REQRES – API

API Automation Framework

8/13/2025

Author: Tuan Le

**Version:** 1.0

**Created:** 08/13/2025

**Last Updated:** 08/15/2025

**Status:** Revied

**Revision and Signoff Sheet**

**Document History** - To maintain a list of changes being made

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Date | Author | Description of Change |
| 1 | 08/13/2025 | Tuan Le | Draft |
| 2 | 08/15/2025 | Tuan Le | Reviewed |
|  |  |  |  |

**Approvers List** - To track who has reviewed and signoff on the Test plan

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Role | Approver / Reviewer | Approval / Review Date |
|  |  |  |  |
|  |  |  |  |

**Reference Documents** - Clearly mark the document used as an input to create the test plan

|  |  |  |
| --- | --- | --- |
| **Version** | **Date** | **Document Name** |
| 1.0 |  | ReqRes API - https://reqres.in/api-docs/ |

Table of Contents

[1. PURPOSE 3](#_Toc206149511)

[2. TEST OBJECTIVES 3](#_Toc206149512)

[3. TIMELINE 3](#_Toc206149513)

[4. TEST ENVIRONMENT 3](#_Toc206149514)

[5. TEST STRATEGY 4](#_Toc206149515)

[5.1. Scope 4](#_Toc206149516)

[5.1.1. In-Scope 4](#_Toc206149517)

[5.1.2. Out Of Scope 4](#_Toc206149518)

[5.2. Approach 4](#_Toc206149519)

[5.3. Test Level 4](#_Toc206149520)

[6. TEST SCENARIOS 5](#_Toc206149521)

[7. TEST DATA/PAYLOAD 5](#_Toc206149522)

[8. ENTRY CRITERIA 6](#_Toc206149523)

[9. EXIT CRITERIA 6](#_Toc206149524)

[10. TEST TOOLS 6](#_Toc206149525)

# PURPOSE

This Test Plan describes the testing approach and practice of building an API automation framework demonstrated with the REQRES API (<https://reqres.in/api-docs/>).

# TEST OBJECTIVES

This document outlines the overall plan for testing selected REQRES API endpoints.

* Four endpoints (one each from **POST**, **GET**, **PUT**, and **DELETE**) will be tested.
* Testing will cover positive and negative scenarios executed via an automated framework built with Java and RestAssured**.**
* Demonstrate the ability to design a maintainable, reusable, and scalable API test automation framework.
* Test reports will be generated.

# TIMELINE

|  |  |  |  |
| --- | --- | --- | --- |
| **Phase** | **Start Date** | **End Date** | **Description** |
| Prepare test plan | 13-Aug-2025 | 14-Aug-2025 | Prepare test plan base on requirement |
| Build APIs framework | 14-Aug-2025 | 15-Aug-2025 | Design & build APIs framework |

# TEST ENVIRONMENT

* Test: <https://reqres.in/api>
* Stage: <https://reqres.in/api>
* API version: v1

# TEST STRATEGY

## Scope

### In-Scope

* Perform manual & automation testing.
* APIs functional testing will be performed on the below.

|  |  |
| --- | --- |
| **APIs** | **Scope of testing** |
| **POST /api/users** – Creates a new user | Creates user with valid data/with missing or invalid fields. |
| **POST /api/register**– Registers a user | Registers a user with defined user/undefined user |
| **GET /api/users/{id}** – Fetches a user | Fetches a user with valid data/with non-existing ID or invalid format. |
| **PUT /api/users/{id}** – Updates a user | Updates existing user with valid data/with invalid data or non-existing ID. |
| **DELETE /api/users/{id}** – Deletes a user | Deletes existing user/with invalid/non-existing ID. |
| **POST /api/login** – Creates a session | Login existing user/non-existing user/with user just registered. |

### Out Of Scope

* UI testing
* Load test
* Security Test
* List of APIs are not mentioned in the test plan

## Approach

* Testing will be conducted using an automated API test framework in Java with RestAssured.
* Validate API responses (status code, data in response) against expected results for both positive and negative cases.
* Execute workflow scenarios combining multiple endpoints:
  + Login > Create user > Update user information > Delete user
  + Register user > Login > Retrieve user information
* The framework should show the test result in HTML report.

## Test Level

Sytem Testing/API Functionals.

# TEST SCENARIOS

|  |  |  |
| --- | --- | --- |
| **Scenario ID** | **Module / Feature** | **Description** |
| TS01 | Login | * Test case: Login – Success * Test case: Login with missing providing password * Test case: Login with invalid email |
| TS02 | Register | * Test case: Register User – Success * Test case: Register User with missing password * Test case: Register User with a user not defined |
| TS03 | Retrieve user information | * Test case: Retrieve a list of users * Test case: Retrieve user data * Test case: Retrieve user with providing invalid user id |
| TS04 | Update user information | * Test case: Updated user information * Test case: Update information and expect invalid status code * Test case: Update user information with missing api key |
| TS05 | Delete user | * Test case: Delete user * Test case: Delete user and expected invalid code * Test case: Delete user - missing user id |
| TS06 | End to end | * Test case: Login, Create, Update & Delete User * Test case: Register user, login with that user and retrieve information |

# TEST DATA/PAYLOAD

Add this api key into request header [*x-api-key: reqres-free-v1*]

|  |  |
| --- | --- |
| APIs | Test data/payload |
| **POST - /api/users** | {  "name": "morpheus",  "job": "leader"  } |
| **PUT - /api/users/{ID}** | {  "name": "morpheus",  "job": "zion resident"  } |
| **POST - /api/register** | {  "email": "eve.holt@reqres.in",  "password": "pistol"  } |
| **POST - /api/login** | {  "email": "eve.holt@reqres.in",  "password": "cityslicka"  } |

# ENTRY CRITERIA

REQRES APIs (https://reqres.in/api-docs/) is available and accessible.

Selected endpoints (one POST, GET, PUT, DELETE, plus login).

Test data is prepared.

Java + RestAssured automation framework is set up and build.

Test environments (Test/Stage) are configured.

# EXIT CRITERIA

* All planned API tests have been executed.
* Test reports have been generated.
* Test plan is documented with source code share on github.
* Framework should be reusable, scalable and easy to maintain.
* Framework runs successfully in both Test and Stage environments.

# TEST TOOLS

* Language: Java
* Test tool / Library: Postman, RestAssured, TestNG
* Report: Allure Report