",       "first_name": "George",       "last_name": "Bluth",       "avatar": "https://reqres.in/img/faces/1-image.jpg"     },     {       "id": 2,       "email": "janet.weaver@reqres.in",       "first_name": "Janet",       "last_name": "Weaver",       "avatar": "https://reqres.in/img/faces/2-image.jpg"     },     {       "id": 3,       "email": "emma.wong@reqres.in",       "first_name": "Emma",       "last_name": "Wong",       "avatar": "https://reqres.in/img/faces/3-image.jpg"     },     {       "id": 4,       "email": "eve.holt@reqres.in",       "first_name": "Eve",       "last_name": "Holt",       "avatar": "https://reqres.in/img/faces/4-image.jpg"     },     {       "id": 5,       "email": "charles.morris@reqres.in",       "first_name": "Charles",       "last_name": "Morris",       "avatar": "https://reqres.in/img/faces/5-image.jpg"     },     {       "id": 6,       "email": "tracey.ramos@reqres.in",       "first_name": "Tracey",       "last_name": "Ramos",       "avatar": "https://reqres.in/img/faces/6-image.jpg"     }   ] }</pre>

			<pre>"avatar": "https://reqres.in/img/faces/6-image.jpg" }</pre>

### 1.1.3 Adding new user and then verify it was added to the list of users

Pre-requisites:

1. Open Postman and create an account
2. Adding new request in the “Collection”
3. Adding “Post” request <https://reqres.in/api/users>
4. Adding to “Body” the Jason data:
 

```
{
  "email": "test@gmail.com",
  "first_name": "Test",
  "last_name": "Example",
  "avatar": "https://reqres.in/img/faces/12-image.jpg"
}
```
5. Adding api key: “x-api-key” , and value: “reqres-free-v1”

1	Press send	1. Status response: 201 2. Creation of new user with the following data: <pre>{   "email": "test@gmail.com",   "first_name": "Test",   "last_name": "Example",   "avatar": "https://reqres.in/img/faces/12-image.jpg",   "id": "749",   "createdAt": "2025-11-15T13:59:11.437Z" }</pre>	Pass
2	Go to the request of “Get” single user information (test 1.1.4): <a href="https://reqres.in/api/users/749">https://reqres.in/api/users/749</a>	User id 749 display with all its data	Fail. Response number “404 not found”.

### 1.1.4 Read single existing user information (for example user number 12)

Pre-requisites:

1. Open Postman and create an account
2. Adding new request in the “Collection”
3. Adding “Get” request <https://reqres.in/api/users/12>

Adding api key: “x-api-key” , and value: “reqres-free-v1”

1	Press on “send”	1. Successful tatus response: 200 2. Getting full information on user 12 only: <pre>{   "data": {     "id": 12,     "email": "rachel.howell@reqres.in",     "first_name": "Rachel",     "last_name": "Howell",     "avatar": "https://reqres.in/img/faces/12-image.jpg"   } }</pre>	Pass

## 1.2 Error Handling

### 1.2.1 Get invalid (not existing) page of user list (for example page 10)

Pre-requisites:			
1. Open Postman and create an account			
2. Adding new request in the "Collection"			
3. Adding "Get" request <a href="https://reqres.in/api/users?page=10">https://reqres.in/api/users?page=10</a> (due to the fact that there are maximum 2 pages of users)			
4. Adding api key: "x-api-key" , and value: "reqres-free-v1"			
1	Press "send"	"Data" is empty	Pass { "page": 10, "per_page": 6, "total": 12, "total_pages": 2, "data": [] }

#### 1.2.2 Read user that is not exist (for example user number 14 that is not existing)

Pre-requisites:			
1. Open Postman and create an account			
2. Adding new request in the "Collection"			
3. Adding "Get" request <a href="https://reqres.in/api/users/14">https://reqres.in/api/users/14</a> (due to the fact that there are maximum 12 users)			
4. Adding api key: "x-api-key" , and value: "reqres-free-v1"			
1	Press "send"	Response status "404 not found"	Pass

