



API TEST PLAN

RESTful Booker API Testing



FEBRUARY 2, 2025
DURGESH SAWANT
SQA ENGINEER PUNE, MAHARASHTRA

Table of Contents

- Objective
- Scope
- Test Coverage (Inclusions)
- Test Environment
- Defect Reporting Procedure
- Test Strategy
- Test Schedule
- Test Deliverables
- Entry and Exit Criteria
- Tools
- Risks and Mitigation Strategies
- Approvals

Test Plan for Restful Booker API Testing (AI Generator)

1. Objective The objective of this test plan is to ensure the quality, functionality, reliability, and performance of the **Restful Booker API** hosted at <https://restful-booker.herokuapp.com>. This API supports a fictional hotel booking system.

2. Scope This test plan covers the following testing areas:

- **Functional Testing:** Ensuring correct functionality of all API endpoints.
- **Data Validation Testing:** Validating input data and responses.
- **Error Handling Testing:** Checking proper error messages and status codes.
- **Performance Testing:** Evaluating response time under different loads.
- **Security Testing:** Assessing vulnerabilities and compliance with security best practices.
- **Integration Testing:** Validating interactions between API endpoints.
- **Regression Testing:** Ensuring bug fixes do not introduce new issues.

3. Test Coverage (Inclusions)

- **Authentication**
 - POST /auth - CreateToken
 - Validate successful token generation with correct credentials.
 - Ensure error handling for invalid credentials.
- **Booking Operations**
 - GET /booking - GetBookingIds
 - POST /booking - CreateBooking
 - PUT /booking/{id} - UpdateBooking
 - PATCH /booking/{id} - PartialUpdateBooking
 - DELETE /booking/{id} - DeleteBooking
 - Validate CRUD operations with different scenarios.
 - Ensure correct response codes and data integrity.
- **Health Check**
 - GET /ping - Verify API uptime.

4. Test Environments

- **QA URL:** <https://restful-booker.herokuapp.com/apidoc/index.html>
- **Pre-Prod URL:** <https://restful-booker.herokuapp.com/apidoc/index.html>
- **Platforms Tested:** Windows 10 (Chrome, Firefox, Edge), macOS (Safari), Android (Chrome), iPhone (Safari).

5. Defect Reporting Procedure

- Defects will be logged in **JIRA**.
- Each defect will include a description, reproduction steps, severity, and attachments if necessary.
- Defects will be triaged based on priority and assigned to the appropriate team members.

6. Test Strategy

- **Test Design Techniques:**
 - Equivalence Class Partitioning
 - Boundary Value Analysis
 - Decision Table Testing
 - State Transition Testing
 - Use Case Testing
- **Testing Types:**
 - Smoke and Sanity Testing
 - Functional Testing
 - Regression Testing
 - Security Testing (SQL Injection, XSS, etc.)
 - Performance and Load Testing
 - API Usability Testing

7. Test Schedule

Task	Duration
Test Plan Creation	2 days
Test Case Development	4 days

Test Execution 5 days

Defect Fixing and Retesting 3 days

Final Test Summary Report 2 days

8. Test Deliverables

- Test Plan Document
- Test Scenarios and Test Cases
- Test Execution Reports
- Defect Reports
- Test Summary Report

9. Entry and Exit Criteria

- **Entry Criteria:**
 - API documentation is available.
 - Stable test environment is set up.
 - Test data is prepared.
- **Exit Criteria:**
 - All critical defects are resolved.
 - Test execution is complete, and reports are submitted.

10. Tools

- **API Testing:** Postman
- **Defect Tracking:** JIRA
- **Documentation:** Word, Excel
- **Performance Testing:** JMeter

11. Risks and Mitigation

Risk	Mitigation Strategy
------	---------------------

Unstable API	Ensure a stable environment before testing.
--------------	---

Authentication Issues	Verify the token generation process and implement retry logic.
-----------------------	--

High Load Handling	Conduct stress testing to identify bottlenecks.
--------------------	---

12. Approvals The following approvals are required before execution:

- **Test Plan Approval:** Team Lead
- **Test Scenarios & Test Cases Approval:** QA Manager
- **Final Test Summary Report Approval:** Project Stakeholders

Filter... X

Auth

CreateToken

Booking

GetBookings

GetBooking

CreateBooking

UpdateBooking

PartialUpdateBooking

DeleteBooking

Ping

HealthCheck

Auth - CreateToken

1.0.0

Creates a new auth token to use for access to the PUT and DELETE /booking

POST

https://restful-booker.herokuapp.com/auth

Example 1:

```
curl -X POST \
  https://restful-booker.herokuapp.com/auth \
  -H 'Content-Type: application/json' \
  -d '{
    "username" : "admin",
    "password" : "password123"
  }'
```

Header

Field	Type	Description
Content-Type	string	Sets the format of payload you are sending Default value: <code>application/json</code>

Request body

Field	Type	Description
username	string	Username for authentication
password	string	Password for authentication