
Steam API

Software Test Plan

Document Change History

Version Number	Date	Contributor	Description
V1.0	09.03.2024	Jameel Nassar	First Document Writing

Table of Contents

1 Introduction	3
1.1 Overview	3
1.2 Scope.....	3
1.2.1 In Scope.....	3
1.2.2 Out of Scope	3
1.3 Quality Objective	3
1.3.1 Primary Objective	3
1.3.2 Secondary Objective	3
2 Test Tree.....	4
2.1 Functional Tests.....	4
2.2 Non-Functional Tests	4
3 Entry and Exit Criteria	5
3.1 Entry Criteria.....	5
3.2 Exit Criteria	5
4 Roles and Responsibilities	5
5 Test Methodology	5
5.1 Validation and Defect Management	5
5.2 Defect Tracking & Reporting.....	5
5.3 Status of the Bug/Fault	5
5.4 Test Management Process	6
6 Test Artifacts	6

1 Introduction

This test approach document outlines strategies, processes, workflows, and methodologies for testing the Steam API. The Steam API serves as a comprehensive platform for accessing various functionalities and data related to Steam, a digital distribution platform for video games.

1.1 Overview

The Steam API provides developers with access to a wide range of features, including retrieving app details, searching the store, managing reviews, and handling currency conversions. This test plan aims to ensure the reliability, functionality, and security of these API endpoints.

1.2 Scope

The scope of testing for the Steam API includes functional testing, performance testing, security testing, and usability testing. It covers various aspects such as app details retrieval, store search, review management, and currency handling.

1.2.1 In Scope

- Functional testing of all API endpoints, including retrieving app details, searching the store, managing reviews, and handling currency conversions.
- Performance testing to evaluate the API's response time and throughput under various load conditions.
- Security testing to identify and address potential vulnerabilities in the API.
- Usability testing to ensure the API is intuitive and user-friendly for developers integrating it into their applications.

1.2.2 Out of Scope

- Testing of client-side applications or user interfaces that interact with the Steam API.
- Testing of business standard operating procedures (SOPs) or internal processes of Steam.

1.3 Quality Objective

1.3.1 Primary Objective

The primary objective of testing the Steam API is to ensure it meets all specified requirements, including functional and non-functional requirements. This involves verifying that the API functions as expected, performs efficiently, and is secure.

1.3.2 Secondary Objective

The secondary objective is to identify and address any issues or risks within the API to enhance its reliability, performance, and security. This includes thorough testing to uncover potential bugs, vulnerabilities, or usability issues.

2 Test Tree

2.1 Functional Tests

Functional tests validate the behavior of various endpoints provided by the Steam API. These tests cover functionalities such as retrieving app details, searching the store, managing reviews, and handling currency conversions.

- **Test_get_app_name:** Retrieve the name of an application using the Steam API.
- **Test_store_search:** Search for applications or items within the Steam store.
- **Test_get_app_details:** Retrieve general app details.
- **Test_get_app_review:** Retrieve reviews for a specific application.
- **Test_filtering_review_by_max_play_time:** Filter reviews based on the maximum playtime.
- **Test_filtering_review_range_of_play_time:** Filter reviews based on a range of playtime.
- **Test_filtering_review_by_min_play_time:** Filter reviews based on the minimum playtime.
- **Test_default_app_review_number:** Verify the default number of reviews displayed for an application.
- **Test_changing_default_app_review_number:** Change the default number of reviews displayed for an application.
- **Test_advanced_search:** Test the advanced search functionality provided by the Steam API.
- **Test_store_search_found_apps:** Verify that search results return found applications.
- **Test_sorting_by_DESC_price_advanced_search:** Sort search results in descending order based on price.
- **Test_sorting_by_ACS_price_advanced_search:** Sort search results in ascending order based on price.

2.2 Non-Functional Tests

Non-functional tests focus on aspects such as performance, compatibility, localization, globalization, and recovery of the Steam API.

- **Test_get_app_details_currency_us:** Retrieve detailed information about an application with pricing in US dollars.
- **Test_get_app_details_currency_IL:** Retrieve app details with pricing in Israeli Shekel.
- **Test_get_app_details_currency_uk:** Retrieve app details with pricing in UK pounds.
- **Test_changing_app_review_language:** Change the language used for application reviews.

3 Entry and Exit Criteria

3.1 Entry Criteria

- Completion of API endpoint development.
- Availability of API documentation.
- Test environment setup.
- Test plan readiness.

3.2 Exit Criteria

- No critical or high priority test failures.
- All functional and non-functional requirements met.
- Approval from relevant stakeholders obtained.
- Resolution of all identified defects.

4 Roles and Responsibilities

During the testing phase of the Steam API, various roles and responsibilities are assigned to ensure effective execution and management of the testing process.

- **QA Testers:** Responsible for executing test cases and reporting defects.
- **Test Manager:** Oversees testing efforts and ensures adherence to timelines and quality standards.
- **Test Lead:** Plans and executes testing activities, analyzes requirements, and presents test results.

5 Test Methodology

5.1 Validation and Defect Management

Defects found during testing will be categorized based on severity and priority using a bug tracking tool. The status of each defect will be tracked, and resolutions will be documented.

5.2 Defect Tracking & Reporting

Defects will be tracked using a bug tracking tool, with statuses including New, Open, Rejected, Fixed, Closed, and Re-Open.

5.3 Status of the Bug/Fault

Each defect will have a status indicating its current state in the resolution process.

5.4 Test Management Process

Various tools will be utilized for test management, including test case design tools and bug tracking tools.

6 Test Artifacts

Key test artifacts produced during the testing lifecycle include the Test Plan, Requirement Traceability Matrix, Test Cases, Test Scripts, Test Environment Configuration, Test Results, and Software Test Report.