TEST PLAN – iTunes API

# Table of Contents

## 1. INTRODUCTION

## 2. QUALITY OBJECTIVES

### • 2.1 Objectives

## 3. SCOPE

### • 3.1 Functions included in testing

### • 3.2 Not included in testing

## 4. ENTRY AND EXIT CRITERIA

### • 4.1 Entry Criteria

### • 4.2 Exit Criteria

## 5. SUSPENSION CRITERIA AND RESUMPTION REQUIREMENTS

### • 5.1 Suspension criteria

### • 5.2 Resumption criteria

## 6. TEST STRATEGY AND APPROACH

### • 6.1 QA role in test process

### • 6.2 Bug entry template to be used

### • 6.3 Types of testing to be performed.

### • 6.4 Testing process

## 7.RESOURCES AND ENVIRONMEN'T NEEDS

### 7.1 Testing Tools

### 7.2 Environment

1. INTRODUCTION

The purpose of Test Plan is to facilitate communication within the team members. This document describes approaches and methodologies that will be applied to the functional testing of Apple iTunes APIs (Search and lookup). This document has clearly identified what the test deliverables will be, and what is deemed in and out of scope.

1. QUALITY OBJECTIVES

A major goal of testing in the Apple iTunes API is to fetch the matches based on the given list of inputs and verify the response returned as per validation/acceptance criteria.

* to ensure that the implementation is working correctly as expected — no bugs.
* to ensure that the implementation is working as specified according to the requirements specification.

2.1 Objectives

A primary objective of testing is to: assure that the system meets the full requirements, including quality requirements, satisfies the use case scenarios and maintain the quality of the project.

3. SCOPE

The Test Plan defines the unit, integration, system, regression, and Client Acceptance testing approach. The test scope includes the following:

* Testing of Search and lookup features functional, application performance, security.
* Quality requirements and search and lookup are testing.
* Test all examples provided for test coverage.

3.1 Functions to be tested with valid/invalid inputs for positive/negative scenarios.

Test Scenarios to be tested:

iTunes Search & Lookup API :

* Validate API is with secure https to access

iTunes Search API: Validate response and limit

* Validate Search API with API parameter term

and verify all search results contains 'Jack Johnson'

* Validate Search API with multiple API parameters term, limit

and verify search results limit with 25 with all response contains 'Jack Johnson'

* Validate Search API with multiple API parameters term, entity

and verify 'kind' is 'music-video'

* Validate Search API with multiple API parameters term, country

and verify 'country' contains 'CA'

* Validate Search API with multiple API parameters term, country, entity, software and verify 'wrapperType' is 'software'

iTunes Search API: Schema validation

* Validate mandatory fields of Search API with API parameter term
* Validate mandatory fields of Search API with multiple API parameters term, limit
* Validate mandatory fields of Search API with multiple API parameters term, entity
* Validate mandatory fields of Search API with multiple API parameters term, country
* Validate mandatory fields of Search API with multiple API parameters term, country, software

iTunes Search API: Response Time validation

* Validate response time is less than 2seconds of Search API with API parameter term
* Validate response time is less than 2seconds of Search API with multiple API parameters term, limit
* Validate response time is less than 2seconds of Search API with multiple API parameters term, entity
* Validate response time is less than 2seconds of Search API with multiple API parameters term, country
* Validate response time is less than 2seconds of Search API with multiple API parameters term, country, entity
* Validate response time is less than 2seconds of Search API with multiple API parameters term, country, entity, media

iTunes Lookup API: response and field validations

* Validate Lookup API for iTunes artist ID
* Validate Lookup API for iTunes ID
* Validate Lookup API with AMG artist id
* Validate Lookup API for multiple artists with multiple AMG artist ids
* Validate all albums returned by Lookup API with id, entity
* Validate Top 5 albums returned by Lookup API for each artist with id, entity
* Validate 5 most recent songs returned by Lookup API for each artist with Artist id, limit, entity, sort
* Validate album or video returned by Lookup API with UPC
* Validate album returned by Lookup API with UPC including tracks on that album
* Validate album returned by Lookup API with AMG Album ID
* Validate Movie returned by Lookup API with videoed
* Validate book returned by Lookup API with isbn

iTunes Lookup API: Response Time

* Validate response time is less than 2seconds of Lookup API for iTunes artist ID
* Validate response time is less than 2seconds of Lookup API for iTunes ID
* Validate response time is less than 2seconds of Lookup API with AMG artist id
* Validate response time is less than 2seconds of Lookup API for multiple artists with multiple AMG artist ids
* Validate response time is less than 2seconds of all albums returned by Lookup API with id, entity
* Validate response time is less than 2seconds of Top 5 albums returned by Lookup API for each artist with id, entity
* Validate response time is less than 2seconds of 5 most recent songs returned by Lookup API for each artist with Artistid, limit, entity, sort
* Validate response time is less than 2seconds of album or video returned by Lookup API with UPC
* Validate response time is less than 2seconds of album returned by Lookup API with UPC including tracks on that album
* Validate response time is less than 2seconds of album returned by Lookup API with AMG Album ID
* Validate response time is less than 2seconds of Movie returned by Lookup API with videoed
* Validate response time is less than 2seconds of book returned by Lookup API with isbn

