**Requirement Document for RESTful Booker API**

**Overview**

The RESTful Booker API is a web service designed to manage hotel room bookings. The API provides endpoints for creating, updating, retrieving, and deleting bookings. It also supports user authentication for secure operations.

**API Base URL**

https://restful-booker.herokuapp.com

**API Endpoints**

**1. Health Check**

* **Endpoint**: /ping
* **Method**: GET
* **Description**: Check the health status of the API.
* **Response**:
  + **200 OK**: API is healthy.

**2. Create Booking**

* **Endpoint**: /booking
* **Method**: POST
* **Description**: Create a new booking.
* **Request**:
  + **Headers**:
    - Content-Type: application/json
  + **Body** (JSON):

json

Copy code

{

"firstname": "Jim",

"lastname": "Brown",

"totalprice": 111,

"depositpaid": true,

"bookingdates": {

"checkin": "2020-01-01",

"checkout": "2020-01-02"

},

"additionalneeds": "Breakfast"

}

* **Response**:
  + **200 OK**: Booking created successfully.
  + **Body** (JSON):

json

Copy code

{

"bookingid": 1,

"booking": {

"firstname": "Jim",

"lastname": "Brown",

"totalprice": 111,

"depositpaid": true,

"bookingdates": {

"checkin": "2020-01-01",

"checkout": "2020-01-02"

},

"additionalneeds": "Breakfast"

}

}

**3. Get Booking**

* **Endpoint**: /booking/{id}
* **Method**: GET
* **Description**: Retrieve a specific booking by ID.
* **Request**:
  + **Path Parameter**:
    - id: Booking ID
* **Response**:
  + **200 OK**: Booking retrieved successfully.
  + **Body** (JSON):

json

Copy code

{

"firstname": "Jim",

"lastname": "Brown",

"totalprice": 111,

"depositpaid": true,

"bookingdates": {

"checkin": "2020-01-01",

"checkout": "2020-01-02"

},

"additionalneeds": "Breakfast"

}

* + **404 Not Found**: Booking not found.

**4. Update Booking**

* **Endpoint**: /booking/{id}
* **Method**: PUT
* **Description**: Update an existing booking by ID.
* **Request**:
  + **Headers**:
    - Content-Type: application/json
    - Authorization: Basic {token}
  + **Path Parameter**:
    - id: Booking ID
  + **Body** (JSON):

json

Copy code

{

"firstname": "Jim",

"lastname": "Brown",

"totalprice": 111,

"depositpaid": true,

"bookingdates": {

"checkin": "2020-01-01",

"checkout": "2020-01-02"

},

"additionalneeds": "Breakfast"

}

* **Response**:
  + **200 OK**: Booking updated successfully.
  + **Body** (JSON):

json

Copy code

{

"firstname": "Jim",

"lastname": "Brown",

"totalprice": 111,

"depositpaid": true,

"bookingdates": {

"checkin": "2020-01-01",

"checkout": "2020-01-02"

},

"additionalneeds": "Breakfast"

}

* + **403 Forbidden**: Authentication failed.
  + **404 Not Found**: Booking not found.

**5. Delete Booking**

* **Endpoint**: /booking/{id}
* **Method**: DELETE
* **Description**: Delete a specific booking by ID.
* **Request**:
  + **Headers**:
    - Authorization: Basic {token}
  + **Path Parameter**:
    - id: Booking ID
* **Response**:
  + **201 Created**: Booking deleted successfully.
  + **403 Forbidden**: Authentication failed.
  + **404 Not Found**: Booking not found.

**6. Partial Update Booking**

* **Endpoint**: /booking/{id}
* **Method**: PATCH
* **Description**: Partially update an existing booking by ID.
* **Request**:
  + **Headers**:
    - Content-Type: application/json
    - Authorization: Basic {token}
  + **Path Parameter**:
    - id: Booking ID
  + **Body** (JSON):

json

Copy code

{

"firstname": "James"

}

* **Response**:
  + **200 OK**: Booking partially updated successfully.
  + **Body** (JSON):

json

Copy code

{

"firstname": "James",

"lastname": "Brown",

"totalprice": 111,

"depositpaid": true,

"bookingdates": {

"checkin": "2020-01-01",

"checkout": "2020-01-02"

},

"additionalneeds": "Breakfast"

}

* + **403 Forbidden**: Authentication failed.
  + **404 Not Found**: Booking not found.

**7. Create Token**

* **Endpoint**: /auth
* **Method**: POST
* **Description**: Create a new authentication token.
* **Request**:
  + **Headers**:
    - Content-Type: application/json
  + **Body** (JSON):

json

Copy code

{

"username": "admin",

"password": "password123"

}

* **Response**:
  + **200 OK**: Token created successfully.
  + **Body** (JSON):

json

Copy code

{

"token": "abc123"

}

* + **200 OK**: Invalid credentials.

**Authentication**

* **Endpoint**: /auth
* **Method**: POST
* **Description**: Authenticate a user and receive a token for secure endpoints.

**Error Codes**

* **400 Bad Request**: Invalid request format.
* **401 Unauthorized**: Authentication required or failed.
* **403 Forbidden**: Access to the resource is forbidden.
* **404 Not Found**: Resource not found.
* **500 Internal Server Error**: Server encountered an error.

**Requirements**

1. **User Authentication**: Secure access to modify booking resources using Basic Auth with a token.
2. **CRUD Operations for Bookings**: Create, read, update, and delete bookings.
3. **Health Check**: Endpoint to verify API status.
4. **Partial Updates**: Ability to partially update booking details.
5. **Error Handling**: Proper error messages and status codes for different scenarios.

**Notes**

* Ensure all requests and responses adhere to the specified JSON formats.
* Authentication tokens should be securely managed and expired tokens handled appropriately.
* Consider rate limiting and logging to prevent abuse and ensure traceability.