|  |
| --- |
| Photo displaying partial image of two pie charts on a canvas-textured page |
| **Test Automation Plan**  API (https://api.genderize.io?name=tibor) |
| |  |  |  | | --- | --- | --- | | Tibor Weigand | 5/26/24 | C# exercise | |

# Table of Contents

[Table of Contents 1](#_Toc167659862)

[**1. Introduction** 2](#_Toc167659863)

[1.1 Purpose and Objectives 2](#_Toc167659864)

[1.2 Scope 2](#_Toc167659865)

[**2. Requirements and Test Cases** 2](#_Toc167659866)

[**3. Test Automation Strategy** 2](#_Toc167659867)

[3.1 Tools and Technologies 2](#_Toc167659868)

[**4. Test Case Design** 2](#_Toc167659869)

[4.1 Test Case: Validate Status Code 2](#_Toc167659870)

[4.2 Test Case: Validate Response Structure 3](#_Toc167659871)

[**5. Test Execution Plan** 3](#_Toc167659872)

[5.1 Schedule 3](#_Toc167659873)

[5.2 Environment 3](#_Toc167659874)

[**6. Resources** 3](#_Toc167659875)

[**7. Risk Management** 4](#_Toc167659876)

[7.1 Risk Management Strategies 4](#_Toc167659877)

[**8. Test Maintenance** 5](#_Toc167659878)

[**9. Approvals** 5](#_Toc167659879)

## **1. Introduction**

### 1.1 Purpose and Objectives

The purpose of this test automation plan is to outline the strategy, approach, resources, and schedule for the automation of test cases for the API Endpoint “[Genderize](https://api.genderize.io/?name=tibor)”. The primary objective is to ensure that the API returns a valid status code and related response structure is thoroughly tested to verify compliance with the specified requirements.

### 1.2 Scope

This plan covers the automation of test cases related to the status code, as well as the expected response structure. Specifically, it addresses the requirements as listed below, ensuring that the API behaves as expected when the test cases are run from Postman.

## **2. Requirements and Test Cases**

|  |  |
| --- | --- |
| Requirement Mark | Requirement Description |
| TC1-R1 | The API should return a status code of 200 |
| TC2-R1 | The response should contain a “name” field with value "tibor" |
| TC2-R2 | The response should contain a “gender” field with value of gender prediction |
| TC2-R3 | The response should contain a “probability” field with value of certainty |
| TC2-R4 | The response should contain a “count” field with value of random integer |

## **3. Test Automation Strategy**

### 3.1 Tools and Technologies

* **Language**: JavaScript
* **Tool**: Postman

## **4. Test Case Design**

### 4.1 Test Case: Validate Status Code

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Requirement Mark | Test Case Description | Preconditions | Steps | Expected Result |
| TC1 | Validate that the status code is 200 | None | 1. Send a GET request to https https://api.genderize.io/?name=tibor. | The response status code should be 200 |

### 4.2 Test Case: Validate Response Structure

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Requirement Mark | Test Case Description | Preconditions | Steps | Expected Result |
| TC2 | Validate the structure of the response | None | 1. Send a GET request to https://dog.ceo/api/breeds/image/random. | 1. The response should contain a name field with value "tibor".  2. The response should contain a “gender” field with value of prediction  3. The response should contain a “probability” field with value of certainty  4. The response should contain a “count” field with value of random integer. |

## **5. Test Execution Plan**

### 5.1 Schedule

* **Test Development**: May 23 – May 26
* **Test Execution**: May 26
* **Report Generation**: May 28

### 5.2 Environment

* **Tool**: Postman
* **Test Environment**: Local machines or CI environment (e.g., Jenkins)

## **6. Resources**

* **Personnel**: 1 Test Automation Engineer
* **Hardware**: Laptops/desktops for development and execution
* **Software**: Postman, Notepad++, Jenkins (optional)

## **7. Risk Management**

Risk management involves identifying potential risks that could affect the testing process or the successful completion of the project. The following table outlines the potential risks, their impacts, likelihoods, and mitigation strategies.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Risk ID | Risk Description | Impact Level | Likelihood | Mitigation Strategy |
| R1 | API Endpoint is Unreachable | High | Medium | Verify network connectivity. Ensure the API server is up and running. Set up monitoring and alerts for API downtime. |
| R2 | API Response Format Changes | Medium | Low | Regularly review API documentation for updates. Implement flexible test scripts that can handle minor changes in response format. |
| R3 | High Volume of Requests Leading to Rate Limiting | Medium | Medium | Implement rate limiting in test scripts. Schedule tests during off-peak hours to avoid hitting rate limits. |
| R4 | Inaccurate Test Data | Medium | Low | Validate test data periodically. Use multiple data sources to ensure accuracy and consistency. |
| R5 | Postman Tool Issues (e.g., bugs, crashes) | Medium | Low | Keep Postman updated to the latest version. Regularly backup Postman collections. Use alternative tools if necessary. |
| R6 | GitHub Repository Access Issues (e.g., permission problems) | Low | Low | Ensure proper access permissions are set. Regularly back up repository data. Use GitHub support if access issues arise. |

### Risk Management Strategies

* **Proactive Monitoring:** Set up alerts and monitoring tools to detect when the API is down or experiencing issues.
* **Regular Updates and Reviews:** Regularly update test cases and review API documentation to stay informed about any changes.
* **Flexible Test Scripts:** Write test scripts that can handle minor changes in the API response structure without failing.
* **Backup Plans:** Maintain backups of your Postman collections and GitHub repository to prevent data loss.
* **Collaboration and Communication:** Maintain clear communication with the API development team to get advance notice of changes and issues.

## **8. Test Maintenance**

* **Regular Review**: Test scripts will be reviewed bi-weekly to ensure they are up to date with application changes.
* **Version Control**: Test scripts will be maintained in a version control system (e.g., Git).

## **9. Approvals**

|  |  |  |  |
| --- | --- | --- | --- |
| Role | Name | Signature | Date |
| Test Lead |  |  |  |
| Project Manager |  |  |  |
| QA Manager |  |  |  |