Reqres API Test Plan

Author: Paul Kent  
Version: 1.0

Table of Contents

[1. Introduction 2](#_Toc72409786)

[1.1 Scope 2](#_Toc72409787)

[1.2 Roles and Responsibilities 2](#_Toc72409788)

[2. Test Methodology 2](#_Toc72409789)

[2.1 Tooling 2](#_Toc72409790)

[2.2 Testing single API endpoints 3](#_Toc72409791)

[2.3 End-to-End (E2E) Scenarios 3](#_Toc72409792)

[3. Test Deliverables 3](#_Toc72409793)

[4. Environment 3](#_Toc72409794)

# Introduction

The purpose of the API testing is to create a basic test suite for the purpose of verifying that at least 4 endpoints of the Reqres API work as expected as part of a technical test.

The Reqres API hosted in the cloud provide publicly accessible endpoints that allow simulation of basic CRUD operations that return basic responses. It is noted that data is sent to the endpoints is not persisted in the cloud service.

## Scope

The following endpoints will be directly tested:

GET single resource

POST create resource

PUT update resource

DELETE resource

The Login endpoint will be called to extract a token and be used in subsequent requests as a Bearer Token. This endpoint is tested indirectly as a positive test case.

## Roles and Responsibilities

The responsibility of creation, execution and presentation of the results is the responsibility of the author of this document.

## Test Methodology

The endpoints to be tested as per the scope of this document will be called based on the documentation provided on the API usage. Data used to test the API will mostly be simple JSON objects created within the tests and verification of status codes and expected returned data. Two End to End (E2E) tests will be written to test multiple requests simulating basic E2E use cases.

Considering that the Reqres API is a simulation API and is not intended to be used as a business application, some behaviour which would not normally be valid in a business application will be treated as expected behaviour. Examples of this are as follows:

* The Reqres API is designed to allow endpoints to work without an authentication process and this is accepted as the expected behaviour for this API. The tests to be developed will however call the Login endpoint to obtain a token to be used as a Bearer token to simulate a more realistic test case.
* The Delete endpoint will return successful even if an invalid resource Id is provided
* Resources are not persisted in the backend of the service and hence tests will not be written to expect persistence. (A useful test case might be to test that there really is no persistence occurring, but this is not done in the scope of this project). Resources are posted to the API will simply returned to the consuming API and what is returned will be inspected in the test.

### Tooling

Postman will be used to as the Test Framework which provides the capability to create and organise requests, allows execution buy running on different environments (Test/Stage), and provides reports of the test execution.

### Testing single API endpoints

The endpoints listed in the scope of this document will be called individually passing in valid parameters where the API is expected to succeed. Response code validation and checking of properties in the returned JSON objects will be done to ensure the APIs are successful.

### End-to-End (E2E) Scenarios

There are two E2E test cases that will be created.

|  |  |
| --- | --- |
| **E2E Test Case 1**  **Test Steps - Document creation process** | **Expected Behaviour** |
| User logs in | Token is returned from service |
| The API gets the details of the user | Fields related to the user object is return from the service |
| The user makes an update to the document | The status code and response data returned from the service shows that an update has occurred |

|  |  |
| --- | --- |
| **E2E Test Case 2**  **Test Steps - Document finding and deletion process** | **Expected Behaviour** |
| User logs in | Token is returned in the response from service |
| Gets a list of documents with Ids | A list of documents is retrieved as valid JSON |
| Delete one of the documents using one of the Ids | The service should return successful |

# Test Deliverables

The main deliverable for this project is a Postman collection containing the tests developed. The collection will be created via the export function of the Postman tool. In addition to this the results generated from the execution of the created tests will be exported and placed on a GitHub repository.

# Environment

The environment that will be used for creating and executing the tests is: <https://reqres.in>