**Test Plan (Order Rest API)**

Objective

Scope

Inclusions

[Test Environment.](#_bc378iay0wp9)

Defect Reporting Procedure

[Test](#_nwcehwadlsec) Strategy

[Test](#_uwmvj1gx5mlt) Schedule

Test Deliverables

Entry and Exit Criteria

Test Execution

Test Closure

**Objective**

The goal of this test plan is to ensure the quality, functionality and reliability of the Order Rest API hosted at <https://simple-books-api.glitch.me>

This API is designed to allow you to reserve a book.

**Scope**

The scope of the Test Plan for Order Rest API:

1. **Functional Testing:**

**-**Verify the correctness and functionality of all API endpoints as per API documentation.

- Test various scenarios for booking creation, modification, and cancellation.

- Validate user authentication and authorization mechanisms for protected endpoints.

**2. Data Validation Testing**:

- Ensure that the API correctly validates input data, rejecting invalid requests.

- Test boundary values for input fields to check for any unexpected behavior.

- Validate the accuracy of data returned in responses.

**3.** **Error Handling Testing**:

- Verify that appropriate error codes and messages are returned for invalid requests.

- Check error responses for sensitive information disclosure.

- Validate the API's ability to handle unexpected errors gracefully.

**4. Performance Testing**:

- Assess the API's response time under normal and peak loads to identify potential bottlenecks.

- Measure the API's throughput and scalability to handle concurrent requests.

**5. Integration Testing**:

- Verify interactions between different API endpoints and services.

- Test data consistency across related endpoints.

**6. Regression Testing:**

- Conduct regression testing after bug fixes or updates to ensure existing functionality remains intact.

**7. Edge Case Testing:**

- Test extreme and boundary scenarios to identify potential issues.

**8. Usability Testing:**

- Evaluate the API's user-friendliness and ease of use from a developer's perspective.

**9. Continuous Integration and Deployment (CI/CD) Testing:**

- Validate the API's behavior within the CI/CD pipeline to ensure smooth deployments.

**10. Documentation Review**:

- Assess the clarity, completeness, and accuracy of the API documentation.

- Verify that the API documentation is in sync with the actual API behavior.

**Inclusions**

**Read (GET) Operations:**

Test the API's ability to retrieve status book information by various criteria.

Verify that the API returns the correct data in response to read requests.

Test for correct handling of non-existent or invalid book order IDs.

**Create (POST) Operations:**

Test the API's ability to create an order using valid input data.

Verify that appropriate error responses are returned for invalid or missing data.

Validate that newly created orders are stored correctly in the system.

**Update (PUT) Operations:**

Test the API's ability to update existing book orders with valid data.

Verify that the API rejects invalid update requests with appropriate error responses.

Validate that the order data is correctly modified in the system after updates.

**Delete (DELETE) Operations:**

Test the API's ability to delete orders by providing valid order IDs.

Verify that the API returns appropriate responses after successful deletion.

Validate that the deleted orders are removed from the system.

**Boundary Testing:**

Test the API with minimum and maximum allowed values for input fields.

Validate the behavior of the API with values close to the boundaries.

**Authentication and Authorization:**

Test CRUD operations for both authenticated and unauthenticated users.

Verify that only authorized users can perform certain CRUD operations.

**Documentation Review:**

Assess the accuracy of API documentation related to CRUD operations.

**Usability Testing:**

Validate the CRUD operations within the CI/CD pipeline to ensure smooth deployments.

**Test Environments:**

The operating system and browser used for this API testing: Windows 10-Chrome

### **Defect Reporting Procedure**

The criteria for identifying a defect, such as deviation from the requirements, user experience issues, or technical errors.

-Tool: Jira

### **Test Strategy**

The first step is to create test scenarios and test cases for the various features in

Scope.

While developing test cases, we'll use a number of test design techniques.

-Equivalence Class Partition

-Boundary Value Analysis

### **Test Schedule**

Following is the test schedule planned for the project –

Task Time Duration:

▪ Creating Test Plan

▪ Test Case Creation

▪ Test Case Execution

▪ Summary Reports Submission Date

2 Sprints to Test the Application

### **Test Deliverables.**

-Test Plan

-Functional Test cases

-Defect Reports

-Summary Reports

### **Entry and Exit Criteria**

Entry Criteria:

-Once the testing team receives the Requirements Documents or details

about the Project

Exit Criteria:

-List of Requirements are explored and understood by the Testing team

- Doubts are cleared

### **Test Execution**

Entry Criteria:

-Application is ready for testing

Exit Criteria:

- Test Case Reports, Defect Reports are ready

### **Test Closure**

Entry Criteria:

-Test Case Reports, Defect Reports are ready

Exit Criteria:

-Test Summary Reports

#### **Tools**

The following are the list of Tools we will be using in this Project:

- JIRA Bug Tracking Tool

-Word and Excel documents

#### **Approvals**

Team will send different types of documents for Client Approval like below:

-Test Plan

-Test Scenarios

-Test Cases

–Reports

\*Testing will only continue to the next steps once these approvals are done.