**TEST PLAN**

**Restful Booker API Testing Project**

**Document Version:** 1.0

**Date:**

**Project:** Restful Booker API Testing

**Prepared By:**

**1. INTRODUCTION**

The Restful Booker API is a web service that provides booking management functionality for a hotel or accommodation booking system. This test plan outlines the comprehensive testing strategy for validating all API endpoints, ensuring data integrity, security compliance, and proper error handling.

The API provides core functionality including user authentication, booking creation and management, booking retrieval and filtering, and booking updates and deletion. Testing will focus on functional validation, security verification, and ensuring the API meets business requirements and industry standards.

This document serves as the master plan for all testing activities related to the Restful Booker API project and will guide the testing team through systematic validation of all system components.

**2. OBJECTIVE**

**Primary Objectives:**

**Functional Validation:** Verify all API endpoints work according to specifications and business requirements

**Security Assurance:** Ensure authentication and authorization mechanisms prevent unauthorized access

**Data Integrity:** Validate that all data operations maintain consistency and accuracy

**Error Handling:** Confirm proper HTTP status codes and error messages for all scenarios

**Performance Baseline:** Establish response time benchmarks for all operations

**Secondary Objectives:**

**Documentation Accuracy:** Verify API behavior matches documented specifications

**Integration Readiness:** Ensure API is ready for client application integration

**Regression Prevention:** Identify and prevent defects from impacting production

**Quality Metrics:** Establish quality benchmarks for future releases

**Risk Mitigation:** Identify and address potential security and functional risks

**3. SCOPE**

**In Scope:**

**Authentication Module**

  Valid credential authentication

  Invalid credential handling

  Token generation and validation

  Missing field validation

**Booking Management**

  Create bookings with complete and partial data

  Retrieve individual and multiple bookings

  Update bookings (full and partial updates)

  Delete bookings with proper authorization

**Data Validation**

  Input data type validation

  Date format and business logic validation

  Required field validation

  Boundary value testing

**Filtering and Search**

  Filter by guest name (firstname/lastname)

  Filter by date ranges (checkin/checkout)

  Handle invalid filter parameters

**Security Testing**

  Authentication bypass attempts

  Authorization validation for protected operations

  Unauthorized access prevention

**Error Handling**

  HTTP status code validation

  Error message consistency

  Invalid request handling

**Out of Scope:**

Database performance optimization

Server infrastructure testing

Load and stress testing

User interface testing (API only)

Thirdparty service integrations

Network latency testing

Browser compatibility (not applicable)