Test case 2: CRUD ("Read" tests been made at previous test case)			
Steps	Action	Expected Result	Actual Results
<b>2.1 Functionality</b>			
<b>2.1.1 Create new user</b>			
Pre-requisites:			
1. Open Postman and create an account			
2. Adding new request in the "Collection"			
3. Adding "Post" request <a href="https://reqres.in/api/users">https://reqres.in/api/users</a>			
4. Adding to "Body" the Jason data:			
{ "name": "Test", "job": "QA tester" }			
5. Adding api key: "x-api-key" , and value: "reqres-free-v1"			
1.	Press on "Send"	1. Successful response status: 201 2. Creation of the user with the correct data. Jason contains: { "name": "Test", "job": "QA tester", "id": "79", "createdAt": "2025-11-15T14:39:23.497Z" }	Pass
<b>2.1.2 Read user – see in previous test case of "user list"</b>			
<b>2.1.3 Update user number 3 and verify the updates are saved when reading user 3</b>			
Pre-requisites:			
1. Open Postman and create an account			
2. Adding new request in the "Collection"			

3. Adding "Put" request <https://reqres.in/api/users/3>
4. Adding to "Body" the Jason data:
 

```
{
        "name": "Emma update",
        "job": "teacher update"
      }
```
5. Adding api key: "x-api-key" , and value: "reqres-free-v1"

1	Press send	<ol style="list-style-type: none"> <li>1. Response: 200</li> <li>2. Jason contains update information:           <pre>{             "name": "Emma update",             "job": "teacher update",             "updatedAt": "2025-11-15T14:48:29.393Z"           }</pre> </li> </ol>	<b>Pass</b>
2	Verify if the data really updated at the request of "Get" user information at " <a href="https://reqres.in/api/users/3">https://reqres.in/api/users/3</a> "	<p>Expecting to see the updated data from above. For example Jason contains:</p> <pre>{   "id": 3,   "email": "emma.wong@reqres.in",   "first_name": "Emma update",   "last_name": "Wong",   "Job": "Teacher update"   "avatar": "<a href="https://reqres.in/img/faces/3-image.jpg">https://reqres.in/img/faces/3-image.jpg</a>"   "updatedAt": "2025-11-15T14:48:29.393Z" },</pre>	<b>Fail. Jason contains the previous data without the update:</b> <pre>{   "data": {     "id": 3,     "email": "emma.wong@reqres.in",     "first_name": "Emma",     "last_name": "Wong",     "avatar": "<a href="https://reqres.in/img/faces/3-image.jpg">https://reqres.in/img/faces/3-image.jpg</a>"   }, }</pre>

#### 2.1.4 Delete user and verify it is not display at the user list

Pre-requisites:

1. Open Postman and create an account
2. Adding new request in the "Collection"
3. Adding "DELETE" request <https://reqres.in/api/users/2>
4. Adding api key: "x-api-key" , and value: "reqres-free-v1"

1	Press on "send"	Response "No content 204"	<b>Pass</b>
2	Verify that user 2 deleted and doesn't display when getting user list. For example: "Get" request of Reading " <a href="https://reqres.in/api/users/2">https://reqres.in/api/users/2</a> "	Response "404", empty body	<b>Fail. "200" response with Jason contains data of user 2:</b> <pre>{   "data": {     "id": 2,     "email": "janet.weaver@reqres.in",     "first_name": "Janet",     "last_name": "Weaver",     "avatar": "<a href="https://reqres.in/img/faces/2-image.jpg">https://reqres.in/img/faces/2-image.jpg</a>"   }, }</pre>
3	Press "send" again (to delete it one more time)	In this Mock API it always returns "204 No content" which indicates on idempotent. However in real world API I was expected to have a return of "404 not found" when trying to delete the same users twice.	

