**RESTful-Booker API Demo Requirement Documentation**

**Introduction**

RESTful-Booker is a simple REST API that allows users to interact with a mock hotel booking system. It provides endpoints for creating, retrieving, updating, and deleting bookings, as well as authentication and booking management functionalities.

This document outlines the requirements for testing the RESTful-Booker API for portfolio demonstration.

**API Base URL**

The base URL for accessing the RESTful-Booker API is: https://restful-booker.herokuapp.com

**Authentication**

The API requires authentication for certain endpoints. It utilizes basic authentication, where the username is "admin" and the password is "password123".

**Endpoints**

1. Booking Endpoints

a. Get Booking

Endpoint: /booking/{bookingId}

Method: GET

Description: Retrieves details of a specific booking identified by its ID.

Parameters:

bookingId (integer): The unique identifier of the booking.

Response:

Status Code: 200 OK

Content: JSON object representing the booking details.

Example:

json

Copy code

{

"bookingid": 1,

"booking": {

"firstname": "John",

"lastname": "Doe",

"totalprice": 100,

"depositpaid": true,

"bookingdates": {

"checkin": "2024-04-07",

"checkout": "2024-04-10"

},

"additionalneeds": "Breakfast"

}

}

b. Create Booking

Endpoint: /booking

Method: POST

Description: Creates a new booking with the provided details.

Parameters:

Booking details (JSON object):

firstname (string): First name of the guest.

lastname (string): Last name of the guest.

totalprice (integer): Total price of the booking.

depositpaid (boolean): Indicates whether deposit is paid or not.

bookingdates (JSON object):

checkin (string): Check-in date in the format "YYYY-MM-DD".

checkout (string): Check-out date in the format "YYYY-MM-DD".

additionalneeds (string, optional): Any additional needs of the guest.

Response:

Status Code: 200 OK

Content: JSON object containing the ID of the newly created booking.

Example:

json

Copy code

{

"bookingid": 11

}

c. Update Booking

Endpoint: /booking/{bookingId}

Method: PUT

Description: Updates the details of an existing booking.

Parameters:

bookingId (integer): The unique identifier of the booking to be updated.

Updated booking details (JSON object):

Same parameters as in the Create Booking endpoint.

Response:

Status Code: 200 OK

Content: None

d. Delete Booking

Endpoint: /booking/{bookingId}

Method: DELETE

Description: Deletes the booking identified by its ID.

Parameters:

bookingId (integer): The unique identifier of the booking to be deleted.

Response:

Status Code: 201 Created

Content: None

**2. Auth Endpoints**

a. Create Token

Endpoint: /auth

Method: POST

Description: Generates a token for authentication.

Parameters:

None

Response:

Status Code: 200 OK

Content: JSON object containing the authentication token.

Example:

json

Copy code

{

"token": "abc123xyz456"

}

**Testing Considerations**

Verify that all endpoints respond correctly to valid requests with appropriate status codes and response bodies.

Test authentication by ensuring that protected endpoints reject requests without a valid token.

Check edge cases such as invalid inputs, missing parameters, and boundary values.

Test both positive and negative scenarios to ensure robustness of the API.

Conduct load testing to assess the performance and scalability of the API under different traffic conditions.

**Conclusion**

This Requirement Documentation provides an overview of the RESTful-Booker API and outlines the requirements for testing it effectively. By following these requirements, testers can ensure thorough evaluation of the API's functionality, security, and performance.