**4. TEST APPROACH**

**Testing Strategy:**

**Black Box Testing:** Focus on inputoutput validation without internal code knowledge

**RiskBased Testing:** Prioritize highrisk areas like security and data integrity

**RequirementBased Testing:** Ensure all functional requirements are validated

**Testing Types:**

**Functional Testing**

**Positive Testing:** Valid inputs and expected workflows

**Negative Testing:** Invalid inputs and error scenarios

**Boundary Testing:** Edge cases and limit values

**Integration Testing:** Endtoend workflow validation

**Non-Functional Testing**

**Security Testing:** Authentication and authorization validation

**Usability Testing:** API ease of use and error message clarity

**Compatibility Testing:** Different data formats and request methods

**Testing Methods:**

**Manual Testing:** Exploratory testing and complex scenario validation

**Automated Testing:** Regression testing and repetitive test execution

**API Testing Tools:** Postman for request/response validation

**DataDriven Testing:** Multiple data sets for comprehensive coverage

**Test Design Techniques:**

**Equivalence Partitioning:** Group similar input conditions

**Boundary Value Analysis:** Test edge cases and limits

**Error Guessing:** Predict potential failure points

**State Transition Testing:** Validate booking status changes

**5. TEST DELIVERABLES**

**Test Planning Documents:**

**Test Plan Document** (this document)

**Test Strategy Document**

**Test Case Specifications** (CSV format)

**Requirements Traceability Matrix (RTM)**

**Test Execution Documents:**

**Test Execution Reports**

**Daily Test Status Reports**

**Defect Reports and Bug Logs**

**Test Coverage Reports**

**Test Automation Artifacts:**

**Postman Collection** with automated test scripts

**Newman Test Reports** (commandline execution)

**CI/CD Integration Scripts**

**Test Data Sets** and configuration files

**Final Deliverables:**

**Test Summary Report**

**Defect Summary and Analysis**

**Lessons Learned Document**

**Recommendations for Future Testing**

**Supporting Documents:**

**Test Environment Setup Guide**

**Test Data Management Plan**

**Risk Assessment Report**

**Signoff and Approval Documents**

**6. ENTRY & EXIT CRITERIA**

**Entry Criteria:**

**Environment Readiness**

  Test environment is set up and accessible

  API endpoints are deployed and functional

  Base URL and authentication credentials are available

  Network connectivity is established

**Documentation Availability**

  API documentation is complete and reviewed

  Business requirements are finalized

  Test cases are prepared and peerreviewed

  Test data is prepared and validated

**Resource Readiness**

  Testing team is assigned and available

  Testing tools (Postman, Newman) are configured

  Test environment access permissions are granted

  Defect tracking system is ready

**Prerequisites Met**

  Smoke testing is completed successfully

  Critical dependencies are resolved

  Test plan is approved by stakeholders

**Exit Criteria:**

**Test Execution Completion**

  All planned test cases are executed (100%)

  Critical and high priority test cases pass (100%)

  Medium priority test cases pass (95%)

  Low priority test cases pass (90%)

**Defect Resolution**

  All critical defects are fixed and verified

  All high priority defects are fixed and verified

  Medium priority defects are addressed or accepted

  No open critical or high priority defects remain

**Quality Gates Met**

  Test coverage meets defined criteria (95% for core features)

