## **Requirement Document for Restful Booker API**

## **Introduction:-**

The RESTful Booker API provides a simple and robust system for managing hotel room reservations. This document specifies the requirements, endpoints, data structures, and behaviour of the API.

## **API Base URL:-**

### <https://restful-booker.herokuapp.com>

### 

## **Functional Requirements:-**

### **1. Authentication** (JSON)

* Endpoint : /auth
* Method : POST
* Description : Creates a new auth token to use for access to the PUT and DELETE /booking

### **Header** application/json

### **Body**

### {

### "username" : "admin",

### "password" : "password123"

### }

## **Response**

200 OK

**Body**

{

"token": "d50f64623e2ef64"

}

**2. Booking IDs**

* Endpoint : /booking
* Method : GET
* Description : Get IDs of all the bookings that exist within the API

## **Response**

200 OK

**Body**

{

"bookingid": 159

},

{

"bookingid": 425

},

{

"bookingid": 588

},

{

"bookingid": 614

},

{

"bookingid": 127

},

{

"bookingid": 437

},

{

"bookingid": 396

},

{

"bookingid": 670

},

### 

### 

### 

### 

### 

### **3. Get Bookings** (JSON)

* Endpoint : /booking/1
* Method : GET
* Description : Creates a specific booking based upon the booking id provided

### **Header** application/json

### **Body**

{

"firstname": "Sally",

"lastname": "Brown",

"totalprice": 111,

"depositpaid": true,

"bookingdates": {

"checkin": "2013-02-23",

"checkout": "2014-10-23"

},

"additionalneeds": "Breakfast"

}

**Response**

200 OK

**Body**

{

"firstname": "Jim",

"lastname": "Jackson",

"totalprice": 864,

"depositpaid": true,

"bookingdates": {

"checkin": "2023-08-13",

"checkout": "2023-10-18"

}

}

## **Bug :** Its Creating some random names as First name, Last name , Total price, Booking Dates

### 

### 

### **4. CreateBooking**

* Endpoint : /booking \
* Method : POST
* Description : Creates a new booking in the API

### **Header** - application/json

### **Body** (JSON)

### { "firstname" : "Jim",

### "lastname" : "Brown",

### "totalprice" : 111,

### "depositpaid" : true,

### "bookingdates" : {

### "checkin" : "2018-01-01",

### "checkout" : "2019-01-01"

### },

### "additionalneeds" : "Breakfast"

### }

## **Response**

200 OK

**Body**

{

"bookingid": 3931,

"booking": {

"firstname": "Jim",

"lastname": "Brown",

"totalprice": 111,

"depositpaid": true,

"bookingdates": {

"checkin": "2018-01-01",

"checkout": "2019-01-01"

},

"additionalneeds": "Breakfast"

}

}

### 

### 

### **5. UpdateBooking**

* Endpoint : /booking/1
* Method : PUT
* Description : Updates a current booking

### **Header** - application/json

### **Body** (JSON)

### {

### "firstname" : "James",

### "lastname" : "Brown",

### "totalprice" : 111,

### "depositpaid" : true,

### "bookingdates" : {

### "checkin" : "2018-01-01",

### "checkout" : "2019-01-01"

### },

### "additionalneeds" : "Breakfast"

### }

## **Response**

### 403 Forbidden

### 

### 

### 

### 

### **6. PartialUpdateBooking** (JSON)

* Endpoint : booking/:id
* Method : PATCH
* Description : Updates a current booking with a partial payload

### **Header** application/json

### **Body**

### {

### "firstname" : "James",

### "lastname" : "Brown",

### "additionalneeds" : "Lunch"

### }

## **Response**

403 Forbidden

### **6.DeleteBooking** (JSON)

* Endpoint : booking/1
* Method : DELETE
* Description : Delete a specific booking by ID

**Header** application/json

### **Response**

403 Forbidden