Note: All Validation (asserts) for each Test Scenarios in automation

3.2 Functions Not to be tested - TBD.

4. ENTRY AND EXIT CRITERIA

4.1 Entry Criteria

* All test hardware software requirements must be successfully installed, configured, and functioning Properly. QA environments should be up & running.
* All the necessary documentation, design, and requirements that will allow testers to test the functionality
* All the standard software tools including the IDE’s, testing tools must be successfully installed and functioning properly.
* Complete set of test cases with test steps, test data and expected results are available to perform testing for needed functionality.
* Proper test data should be available.
* QA team should setup testing tool with testing automation scripts with correct configuration
* The test environment such as lab, hardware, software, and system administration support should be ready and available.
* QA team have completely understood the requirements

4.2 Exit Criteria

* Minimum Agreed °/o testcase coverage has been achieved.
* Regression testing completed successfully after bug fixes
* No high priority or severe bugs are left outstanding.
* All high-risk areas have been fully tested, with only minor residual risks left outstanding.
* Cost - when the budget has been spent.
* Deliverables and schedules have been met.

5. SUSPENSION CRITERIA AND RESUMPTION REQUIREMENTS

5.1 Suspension criteria

* The build contains several serious defects which seriously limit testing progress.
* Significant change(s) in requirements are suggested by team.
* Software/Hardware problems.
* Assigned resources are not available when needed by the test team.
* blockers identified during testing

5.2 Resumption criteria

* Resumption will only occur when the problem(s) that caused the suspension have been resolved

6. TEST STRATEGY AND APPROACH

6.1 QA role in test process

6.1.1 Understanding Requirements:

* Requirement specifications will be made available to QA Lead and team.
* Understanding of requirements will be done by QA in the form of written acceptance criteria (AC).

6.1.2 Preparing Test Cases linked to the requirements on JIRA for Traceability Matrix

(RTM):

* QA will be preparing test cases based on Requirements and AC. This will cover all the scenarios for requirements using test design section in JIRA.
* Preparing Requirements Traceability Matrix (RTM): To ensure coverage for the requirements QA will be preparing RTM which will map test scenarios to respective requirements.

6.1.3 Creating Test Data:

* Test data will be made available on test environments.

6.1.4 Executing Test Cases/Scenarios:

* Test cases will be executed by QA team in QA environment based on designed
* scenarios, test cases and test data.
* Test result (Actual Result, Pass/ Fail) will be updated in the test case execution tasks that will be linked to the respective ticket(s) in bug tracking tool.

6.1.5 Retesting the fixed bugs:

* Retesting fixed bugs will be done by QA team once they are resolved by respective developer, bug/defect status will be updated accordingly.
* Smoke and regression testing will be performed if required.

6.1.7 QA team involved in sprint planning meetings and stand ups along with another project meetings that required QA involvement. Presenting daily and weekly QA status.

6.1.8 Deployment/Delivery should meet criteria as planned.

6.2 Standard Bug entry template to be used to raise the defects.

6.2.1 Bug should be raised along with Severity and Priority Definition

* QA entering a bug into defect tracking tool is responsible for entering the bug Severity.
* QA entering a bug into defect tracking tool is responsible for entering bug priority. Priority is subject for review by QA Lead/Manager.

6.3 Types of testing to be performed.

* Functional Testing to ensure that the requirements are properly satisfied by the
* application. Using both manual and automated testing techniques.
* API testing to ensure the correct response provided when request made
* Regression Testing to ensure that changes (enhancements or defect fixes) to the software have not adversely affected it
* Smoke testing is performed to determine if a new software build is ready for the next testing phase
* Any other testing as per project requirement.

6.4 Testing process.

* Manual testing to ensure functionality developed is working as expected.
* Automated functional testing in sprint.
* Automated regression and Smoke test
* User Acceptance Testing (UAT) where a system will be tested for acceptability.
* etc.

6.4.1 Reviewing Functional Requirement

During the functional requirements review, the QA team will define an approach for testing both manual and automated.

* QA Team will determine the scope of testing what functions needs to automate and
* function can be tested manually.
* QA Team will determine the scope of testing what functions needs to automate, and function can be tested manually
* QA Lead/Manager will have to determine when and what type of testing should occur
* These requirements will be clearly outlined for testing by the BA/Stakeholder.

6.4.2 Creating Acceptance Criteria!

Once requirements are clear, the QA team will begin writing Acceptance into PM tool board user stories.

7.RESOURCES AND ENVIRONMEN'T NEEDS

7.1 Testing Tools

* IDE –IDE supports JAVA programming.
* Programming Language JAVA

7.2 Environment TBD