# **Test Plan for Restful Booker API**

## **1. Objective**

This document outlines the test plan for the **Restful Booker API** application. The objective is to ensure that all booking-related functionalities work as expected, providing a seamless experience for users booking hotels.

## **2. Scope**

The scope of this test plan includes:

**Features to be tested:**

* Create Booking
* Update Booking
* Partial Update Booking
* Delete Booking
* Health Check (Ping)
* Authentication (Token Creation)
* Retrieve Booking IDs

**Types of testing:**

* Manual Testing
* Automated Testing
* Functional Testing
* API Testing
* Performance Testing
* Regression Testing

**Environments:**

* API Testing Tools: Postman, SoapUI
* Operating Systems: Windows 10, macOS, Linux
* Network Connectivity: Wi-Fi, cellular, wired connections

**Evaluation Criteria:**

* Number of defects found
* Time taken to complete testing
* API response times and correctness

## **3. Inclusions**

### **Introduction**

This test plan covers API endpoints for booking hotel rooms using the Restful Booker API. The goal is to ensure correctness, stability, and efficiency in the booking process.

### **Test Objectives**

* Validate that all API endpoints function correctly as per specifications.
* Identify defects in the booking workflow.
* Ensure API meets performance and response time requirements.

## **4. Exclusions**

* UI testing (since this is an API test plan)
* Security Testing

## **5. Test Environments**

**API Base URL:** https://restful-booker.herokuapp.com

### **Hardware/Software Requirements**

* API Testing Tools: Postman, JMeter, Newman
* Test Management Tools: JIRA, TestRail
* Authentication: Token-based authentication

## **6. Defect Reporting Procedure**

**Criteria for Identifying Defects:**

* HTTP status codes that do not match expected responses
* Incorrect data returned by API
* Performance issues (slow response time > 2s)

**Steps for Reporting Defects:**

* Capture API request & response
* Log the defect in JIRA/TestRail
* Assign severity levels (Critical, High, Medium, Low)

## **7. Test Strategy**

### **Step 1: Test Scenarios & Cases Creation**

#### **Techniques Used:**

* Equivalence Class Partitioning
* Boundary Value Analysis
* Exploratory Testing
* Positive & Negative Testing

### **Step 2: Testing Procedure**

* **Smoke Testing:** Ensure core API endpoints are functional.
* **Functional Testing:** Verify each API endpoint against expected results.
* **Regression Testing:** Ensure changes do not break existing functionalities.

### **Step 3: Best Practices**

* Context-Driven Testing
* Shift-Left Testing
* End-to-End API Workflows

## **8. Test Schedule**

| **Task** | **Duration** |
| --- | --- |
| Test Plan Creation | 2 days |
| Test Case Preparation | 3 days |
| API Testing Execution | 5 days |
| Bug Reporting & Fix Validation | Ongoing |
| Test Summary & Closure | 2 days |

## **9. Test Deliverables**

* Test Plan Document
* API Test Cases
* Bug Reports
* Test Summary Report

## **10. Entry and Exit Criteria**

### **Entry Criteria**

* API requirements are finalized.
* API endpoints are accessible.

### **Exit Criteria**

* All critical and major defects are resolved.
* Test summary report is completed.

## **11. Tools**

* **Postman** - API Testing
* **JMeter** - Performance Testing
* **JIRA** - Defect Tracking

## **12. Risks and Mitigations**

| Risk | Mitigation |
| --- | --- |
| API downtime | Test in a different environment or retry later |
| Incomplete documentation | Collaborate with developers for clarity |
| Performance issues | Use JMeter for load testing and optimize requests |

## **13. Approvals**

**Documents for Approval:**

* Test Plan
* Test Cases
* Test Summary Reports