# **Test Plan (Restful Booker)**

## Objective

The goal of this test plan is to ensure the quality, functionality, and reliability of the Restful Booker API hosted at <https://restful-booker.herokuapp.com>.

The API is designed to handle booking requests for a fictional hotel booking system.

## Scope

Scope of Test Plan for Restful Booker API:

1. **Functional Testing:**

- Verify the correctness and functionality of all API endpoints as per the API documentation.

- Test various scenarios for booking creation, modification, and cancellation.

- Validate user authentication and authorization mechanisms for protected endpoints.

2. **Data Validation Testing**:

- Ensure that the API correctly validates input data, rejecting invalid requests.

- Test boundary values for input fields to check for any unexpected behavior.

- Validate the accuracy of data returned in responses.

3. **Error Handling Testing**:

- Verify that appropriate error codes and messages are returned for invalid requests.

- Check error responses for sensitive information disclosure.

- Validate the API's ability to handle unexpected errors gracefully.

4. **Documentation Review**:

- Assess the clarity, completeness, and accuracy of the API documentation.

- Verify that the API documentation is in sync with the actual API behavior.

5. **Usability Testing:**

- Evaluate the API's user-friendliness and ease of use from a developer's perspective.

6. **Continuous Integration and Deployment (CI/CD) Testing:**

- Validate the API's behavior within the CI/CD pipeline to ensure smooth deployments.

7. **Performance Monitoring:**

- Implement monitoring to track API performance in real-time.

It's important to note that the scope of the test plan may evolve during the testing process based on feedback, changing requirements, or discoveries during testing. The scope should be reviewed and adjusted accordingly throughout the testing phase to ensure comprehensive coverage of the Restful Booker API.

### **Inclusions**

1. **Create (POST) Operations:**

Test the API's ability to create new bookings using valid input data.

Verify that appropriate error responses are returned for invalid or missing data.

Validate that newly created bookings are stored correctly in the system.

1. **Read (GET) Operations:**

Test the API's ability to retrieve booking information by various criteria (e.g., booking ID, date range, guest name).

Verify that the API returns the correct data in response to read requests.

Test for correct handling of non-existent or invalid booking IDs.

1. **Update (PUT) Operations:**

Test the API's ability to update existing bookings with valid data.

Verify that the API rejects invalid update requests with appropriate error responses.

Validate that the booking data is correctly modified in the system after updates.

1. **Partial Update (PATCH) Operations:**

Test the API's ability to do a partial update for existing bookings with valid data.

Verify that the API rejects invalid update requests with appropriate error responses.

Validate that the booking data is correctly modified in the system after updates.

1. **Delete (DELETE) Operations:**

Test the API's ability to delete bookings by providing valid booking IDs.

Verify that the API returns appropriate responses after successful deletion.

Validate that the deleted bookings are removed from the system.

1. **Authentication and Authorization:**

Test CRUD operations for both authenticated and unauthenticated users.

Verify that only authorized users can perform certain CRUD operations.

1. **Error Handling:**

Test the API's response when invalid or malformed requests are made for CRUD operations.

Validate that appropriate error codes and messages are returned.