API Test Plan

# Project Introduction

## This project aims to ensure the reliability and functionality of the ReqRes API (<https://reqres.in/>) endpoints through a robust automation testing framework. The API provides various endpoints for user management, including login, retrieval, updating, and deletion of users.

## The Test Plan has been created to communicate the test approach to team members. It includes the objectives, scope, schedule, risks and approach.  This document will clearly identify what the test deliverables will be and what is deemed in and out of scope.

# Test Approach

#### **Tools and Framework:**

**Playwright**: Chosen for the following reasons:

* + **JavaScript-based**: Playwright is a JavaScript base framework that aligns with the project requirements.
  + **Reusable and Scalable:** Playwright support Page Object Model (POM) Pattern, Data-Driven Testing, provide reuse common setups (e.g., browser launches, authentication) across multiple tests, support Cross-Browser Compatibility.
  + **Easy to Maintain:** Offers a clear syntax and well-documented API.
  + Supports API testing out of the box.
  + Provides powerful browser automation capabilities which can be extended to API testing.
  + Rich set of features for asserting API responses and handling different environments.
  + Supports parallel execution, speeding up the test suite run times.
  + **Detailed Reporting**: Playwright offers detailed and clear reports which help in understanding the test outcomes effectively.
  + **Integration with CI/CD**: Playwright can be easily integrated with continuous integration and continuous deployment pipelines, streamlining the testing process.

#### **Test Script Development and Execution:**

* + **Incremental Development**: Test scripts will be developed immediately after the completion of manual testing for each user story.
  + **Execution Strategy**: Tests will be executed incrementally with internal builds and major releases.
  + **Testing Goals**: The test goal will be executed for smoke test and regression test.
  + **These test cases will be automated:**
    - The repetitive tasks look like smoke test and regression test
    - Involve complex business logic
  + **Sprint-wise Automation**: Test automation will start in each sprint.
  + **Endpoint and End2End Testing**: Individual endpoint tests as well as workflow-based tests will be automated.
  + **Environment Coverage**: Tests will run in different environments (Test/Stage).
  + **Reporting**: Test reports will be clear, concise, and easy to interpret.

# Scope of testing

We will conduct comprehensive testing on **4** **endpoints** of the Reqres API:

1. POST: /api/login (Login - Successful/Unsuccessful)

2. GET: /api/users/{id} (Retrieve a single user)

3. PUT: /api/users/{id} (Update a user)

4. DELETE: /api/users/{id} (Delete a user)

Below is a detailed list of test cases for each endpoint.

1. **POST: /api/login (Login - Successful/Unsuccessful)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No | Test case Description | Test Type | Test Data | Expected result |
| 1 | |  | | --- | | Successful login with valid credentials |  |  | | --- | |  | | |  | | --- | | Endpoint Test |  |  | | --- | |  | | |  | | --- | | Valid email and password |  |  | | --- | |  | | 200 OK, token in response |
| 2 | Unsuccessful login with invalid email | Endpoint Test | Invalid email, valid password | 400 Bad Request, error message |
| 3 | |  | | --- | | Unsuccessful login with invalid password |  |  | | --- | |  | | Endpoint Test | Valid email, invalid password | |  | | --- | | 400 Bad Request, error message |  |  | | --- | |  | |
| 4 | |  | | --- | | Login with missing email |  |  | | --- | |  | | Endpoint Test | No email, valid password | 400 Bad Request, error message |
| 5 | |  | | --- | | Login with missing password |  |  | | --- | |  | | Endpoint Test | |  | | --- | | Valid email, no password |  |  | | --- | |  | | 400 Bad Request, error message |
| 6 | |  | | --- | | Login with empty email |  |  | | --- | |  | | Endpoint Test | |  | | --- | | Empty email, valid password |  |  | | --- | |  | | 400 Bad Request, error message |
| 7 | |  | | --- | | Login with empty password |  |  | | --- | |  | | Endpoint Test | |  |  |  | | --- | --- | --- | | |  | | --- | | Valid email, empty password |  |  | | --- | |  | |  |  | | --- | |  | | 400 Bad Request, error message |
| 8 | |  | | --- | | Login with SQL injection attempt |  |  | | --- | |  | | |  | | --- | | Security Test |  |  | | --- | |  | | |  | | --- | | Email with SQL injection code, valid password |  |  | | --- | |  | | |  | | --- | | 400 Bad Request or specific error handling |  |  | | --- | |  | |
| 9 | |  | | --- | | Login with XSS attack in email |  |  | | --- | |  | | Security Test | |  | | --- | | Email with XSS code, valid password |  |  | | --- | |  | | |  | | --- | | 400 Bad Request or specific error handling |  |  | | --- | |  | |
| 10 | |  | | --- | | Login with special characters in email/password |  |  | | --- | |  | | Endpoint Test | |  | | --- | | Email and password with special characters |  |  | | --- | |  | | |  | | --- | | 400 Bad Request or specific error handling |  |  | | --- | |  | |

