

Test Plan

Book Store API Testing

Date	Version	Change Log	Created by
01/08/2023	1	The document has been created	Rimsha Hassan

Contents

1. Introduction	3
1.1 Purpose	3
1.2 Scope	3
2. Objectives	3
3. Test Strategy	3
3.1 Test Levels	3
3.2 Types of Testing	3
3.3 Tools	4
4. Test Environment	4
5. Test Cases	4
5.1 Authentication	4
5.2 Book Creation	4
5.3 Book Retrieval	4
5.4 Book Update	4
5.5 Book Deletion	4
5.6 Pagination and Filtering	5
5.7 Error Handling	5
6. Schedule	5
7. Resources	5
8. Risks and Assumptions	5
9. Approval	5

1. Introduction

1.1 Purpose

The purpose of this test plan is to outline the testing strategy, objectives, resources, and schedule for testing the **Book Store API**. The goal is to ensure that all API endpoints function correctly and meet the defined requirements, providing a reliable and user-friendly experience for users interacting with the Book Store application.

1.2 Scope

This test plan covers the testing of the following functionalities within the Book Store API:

- Book retrieval
- Order creation, retrieval, updating, and deletion
- Authentication and Authorization
- Data validation and error handling
- Pagination and filtering of book records and orders

2. Objectives

- Verify that all API endpoints of the Book Store function as expected, returning the correct responses based on the inputs provided.
- Identify and document any defects, bugs, or issues within the API.
- Ensure that the API handles invalid inputs gracefully and returns appropriate error messages.
- Validate the performance and reliability of the API under expected load conditions.

3. Test Strategy

3.1 Test Levels

- **Integration Testing:** Testing of API endpoints in combination with the database and other services to ensure they work together as expected.
- **System Testing:** End-to-end testing of the entire API to validate overall functionality and performance.

3.2 Types of Testing

- **Functional Testing:**
 - Verify that each API endpoint performs the expected actions, such as creating, retrieving, updating, and deleting books.
 - Ensure that the API correctly handles authentication and authorization.
- **API Testing:**
 - Validate the API's behaviour using tools like Postman.
 - Test the API for response codes, response times, and data consistency.

- **Data Validation:**
 - Ensure that all required fields are present and correctly formatted in API requests and responses.
 - Test for constraints such as non-empty titles and valid ISBN numbers.
- **Error Handling:**
 - Test the API's response to invalid inputs, such as missing required fields or incorrect data types.
 - Verify that the API returns appropriate error codes and messages, such as 400 for bad requests and 401 for unauthorised access.
- **Regression Testing:**
 - Ensure that changes to the API do not introduce new bugs or break existing functionality.
- **Performance Testing** (if applicable):
 - Test the API's performance under expected and peak load conditions.

3.3 Tools

- **Postman:** For manual API testing and automation of test cases.
- **JIRA:** For tracking bugs and managing test cases.
- **Newman:** For running Postman collections in CI/CD pipelines (if automation is included).

4. Test Environment

- **Environment:** The API will be tested in a staging environment that closely resembles the production environment.
- **Test Data:** Test data will include sample book records with varying attributes (titles, authors, publication dates, etc.).

5. Test Cases

All these Test Cases are mentioned in the "Test Cases" document.

- Authentication
- Book Creation
- Book Retrieval
- Book Update
- Book Deletion
- Pagination and Filtering
- Error Handling

6. Schedule

- **Test Case Development:** [22/08/2024] - 22/08/2024]
- **Test Execution:** [23/08/2024] - [24/08/2024]
- **Final Report and Sign-off:** [25/08/2024]

7. Resources

- **QA Team:** Rimsha Hassan
- **Tools:** Postman, GitHub, Google Sheets, Google Docs

8. Risks and Assumptions

- **Risks:**
 - Limited access to the production environment may hinder accurate performance testing.
 - Unavailability of test data could delay testing.
- **Assumptions:**
 - The staging environment is a true replica of the production environment.
 - All necessary documentation and access rights are provided before testing begins.