#### 2.2 Error Handling

##### 2.2.1 Create user with empty name parameter

Pre-requisites:

1. Open Postman and create an account
2. Adding new request in the "Collection"
3. Adding "Post" request <https://reqres.in/api/users>
4. Adding to "Body" the Jason data:

```
{
  "name": "",
  "job": "example"
}
```

5. Adding api key: "x-api-key" , and value: "reqres-free-v1"

1	Press "send"	1. Response "400" Bad request 2. Jason contains error message such as <pre>{   "error": "Name is required" }</pre>	<b>Fail</b> "201" response with Jason contains creation of new user despite there is no name: <pre>{   "name": "",   "job": "QA tester",   "id": "304",   "createdAt": "2025-11-15T15:05:40.272Z" }</pre>

### 2.2.2 Creating duplicating user

Pre-requisites:

1. Open Postman and create an account
2. Adding new request in the "Collection"
3. Adding "Post" request <https://reqres.in/api/users>
4. Adding to "Body" the Jason data:  

```
{
  "name": "USER",
  "job": "duplicate"
}
```

Adding api key: "x-api-key" , and value: "reqres-free-v1"

1	Press "send"	Response status "201" with Jason contains: <pre>{   "name": "USER",   "job": "duplicate",   "id": "886",   "createdAt": "2025-11-15T15:12:03.688Z" }</pre>	<b>Pass</b>
2	Repeat exactly the pre-requisites and press send	Response that indicates duplication/ conflicts such as "409 conflict", with Jason contains error message such as: <pre>{"error": "User already exists"}</pre>	<b>"201" response with Jason contains creation of user:</b> <pre>{   "name": "USER",   "job": "duplicate",   "id": "348",   "createdAt": "2025-11-15T15:15:06.405Z" }</pre> <p><b>Although a new user is created with a unique ID, the same name and job are reused, which can cause confusion. In this situation, I would consult with the development and product teams about adding more fields to help distinguish similar users.</b></p>

### Test case 3: Login

Steps	Action	Expected Result	Actual Results
<b>3.1 Functionality</b>			
<b>3.1.1 Successful Login</b>			
Pre-requisites:			
1. Open Postman and create an account			

2. Adding new request in the “Collection”
3. Adding “Post” request <https://reqres.in/api/login>
4. Adding to “Body” the Jason data:

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

5. Adding api key: “x-api-key”, and value: “reqres-free-v1”

1.	Press on “Send”	1. Successful response status: 200 2. Jason contains of token: <pre>{   "token": "QpwL5tke4Pnpja7X4" }</pre>	Pass

### 3.1.2 Try to log in after having new registration

Pre-requisites:

1. Open Postman and create an account
  2. Adding new request in the “Collection”
  3. Adding “Post” request <https://reqres.in/api/register>
  4. Adding to “Body” the Jason data:
- ```
{
  "email": "new@gmail.com",
  "password": "123456"
}
```
5. Adding api key: “x-api-key”, and value: “reqres-free-v1”

|   |              |                                                                                                  |                                                                                                                                                                                                                         |
|---|--------------|--------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1 | Press “send” | “200” response, with Jason contains:<br><pre>{   "id": 4,   "token": "QpwL5tke4Pnpja7X4" }</pre> | <b>Fail (maybe the system doesn't support creating new registrations).</b><br><b>“400 Bad response” with Jason contains error message:</b><br><pre>{   "error": "Note: Only defined users succeed registration" }</pre> |
|   |              |                                                                                                  |                                                                                                                                                                                                                         |

## 3.2 Error Handling

### 3.2.1 Trying to log in with user that is not register

Pre-requisites:

1. Open Postman and create an account
  2. Adding new request in the “Collection”
  3. Adding “Post” request <https://reqres.in/api/register>
  4. Adding to “Body” the Jason data:
- ```
{
  "email": "notregister@gmail.com",
  "password": "0000"
}
```

Adding api key: “x-api-key”, and value: “reqres-free-v1”

1	Press “send”	1. Response “400” Bad request 2. Jason contains error message such as <pre>{   "error": "user not found" }</pre>	Pass

### 3.2.1 Log in with missing password

Pre-requisites:

1. Open Postman and create an account
2. Adding new request in the “Collection”
3. Adding “Post” request <https://reqres.in/api/login>
4. Adding to “Body” the Jason data:

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

Adding api key: “x-api-key” , and value: “reqres-free-v1”

1	Press “send”	1. Response “400” Bad request 2. Jason contains error message such as { "error": "Missing password" }	Pass
---	--------------	---	------