1. **GET: /api/users/{id} (Retrieve a Single User)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No | Test case Description | Test Type | Test Data | Expected result |
| 1 | |  | | --- | | Retrieve an existing user with valid ID |  |  | | --- | |  | | |  | | --- | | Endpoint Test |  |  | | --- | |  | | |  | | --- | | Valid user ID |  |  | | --- | |  | | |  | | --- | | 200 OK, user details in response |  |  | | --- | |  | |
| 2 | |  | | --- | | Retrieve a non-existent user |  |  | | --- | |  | | |  | | --- | | Endpoint Test |  |  | | --- | |  | | |  | | --- | | Invalid user ID |  |  | | --- | |  | | |  | | --- | | 404 Not Found, error message |  |  | | --- | |  | |
| 3 | |  | | --- | | Retrieve user with missing ID |  |  | | --- | |  | | |  | | --- | | Endpoint Test |  |  | | --- | |  | | |  | | --- | | No user ID |  |  | | --- | |  | | |  | | --- | | 400 Bad Request, error message |  |  | | --- | |  | |
| 4 | |  | | --- | | Retrieve user with negative ID |  |  | | --- | |  | | |  | | --- | | Endpoint Test |  |  | | --- | |  | | |  | | --- | | Negative user ID |  |  | | --- | |  | | |  | | --- | | 400 Bad Request, error message |  |  | | --- | |  | |
| 5 | |  | | --- | | Retrieve user with zero ID |  |  | | --- | |  | | |  | | --- | | Endpoint Test |  |  | | --- | |  | | |  | | --- | | Zero as user ID |  |  | | --- | |  | | |  | | --- | | 404 Not Found, error message |  |  | | --- | |  | |
| 6 | |  | | --- | | Retrieve user with very large ID |  |  | | --- | |  | | |  | | --- | | Performance Test |  |  | | --- | |  | | |  | | --- | | Very large user ID |  |  | | --- | |  | | |  | | --- | | 404 Not Found or appropriate handling |  |  | | --- | |  | |
| 7 | |  | | --- | | Retrieve user with special characters in ID |  |  | | --- | |  | | |  | | --- | | Security Test |  |  | | --- | |  | | |  | | --- | | ID with special chars |  |  | | --- | |  | | |  | | --- | | 400 Bad Request or appropriate handling |  |  | | --- | |  | |
| 8 | |  | | --- | | Retrieve user with SQL injection attempt in ID |  |  | | --- | |  | | |  | | --- | | Security Test |  |  | | --- | |  | | |  | | --- | | ID with SQL injection |  |  | | --- | |  | | |  | | --- | | 400 Bad Request or specific error handling |  |  | | --- | |  | |
| 9 | |  | | --- | | Retrieve user with XSS attack in ID |  |  | | --- | |  | | |  | | --- | | Security Test |  |  | | --- | |  | | |  | | --- | | ID with XSS code |  |  | | --- | |  | | |  | | --- | | 400 Bad Request or specific error handling |  |  | | --- | |  | |
| 10 | |  | | --- | | Retrieve user with ID as null |  |  | | --- | |  | | |  | | --- | | Endpoint Test |  |  | | --- | |  | | |  | | --- | | Null as user ID |  |  | | --- | |  | | 400 Bad Request, error message |

