**API Testing Project - RestFul Booker API Testing**

**Project Overview**

This project involves testing the RestFul Booker API, simulating real-world testing scenarios using various HTTP request methods (GET, POST, PUT, PATCH, DELETE). The API is used to manage bookings, create tokens, and authenticate users, among other functions.

Since no API was available for TradeOxy, I utilized the RestFul Booker API to demonstrate my skills in API testing, focusing on request validation, response validation, and error handling.

**API Endpoints Tested**

1. **Auth - CreateToken (POST)**

* **Purpose**: Creates a token for authenticating subsequent API requests.
* **Test Asserti:**
* **Status code is 200.**
* **Request Body**

{

"username": "admin",

"password": "password123"

}

1. **Booking - GetBookingIds (GET)**

* **Purpose:** Retrieves a list of all booking IDs.
* **Test Assertions:**

Verifies that the response contains a bookingid property.

* **Request**:

No request body; simply makes a GET request to /booking.

1. **Booking - GetBooking (GET)**

* **Purpose**: Retrieves a booking by its ID.
* **Test Assertions:**

Verifies that the response contains the firstname property.

Logs the response data for further inspection.

* **Request**:

Endpoint: /booking/:id

1. **Booking - CreateBooking (POST)**

* **Purpose**: Creates a new booking.
* **Test Assertions:**

Verifies that the Content-Type header of the response contains application/json.

* **Request Body**:

**{**

**"firstname": "karishma",**

**"lastname": "Yadav",**

**"totalprice": 140,**

**"depositpaid": true,**

**"bookingdates": {**

**"checkin": "2024-11-10",**

**"checkout": "2024-11-27"**

**},**

**"additionalneeds": "Lunch"**

**}**

1. **Booking - UpdateBooking (PUT)**

* **Purpose:** Updates an existing booking.
* **Test Assertions**: No specific assertions, but typically, you would check the response for confirmation.
* **Request Body:**

{

"firstname": "Riya",

"lastname": "Yadav",

"totalprice": 143,

"depositpaid": true,

"bookingdates": {

"checkin": "2024-01-01",

"checkout": "2024-08-11"

},

"additionalneeds": "Breakfast"

}

1. **Booking - PartialUpdateBooking (PATCH)**

* **Purpose**: Partially updates a booking.
* **Test Assertions**:

Verifies that the Content-Type header contains application/json.

* **Request Body**:

**{**

**"firstname": "James",**

**"lastname": "Brown"**

**}**

1. **Booking - DeleteBooking (DELETE)**

* **Purpose**:
* **Test Assertions**:
* **Request**:

Endpoint: /booking/:id

**Environment Variables**

* **RestfulBooker\_baseURL**: The base URL for the RestFul Booker API used in this project is:

**"RestfulBooker\_baseURL":** <https://restful-booker.herokuapp.com>

**Test Scripts**

Each of the requests in this collection includes associated tests to validate the response:

* **Status code validation**: Verifies if the response status is 200.
* **Response body validation**: Checks if specific properties exist in the response body
* **Content-Type validation**: Ensures the response has the correct content type (application/json).

**Tools Used**

* **Postman**: Used for creating, sending requests, and validating responses.
* **Postman Scripts**: JavaScript used within Postman to validate response status, body content, and headers.
* **API Authentication**: Basic authentication was used for some of the endpoints (e.g., Booking - GetBooking).

**Why Use RestFul Booker API?**

Since there was no API available for TradeOxy, I opted for the RestFul Booker API to showcase my API testing capabilities. The RestFul Booker API provides a wide range of endpoints that cover common CRUD operations and authentication mechanisms, which allowed me to demonstrate a variety of test cases and scenarios.