

Test Plan Document

Title: Test Plan for Restful Booker API Automation

Version: 1.0

Author: Kasuni Gamage

Date: 2025/10/17

1. Introduction

This document is created for the purpose of defining the test plan/strategy, objectives, scope, and approach for automating and validating the **Restful Booker API**. The focus is on ensuring correctness, reliability, and performance of booking-related endpoints in a cloud-native environment.

2. Objectives

- Validate functional correctness of GET, PATCH, and DELETE booking endpoints.
- Ensure backward compatibility and idempotency.
- Automate tests for regression and CI/CD integration.
- Measure response time performance (< 2 seconds).
- Identify and report defects, inconsistencies, and bottlenecks.

3. Scope

In Scope:

- GET /booking — including filters and invalid input handling
- PATCH /booking/{id} — partial updates and error validation
- DELETE /booking/{id} — deletion and authentication handling
- Authentication with POST /auth
- End-to-end life-cycle validation: Create → Update → Verify → Delete

Out of Scope:

- UI or front-end validation
- Load or stress testing beyond light performance checks
- Database verification (since API is public)

4. Test Approach

Framework:

Language: C#

Data Management: JSON files for parameterized testing

Reporting:

Types of Testing:

Type	Description
Functional	Validate each endpoint according to API spec
Negative	Test invalid requests and data
Integration	Verify complete booking life cycle
Regression	Run on each pipeline trigger
Performance	Validate API latency and response time

Test Design:

- Parameterized test cases to support multiple data combinations
- Fixtures for authentication token setup and tear down
- Request/response schema validation

5. Test Environment

Component	Description
Base URL	https://restful-booker.herokuapp.com
Auth Endpoint	/auth
Tools	Playwright, Nunit, RestSharp
CI/CD	GitHub Actions
OS	Windows

6. Test Data

- Test data stored in resources “\RestfulBookerTests\TestData\”
- Includes valid and invalid payloads for create, update, and delete tests
- Data-driven tests pull from JSON file.

7. Entry & Exit Criteria

Entry:

- API endpoints are stable and reachable.
- Authentication token generation is functional properly.

Exit:

- All high and medium priority test cases executed.
- No critical or blocker defects remain open.
- Test report and logs uploaded to CI pipeline.

8. Deliverables

- Automated test suite.
- Test plan and test report documents.
- Test execution logs and report
- README with setup and run instructions

9. Risks and Mitigation

Risk	Mitigation
API downtime	CI/CD pipeline to verify availability
Token expiry	Introducing Token Refresh Logic
Flaky tests	Retry logic

10. Schedule

Phase	Duration	Deliverable
Planing and Framework setup	Day 1	Test plan document , Implemented framework
Test case development	Day 2	Complete test suites
CI/CD setup + report	Day 3	CI/CD pipeline implementation and Test Report
Final submission	Day 3	Repository finalization