# 

1. **PUT: /api/users/{id} (Update a user)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No | Test case Description | Test Type | Test Data | Expected result |
| 1 | |  |  |  | | --- | --- | --- | | |  | | --- | | Update an existing user with valid data |  |  | | --- | |  | |  |  | | --- | |  | | |  | | --- | | Endpoint Test |  |  | | --- | |  | | |  |  |  | | --- | --- | --- | | |  | | --- | | Valid user ID, valid user details |  |  | | --- | |  | |  |  | | --- | |  | | |  |  |  | | --- | --- | --- | | |  | | --- | | 200 OK, updated user details |  |  | | --- | |  | |  |  | | --- | |  | |
| 2 | |  |  |  | | --- | --- | --- | | |  | | --- | | Attempt to update a non-existent user |  |  | | --- | |  | |  |  | | --- | |  | | |  | | --- | | Endpoint Test |  |  | | --- | |  | | |  |  |  | | --- | --- | --- | | |  | | --- | | Invalid user ID, valid user details |  |  | | --- | |  | |  |  | | --- | |  | | |  |  |  | | --- | --- | --- | | |  | | --- | | 404 Not Found, error message |  |  | | --- | |  | |  |  | | --- | |  | |
| 3 | |  |  |  | | --- | --- | --- | | |  | | --- | | Update user with missing ID |  |  | | --- | |  | |  |  | | --- | |  | | |  | | --- | | Endpoint Test |  |  | | --- | |  | | |  |  |  | | --- | --- | --- | | |  | | --- | | No user ID, valid user details |  |  | | --- | |  | |  |  | | --- | |  | | |  |  |  | | --- | --- | --- | | |  | | --- | | 400 Bad Request, error message |  |  | | --- | |  | |  |  | | --- | |  | |
| 4 | |  |  |  | | --- | --- | --- | | |  | | --- | | Update user with missing user details |  |  | | --- | |  | |  |  | | --- | |  | | |  | | --- | | Endpoint Test |  |  | | --- | |  | | |  |  |  | | --- | --- | --- | | |  | | --- | | Valid user ID, no user details |  |  | | --- | |  | |  |  | | --- | |  | | |  | | --- | | 400 Bad Request, error message |  |  | | --- | |  | |
| 5 | |  |  |  | | --- | --- | --- | | |  | | --- | | Update user with partial user details |  |  | | --- | |  | |  |  | | --- | |  | | |  | | --- | | Endpoint Test |  |  | | --- | |  | | |  |  |  | | --- | --- | --- | | |  | | --- | | Valid user ID, partial user details |  |  | | --- | |  | |  |  | | --- | |  | | |  |  |  | | --- | --- | --- | | |  | | --- | | 200 OK, updated user details |  |  | | --- | |  | |  |  | | --- | |  | |
| 6 | |  |  |  | | --- | --- | --- | | |  | | --- | | Update user with invalid email format |  |  | | --- | |  | |  |  | | --- | |  | | |  |  | | --- | --- | | Endpoint Test |  |  |  | | --- | |  | | |  |  |  | | --- | --- | --- | | |  | | --- | | Valid user ID, invalid email format |  |  | | --- | |  | |  |  | | --- | |  | | |  |  |  | | --- | --- | --- | | |  | | --- | | 400 Bad Request, error message |  |  | | --- | |  | |  |  | | --- | |  | |
| 7 | |  |  |  | | --- | --- | --- | | |  | | --- | | Update user with special characters in name |  |  | | --- | |  | |  |  | | --- | |  | | |  |  | | --- | --- | | Endpoint Test |  |  |  | | --- | |  | | |  |  |  | | --- | --- | --- | | |  | | --- | | Valid user ID, special characters in name |  |  | | --- | |  | |  |  | | --- | |  | | |  |  |  | | --- | --- | --- | | |  | | --- | | 200 OK, updated user details |  |  | | --- | |  | |  |  | | --- | |  | |
| 8 | |  |  |  | | --- | --- | --- | | |  | | --- | | Update user with SQL injection attempt |  |  | | --- | |  | |  |  | | --- | |  | | |  | | --- | | Security Test |  |  | | --- | |  | | |  |  |  | | --- | --- | --- | | |  | | --- | | Valid user ID, SQL injection in details |  |  | | --- | |  | |  |  | | --- | |  | | |  |  |  | | --- | --- | --- | | |  | | --- | | 400 Bad Request or specific error handling |  |  | | --- | |  | |  |  | | --- | |  | |
| 9 | |  |  |  | | --- | --- | --- | | |  | | --- | | Update user with XSS attack in details |  |  | | --- | |  | |  |  | | --- | |  | | |  | | --- | | Security Test |  |  | | --- | |  | | |  | | --- | | Valid user ID, XSS code in details |  |  | | --- | |  | | |  |  |  | | --- | --- | --- | | |  | | --- | | 400 Bad Request or specific error handling |  |  | | --- | |  | |  |  | | --- | |  | |
| 10 | |  |  |  | | --- | --- | --- | | |  | | --- | | Update user with large payload |  |  | | --- | |  | |  |  | | --- | |  | | |  |  |  | | --- | --- | --- | | |  | | --- | | Performance Test |  |  | | --- | |  | |  |  | | --- | |  | | |  |  |  | | --- | --- | --- | | |  | | --- | | Valid user ID, very large user details |  |  | | --- | |  | |  |  | | --- | |  | | 200 OK or appropriate handling |

