# **Test Plan for Restful-Booker API**

**Test Plan Overview**

**Project Name:** Restful-Booker API Testing  
**Prepared by:** Ronish Shakya  
**Approved by:**  
**Date:** 4/4/2025

**Objective**

The purpose of this test plan is to define the scope, approach, resources, and schedule for testing the Restful-Booker API. The goal is to ensure that the API functions correctly, is secure, and meets business requirements while identifying and documenting potential bugs.

**Scope**

**In Scope (What will be tested?)**

* **Authentication:** Token Generation & Expiry, Unauthorized Access Handling
* **Booking Management:**
  + Create a new booking
  + Retrieve booking details
  + Update booking information
  + Delete a booking
* **Input Validation:** Handling of valid and invalid inputs
* **Concurrency Testing:** Simultaneous API requests handling
* **Error Handling:** API responses for invalid requests
* **Performance Testing:** Response time and request handling efficiency
* **Boundary Testing:** Edge cases for input data
* **Security Testing (Basic):** Unauthorized access attempts

**Out of Scope (What will not be tested?)**

* Database performance tuning
* API integration with third-party services
* Load and stress testing beyond manual testing scope
* Advanced security penetration testing

**Testing Approach**

Only **manual testing** will be performed using API testing tools such as Postman and Curl.

**Types of Testing**

* **Functional Testing:** Verify API endpoints perform expected CRUD operations
* **Error Handling Testing:** Ensure appropriate status codes and messages are returned
* **Security Testing:** Validate authentication and access control
* **Performance Testing:** Observe response times under normal conditions
* **Boundary Testing:** Test with extreme input values
* **Regression Testing:** Ensure bug fixes don’t break existing features

**Tools to be Used**

* **Postman** (Manual API Testing)
* **Curl** (Command-line API Testing)
* **JIRA** (Bug Tracking & Test Management)
* **Excel/Word** (Test Case Documentation)

**Test Environment**

* **OS:** Windows, macOS
* **API Testing Tools:** Postman, Curl
* **Test Data:** Dummy booking details, invalid input data sets
* **Environment:** Localhost/Test Server

**Test Deliverables**

* Test Plan Document
* Test Cases & Test Scenarios
* Defect Reports (Bug Reports)
* Test Execution Reports
* Test Summary Report

**Test Strategy**

**Component Descriptions**

|  |  |
| --- | --- |
| **Component** | **Description** |
| Objectives | Define the testing goals and scope for the API. |
| Test Levels | API functional testing at various levels: Unit, Integration, System. |
| Test Types | Functional, Security, Performance, and Regression Testing. |
| Test Techniques | Manual black-box testing approaches. |
| Test Deliverables | List of testing documents and reports. |
| Test Environment | The required tools and setup for testing. |
| Test Schedule | Time estimation for each testing phase. |
| Resource Allocation | Define roles and responsibilities. |
| Risk Management | Identify risks and contingency measures. |
| Test Exit Criteria | Define completion criteria for the testing phase. |

**Testing Steps**

**Step 1: Creating Test Scenarios and Test Cases**

We will develop test cases using the following test design techniques:

* Equivalence Class Partition
* Boundary Value Analysis
* Decision Table Testing
* Error Guessing

**Step 2: Testing Procedure**

1. Verify the API is accessible and running.
2. Conduct **Smoke Testing** to confirm key endpoints are working.
3. Perform **Authentication Testing**:
   * Attempt to access endpoints without authentication.
   * Generate a valid token and check expiry.
4. Execute **CRUD Testing**:
   * Create a new booking.
   * Retrieve an existing booking.
   * Update a booking.
   * Delete a booking.
5. Validate **Error Handling**:
   * Send invalid input data.
   * Check error messages and status codes.
6. Conduct **Performance Observations**:
   * Measure API response times manually.
   * Check handling of concurrent requests.
7. Perform **Regression Testing** to ensure no new defects arise.

**Entry and Exit Criteria**

**Entry Criteria:**

* API endpoints are deployed and available for testing.
* Test cases are reviewed and approved.
* Test environment is set up and stable.
* Required API documentation is available.

**Exit Criteria:**

* All major defects are identified, reported, and verified.
* 95% of test cases pass successfully.
* Basic security and performance validations are completed.
* Test summary report is approved.

**Risk and Mitigation**

|  |  |
| --- | --- |
| **Risk** | **Mitigation** |
| API downtime during testing | Coordinate with the development team for stable builds. |
| Unauthorized API access issues | Ensure proper authentication testing is conducted. |
| Response time degradation | Report slow API responses and perform basic performance validation. |
| Incomplete API documentation | Request detailed API documentation from developers. |

**Schedule and Timeline**

|  |  |  |
| --- | --- | --- |
| **Phase** | **Start Date** | **End Date** |
| Test Planning |  |  |
| Test Case Creation |  |  |
| Test Execution |  |  |
| Bug Fixing and Retesting |  |  |

**Approvals**

|  |  |  |
| --- | --- | --- |
| **Role** | **Name** | **Signature** |
| Test Manager |  |  |
| Project Manager |  |  |