  Performance benchmarks are established

  Security validation is completed successfully

  Regression testing shows no new defects

**Documentation Complete**

  Test execution report is finalized

  Defect summary report is prepared

  Signoff is obtained from stakeholders

  Lessons learned are documented

**7. RISKS & MITIGATION**

**High Risk Areas:**

**Risk 1: Security Vulnerabilities**

**Risk Description:** Unauthorized access to booking operations or data breaches

**Impact:** Critical - Could compromise customer data and system integrity

**Probability:** Medium

**Mitigation Strategies:**

Prioritize all security-related test cases

Perform comprehensive authentication bypass testing

Validate all authorization mechanisms thoroughly

Conduct penetration testing for critical endpoints

**Risk 2: Data Integrity Issues**

**Risk Description:** Invalid data acceptance leading to corrupted bookings

**Impact:** High - Could result in business process failures

**Probability:** Medium

**Mitigation Strategies:**

Implement extensive data validation testing

Test all boundary conditions and edge cases

Validate business logic thoroughly

Create comprehensive negative test scenarios

**Risk 3: Environment Instability**

**Risk Description:** Test environment downtime or inconsistent behavior

**Impact:** Medium - Could delay testing schedule

**Probability:** High

**Mitigation Strategies:**

Establish environment monitoring procedures

Create backup testing environments

Implement automated health checks

Maintain communication with infrastructure team

**Risk 4: Incomplete Requirements**

**Risk Description:** Missing or unclear API specifications

**Impact:** Medium - Could lead to inadequate test coverage

**Probability:** Low

**Mitigation Strategies:**

Conduct thorough requirements review sessions

Maintain regular communication with development team

Document assumptions and get stakeholder confirmation

Implement exploratory testing to uncover gaps

**Medium Risk Areas:**

**Risk 5: Resource Availability**

**Risk Description:** Key team members unavailable during critical testing phases

**Impact:** Medium - Could impact testing timeline

**Probability:** Medium

**Mitigation Strategies:**

Crosstrain team members on critical testing areas

Maintain detailed documentation of test procedures

Plan for resource backup and coverage

Communicate resource needs early to management

**Risk 6: Tool Limitations**

**Risk Description:** Testing tools may not support all required test scenarios

**Impact:** Low - Could require manual testing alternatives

**Probability:** Low

**Mitigation Strategies:**

Evaluate tool capabilities during planning phase

Prepare manual testing alternatives

Consider additional tools if needed

Train team on multiple testing approaches

**8. ROLES & RESPONSIBILITIES**

**Test Manager**

**Primary Responsibilities:**

Overall test planning and strategy development

Resource allocation and team coordination

Stakeholder communication and reporting

Risk management and mitigation planning

Test schedule management and milestone tracking

Quality gate enforcement and signoff coordination

**Key Activities:**

Approve test plan and test cases

Monitor test execution progress

Escalate issues and risks to management

Ensure deliverable quality and completeness

**Senior Test Engineer / Test Lead**

**Primary Responsibilities:**

Test case design and review

Test execution coordination and oversight

Defect triage and priority assignment

Technical guidance to testing team

Test automation strategy and implementation

Test environment coordination

**Key Activities:**

Review and approve test cases

Lead test execution activities

Coordinate with development team for defect resolution

Mentor junior team members

**API Test Engineers (2 resources)**

**Primary Responsibilities:**

Test case execution and result documentation

Defect identification and detailed reporting

Test data preparation and management

Regression testing execution

Test script development and maintenance

**Key Activities:**

Execute functional and negative test cases

Document test results and evidence

Report and track defects through resolution

Participate in test case reviews

**Automation Engineer**

**Primary Responsibilities:**

Test automation framework development

Automated test script creation and maintenance

CI/CD pipeline integration

Performance baseline establishment

Tool evaluation and implementation

**Key Activities:**

Develop Postman collections with automated assertions

Set up Newman for command-line execution

Integrate tests with build pipeline

Generate automated test reports

**Business Analyst / Subject Matter Expert**

**Primary Responsibilities:**

Requirements clarification and validation

Business logic verification

Test case review from business perspective

User acceptance criteria definition

Stakeholder communication support

**Key Activities:**

Review test cases for business accuracy

Validate test scenarios against requirements

Provide domain expertise for complex scenarios

Support defect impact assessment

**9. TIMELINE**

**Phase 1: Test Planning and Preparation (3 days)**

**Duration:**

**Day 1:**

Test plan creation and review

Stakeholder review and feedback incorporation

Test plan approval and signoff

**Day 2:**

Test case development and peer review

Test data preparation and validation

Test environment setup and verification

**Day 3:**

Test case finalization and approval

Tool configuration and setup

Team training and knowledge transfer

**Phase 2: Test Execution - Round 1 (4 days)**

**Duration**

**Day 1:**

Smoke testing and environment validation

Authentication module testing (TC001TC005)

Initial defect identification and reporting

**Day 2:**

Booking creation module testing (TC006TC012)

Data validation and business logic testing

Critical defect escalation

**Day 3:**

Booking retrieval and filtering testing (TC013TC018)

Integration workflow testing

Defect verification and retesting

**Day 4:**

Booking update and deletion testing (TC019TC025)

Security testing completion

Round 1 test execution report

**Phase 3: Defect Resolution and Retesting (3 days)**

**Duration:**

**Day 1:**

Critical defect fixes deployment

High-priority defect retesting

Regression testing for fixed issues

**Day 2:**

Medium-priority defect retesting

Additional exploratory testing

Test coverage gap analysis

**Day 3:**

Final regression testing

Low-priority defect verification

Test completion validation

**Phase 4: Test Closure and Reporting (2 days)**

**Duration:**

**Day 1:**

Test summary report preparation

Defect analysis and metrics compilation

Lessons learned documentation

**Day 2:**

Final deliverable review and approval

Stakeholder presentation and signoff

Project closure and handover

**Key Milestones:**

**Test Plan Approval:**

**Test Execution Start:**

**Critical Defects Identified:**

**Round 1 Testing Complete:**

**Defect Retesting Complete:**

**Final Signoff:**

**Total Duration: 12 working days**

**Project Start:**

**Project End:**

**Document Control:**

**Version:** 1.0

**Last Updated:**