# 

1. **DELETE: /api/users/{id} (Delete a User)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No | Test case Description | Test Type | Test Data | Expected result |
| 1 | |  |  |  | | --- | --- | --- | | |  | | --- | | Delete an existing user with valid ID |  |  | | --- | |  | |  |  | | --- | |  | | |  | | --- | | Endpoint Test |  |  | | --- | |  | | |  | | --- | | Valid user ID |  |  | | --- | |  | | |  |  |  | | --- | --- | --- | | |  | | --- | | 204 No Content |  |  | | --- | |  | |  |  | | --- | |  | |
| 2 | |  |  |  | | --- | --- | --- | | |  | | --- | | Attempt to delete a non-existent user |  |  | | --- | |  | |  |  | | --- | |  | | |  | | --- | | Endpoint Test |  |  | | --- | |  | | |  | | --- | | Invalid user ID |  |  | | --- | |  | | |  | | --- | | 404 Not Found, error message |  |  | | --- | |  | |
| 3 | |  | | --- | | Delete user with missing ID |  |  | | --- | |  | | |  | | --- | | Endpoint Test |  |  | | --- | |  | | |  | | --- | | No user ID |  |  | | --- | |  | | |  | | --- | | 400 Bad Request, error message |  |  | | --- | |  | |
| 4 | |  | | --- | | Delete user with negative ID |  |  | | --- | |  | | |  | | --- | | Endpoint Test |  |  | | --- | |  | | |  | | --- | | Negative user ID |  |  | | --- | |  | | |  | | --- | | 400 Bad Request, error message |  |  | | --- | |  | |
| 5 | |  | | --- | | Delete user with zero ID |  |  | | --- | |  | | |  | | --- | | Endpoint Test |  |  | | --- | |  | | |  | | --- | | Zero as user ID |  |  | | --- | |  | | |  | | --- | | 404 Not Found, error message |  |  | | --- | |  | |
| 6 | |  | | --- | | Delete user with very large ID |  |  | | --- | |  | | |  | | --- | | Performance Test |  |  | | --- | |  | | |  | | --- | | Very large user ID |  |  | | --- | |  | | |  | | --- | | 404 Not Found or appropriate handling |  |  | | --- | |  | |
| 7 | |  | | --- | | Delete user with special characters in ID |  |  | | --- | |  | | |  | | --- | | Security Test |  |  | | --- | |  | | |  | | --- | | ID with special chars |  |  | | --- | |  | | |  | | --- | | 400 Bad Request or appropriate handling |  |  | | --- | |  | |
| 8 | |  | | --- | | Delete user with SQL injection attempt in ID |  |  | | --- | |  | | |  | | --- | | Security Test |  |  | | --- | |  | | |  | | --- | | ID with SQL injection |  |  | | --- | |  | | |  | | --- | | 400 Bad Request or specific error handling |  |  | | --- | |  | |
| 9 | |  | | --- | | Delete user with XSS attack in ID |  |  | | --- | |  | | |  | | --- | | Security Test |  |  | | --- | |  | | |  | | --- | | ID with XSS code |  |  | | --- | |  | | |  | | --- | | 400 Bad Request or specific error handling |  |  | | --- | |  | |
| 10 | |  | | --- | | Delete user with ID as null |  |  | | --- | |  | | |  | | --- | | Endpoint Test |  |  | | --- | |  | | |  | | --- | | Null as user ID |  |  | | --- | |  | | 400 Bad Request, error message |

# Out of scope

* Testing integrations with third-party services.
* Performance testing.
* Security testing (requires specialized tools).

# Test schedules

* **Post-Manual Testing**: Test scripts will be developed right after the completion of manual tests for each user story.
* **Spillover Sprints**: If there is insufficient time to develop test scripts within the current sprint, they will be scheduled for the next sprint.

# Environment

Two environments will be maintained:

* **Test Environment**: For initial development and testing of scripts.
* **Stage Environment**: For pre-production validation ensuring the reliability of the API in a near-live setting.

# Risks and challenges

* **API Changes**: Frequent changes in API endpoints or parameters can disrupt the automation scripts.
* **Data Dependency**: Tests might fail due to the dependency on data which might be altered by other tests or users.
* **Incomplete Requirements**: If requirements are not thoroughly defined, it can lead to inadequate test coverage.
* **Environment Stability**: Test and Stage environments must be stable and reflect production settings as closely as possible.
* **Time Constraints**: Limited time for developing and executing test scripts within sprints.
* **Data Management**: Ensuring test data availability and restoring data states after tests.
* **Tool Limitations**: Potential limitations or bugs within the Playwright framework.
* **Resource Availability**: Availability of skilled personnel to develop and maintain the test scripts.