RESTful Booker API – Test Plan

# 1. Overview

This test plan defines the strategy and scope for testing the RESTful Booker API available at https://restful-booker.herokuapp.com. This API simulates a booking system and is commonly used for learning and testing REST API automation.

# 2. Objectives

* To verify that all API endpoints function as expected.
* To validate API behavior with valid and invalid inputs.
* To ensure that CRUD operations work correctly.
* To check for proper authentication and authorization.
* To perform boundary, security, and negative testing.

# 3. API Endpoints Summary

|  |  |  |  |
| --- | --- | --- | --- |
| Endpoint | Method | Description | Auth Required |
| /ping | GET | Health check | No |
| /auth | POST | Create a token for authentication | No |
| /booking | GET | Get all bookings | No |
| /booking/:id | GET | Get booking by ID | No |
| /booking | POST | Create a new booking | No |
| /booking/:id | PUT | Update booking by ID | Yes |
| /booking/:id | PATCH | Partial update to booking | Yes |
| /booking/:id | DELETE | Delete booking by ID | Yes |

# 4. Test Scope

\*\*In-Scope:\*\*  
- Functional testing of all endpoints.  
- Security and authentication testing.  
- Positive and negative scenarios.  
- Data validation (data types, required fields).  
- Load/Performance basics (ping, bulk create, etc.).  
  
\*\*Out-of-Scope:\*\*  
- UI testing  
- Backend database validation (since no DB is exposed)

# 5. Types of Testing

* Smoke Testing – Check if basic operations (ping, auth, GET, POST) work.
* Functional Testing – Validate API responses with correct inputs.
* Negative Testing – Test with incorrect headers, body, method types.
* Security Testing – Test with/without valid tokens, invalid tokens.
* Boundary Testing – Create bookings with min/max input values.
* CRUD Testing – Ensure Create, Read, Update, Delete work as intended.

# 6. Test Environment

- Base URL: https://restful-booker.herokuapp.com  
- Tools: Postman / Newman, RestAssured, Pytest, JMeter  
- Authentication: Uses /auth endpoint to retrieve a token (cookie-based)

# 7. Test Data Requirements

|  |  |  |
| --- | --- | --- |
| Field Name | Data Type | Sample Value |
| firstname | string | Jim |
| lastname | string | Brown |
| totalprice | integer | 111 |
| depositpaid | boolean | true |
| bookingdates | object | {"checkin": "2024-01-01", "checkout": "2024-01-05"} |
| additionalneeds | string | Breakfast |

# 8. Test Scenarios

* Health Check – GET /ping → expect status 200 or 201.
* Authentication – POST /auth with valid and invalid credentials.
* Create Booking – POST /booking with valid/missing/invalid data.
* Get Booking – GET /booking/:id with valid/invalid ID, query filters.
* Update Booking – PUT/PATCH /booking/:id with valid/invalid token.
* Delete Booking – DELETE /booking/:id with and without auth.

# 9. Security Test Cases

* Access PUT/DELETE without token.
* Use expired/invalid tokens.
* Injection in fields (e.g., <script>).
* Invalid method types (e.g., DELETE on /booking list).

# 10. Performance (Optional)

* Run multiple POST requests and measure response time.
* Use JMeter for GET /booking with 50–100 virtual users.

# 11. Defect Tracking

* Log all bugs in JIRA, Bugzilla, or Excel.
* Each defect must include: steps to reproduce, expected vs actual, severity/priority.

# 12. Schedule

|  |  |
| --- | --- |
| Task | Timeline |
| API Review | Day 1 |
| Test Plan Creation | Day 2 |
| Test Case Design | Day 3-4 |
| Test Execution (Manual) | Day 5-6 |
| Automation (Optional) | Day 7-9 |
| Bug Fix Verification | Day 10 |

# 13. Test Entry/Exit Criteria

\*\*Entry Criteria:\*\*  
- API is accessible and stable.  
- Documentation is available.  
- Required tools are installed.  
  
\*\*Exit Criteria:\*\*  
- All critical and high priority test cases executed.  
- All major bugs fixed or deferred with sign-off.  
- Test report shared and approved.

# 14. Roles & Responsibilities

|  |  |
| --- | --- |
| Role | Responsibility |
| QA Lead | Test planning, reporting, sign-off |
| Testers | Design and execute test cases |
| Developers | Fix defects |
| Automation QA | Write and run API test scripts |

# 15. Deliverables

* Test Plan Document
* Test Case Sheet (CSV/Postman)
* Defect Reports
* Automation Scripts (if applicable)
* Final Test Summary Report