# **RESTful Booker API Documentation**

**Author: Leila**

**Date: October 25, 2024**

## **Overview**

The RESTful Booker API is designed to facilitate the management of hotel bookings through a straightforward RESTful interface. It allows for comprehensive operations on booking records, enabling developers to integrate booking functionalities into applications seamlessly.

## **Base URL**

The API is accessible at:  
https://restful-booker.herokuapp.com

## **Key Functionalities**

### **Authentication**

* **Endpoint**: POST /auth
* **Function**: Authenticates users and returns a token for subsequent secure requests.
* **Parameters**: Requires a valid username and password.
* **Response Codes**:
  + 200 OK: Successful authentication with token return.
  + 401 Unauthorized: Invalid credentials.

### **Booking Management**

1. **Create a Booking**
   * **Endpoint**: POST /booking
   * **Function**: Adds a new booking to the system.
   * **Request Body**: Must include firstname, lastname, totalprice, depositpaid, bookingdates, and optional additionalneeds.
   * **Response Codes**:
     + 201 Created: Booking successfully created.
2. **Retrieve All Bookings**
   * **Endpoint**: GET /booking
   * **Function**: Returns a list of all bookings in the system.
   * **Response Codes**:
     + 200 OK: List of bookings.
3. **Retrieve a Specific Booking**
   * **Endpoint**: GET /booking/{id}
   * **Function**: Fetches details for a specific booking identified by its unique ID.
   * **Response Codes**:
     + 200 OK: Booking details found.
     + 404 Not Found: Booking ID does not exist.
4. **Update a Booking**
   * **Endpoint**: PUT /booking/{id}
   * **Function**: Updates an existing booking's details.
   * **Request Body**: Same structure as the create booking request.
   * **Response Codes**:
     + 200 OK: Booking successfully updated.
     + 404 Not Found: Booking ID does not exist.
5. **Partially Update a Booking**
   * **Endpoint**: PATCH /booking/{id}
   * **Function**: Allows modification of specific fields in a booking.
   * **Request Body**: Contains only fields to be updated.
   * **Response Codes**:
     + 200 OK: Booking successfully updated.
6. **Delete a Booking**
   * **Endpoint**: DELETE /booking/{id}
   * **Function**: Removes a specific booking from the system.
   * **Response Codes**:
     + 204 No Content: Booking successfully deleted.
     + 404 Not Found: Booking ID does not exist.

### **Error Handling**

* The API employs standard HTTP response codes to indicate the outcome of requests:
  + 200 OK: Successful operation.
  + 201 Created: Resource successfully created.
  + 204 No Content: Resource successfully deleted.
  + 400 Bad Request: Malformed request or validation error.
  + 401 Unauthorized: Authentication failure.
  + 404 Not Found: Resource does not exist.

### 

### **Security Considerations**

* All operations requiring sensitive actions, such as creating, updating, or deleting bookings, necessitate authentication via the token.
* It is recommended to use HTTPS for secure data transmission.

## 

## **Conclusion**

The RESTful Booker API provides a robust and user-friendly interface for managing hotel bookings. By adhering to RESTful principles, it allows developers to easily integrate booking functionalities into their applications, ensuring a streamlined experience for both developers and end-users.