# Test Plan for Reqres API Testing

### 1. Introduction

The purpose of this test plan is to ensure the functionality, reliability, and security of the Reqres API. The API endpoints cover user registration, resource retrieval, and user-related operations.

### 2. Test Objectives

The objectives of testing the Reqres API are as follows:

* Validate successful user registration.
* Confirm accurate retrieval of resource data.
* Ensure proper handling of user-related API operations (GET, POST, PUT, PATCH, DELETE).

### 3. Scope

The test scope includes the following:

#### **3.1 In-Scope**

* User Registration
* Resource Retrieval
* User Operations (GET, POST, PUT, PATCH, DELETE)
* Data Validity

#### **3.2 Out-of-Scope**

* Authentication
* Load and Performance Testing

### 4. Test Types

#### **4.1 Functional Testing**

User Endpoints Testing:

* Retrieve data for all user
* Retrieve data for a single area.
* Verify error handling for invalid or missing authentication tokens.

Competition Endpoints Testing:

* Retrieve data for all competitions.
* Retrieve data for a single competition.
* Verify error handling for authorization issues.

Match Endpoints Testing:

* Retrieve match data for all competitions.
* Verify error handling for various scenarios in match data retrieval.

### 5. Test Environment

* API Base URL: <https://reqres.in/api>
* Testing Framework: pytest
* Programming Language: Python
* External Libraries: requests (for making HTTP requests)
* Version Control: Git

### 6. Test Data

* Utilize valid and invalid data for registration and user operations.
* Verify the existence of resources for resource retrieval.

.

### 7. Test Execution

**Preconditions:**

* Ensure the Reqres API is accessible.
* Verify the correctness of the provided scripts.
* Execute tests using the pytest framework.
* Use automation for repetitive and complex test scenarios.
* Run the user registration script.
* Run the resource retrieval script.
* Run the user operations script.

### 8. Defect Management

Log any identified defects using a standardized format.

Prioritize and categorize defects based on severity and impact.

### 9. Test Reporting

Generate and share test reports after each test script execution.

Include details such as test cases executed, pass/fail status, and any identified defects.

### 10. Test Deliverables

* Test scripts for functional, security, performance, and compatibility testing.
* Test reports with detailed results, issues, and recommendations.

### 11. Test Schedule

* Unit Testing: Verify individual API endpoints and functions.
* Integration Testing: Validate the interactions between different API components.
* System Testing: Comprehensive testing of the entire system.

### 12. Risks and Assumptions

* Assumption: The API documentation accurately reflects the expected behaviour.
* Risk: Changes to the API endpoints during testing may impact test cases.

**13. Sign-Off Criteria**

* All test cases executed.
* Test reports reviewed and approved.
* No critical defects outstanding.

**14. Review and Update**

Regularly review and update the test plan to reflect any changes in the API or testing requirements.