

# API Test Plan – Reqres API

## Project Information

- **Project Name:** Reqres API Testing
- **Tester:** Alana Oleksiienko
- **Environment:** Public API – <https://reqres.in>
- **Tool Used:** Postman (manual testing)
- **Test Type:** Functional API Testing
- **Date:** 04/27/25

## Objective

To validate the functionality, stability, and correctness of the Reqres public API endpoints. This includes verifying HTTP status codes, payload structure, response data, and basic error handling. The API is used to simulate user-related CRUD operations and authentication scenarios.

## Test Scope

### In-Scope

- Authentication (Login)
- Create, Read, Update, Delete (CRUD) operations for users
- Status code validation
- Response body structure validation

### Out-of-Scope

- Backend/database validation (not available via public API)
- Performance testing
- Security testing

## API Endpoints Covered

No.	Feature	Method	Endpoint	Description
1	Login User	POST	/api/login	Authenticates the user and returns a token.
2	Create User	POST	/api/users	Creates a new user with name and job.
3	Update User	PUT	/api/users/2	Updates all fields of user ID 2.
4	Delete User	DELETE	/api/users/2	Deletes the user with ID 2.
5	Get User List	GET	/api/users?page=2	Retrieves a list of users from page 2.
6	Get Single User	GET	/api/users/2	Retrieves details of the user with ID 2.

## Test Scenarios

1.

### Login API

- TC01 – Login with valid credentials → Expect 200 OK and non-empty token
- TC02 – Login with invalid credentials → Expect 400 Bad Request and error message

2.

### Create User

- TC03 – Create user with valid data → Expect 201 Created and response with id, name, job, and createdAt
- TC04 – Create user with missing fields → Expect appropriate error (if handled)

3.

### Update User

- TC05 – Full update of existing user (PUT) → Expect 200 OK and updated name, job, updatedAt
- TC06 – Update user with empty payload → Observe and record behavior

4.

### Delete User

- TC07 – Delete existing user → Expect 204 No Content and no response body
- TC08 – Delete non-existing user → Behavior depends on API design (still returns 204 or error)

5.

### Get User List

- TC09 – Retrieve list of users (page 2) → Expect 200 OK and list of user objects

6.

### Get Single User

- TC10 – Retrieve valid user (ID 2) → Expect 200 OK and user details
- TC11 – Retrieve invalid user (non-existing ID) → Expect 404 Not Found

## Test Data

### Login Credentials

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

```
"password": "cityslicka"
}
```

### ✓ Create User Payload

```
{
  "name": "morpheus",
  "job": "leader"
}
```

### ✓ Update User Payload

```
{
  "name": "neo",
  "job": "the one"
}
```

## Test Execution Checklist

Test Case ID	Description	Status
TC01	Valid login	✓ Pass
TC02	Invalid login	✓ Pass
TC03	Create user with valid data	✓ Pass
TC04	Create user with missing fields	✓ Pass
TC05	Update user with full data (PUT)	✓ Pass
TC06	Update user with empty body	✓ Pass
TC07	Delete existing user	✓ Pass
TC08	Delete non-existing user	✓ Pass
TC09	Get list of users	✓ Pass
TC10	Get single valid user	✓ Pass
TC11	Get single invalid user	✓ Pass

## Expected Outcomes

- All CRUD endpoints respond with correct HTTP status codes
- Response bodies are structured correctly with all expected fields
- Appropriate error messages are returned for negative tests

POSTMAN COLLECTION ATTACHED BELOW

<https://www.postman.com/alealex22/workspace/api-testing-regres/collection/19623284-26582b45-e89c-4f3c-8f1c-2f1faf58ade6?action=share&creator=19623284>