**API-AUTOMATION TESTING**

**Restful-booker: https://restful-booker.herokuapp.com/apidoc/index.html#api-Booking-DeleteBooking**

**Test Case Name: Create Token**

**Objective**:  
To generate a token required for authenticating subsequent requests.

**Test Prerequisites**:

* Ensure the API endpoint https://restful-booker.herokuapp.com/auth is accessible.
* Ensure valid username and password are provided (username: 'admin', password: 'password123').

**Test Steps**:

1. **Send POST Request**:  
   Send a POST request to the /auth endpoint with username and password as request data.
2. **Verify Status Code**:  
   Ensure the status code is 200.
3. **Extract Token**:  
   Retrieve the token from the response body for use in further requests.

**Expected Result**:  
A successful authentication request returns a 200 status and a token.

**Test Case Name: Create Booking**

**Objective**:  
To create a new booking and verify the successful creation.

**Test Prerequisites**:

* Ensure the token from the previous test is available for use.

**Test Steps**:

1. **Send POST Request**:  
   Send a POST request to the /booking endpoint with booking details (firstname, lastname, totalprice, etc.) in the request body.
2. **Verify Status Code**:  
   Ensure the status code is 200.
3. **Log Booking ID**:  
   Extract the bookingid from the response body.

**Expected Result**:  
The booking is successfully created, returning a 200 status and a valid bookingid.

**Test Case Name: Get Booking**

**Objective**:  
To retrieve a booking by its bookingId.

**Test Prerequisites**:

* Ensure the bookingId from the previous test is available.

**Test Steps**:

1. **Send GET Request**:  
   Send a GET request to the /booking/{bookingId} endpoint with the bookingId.
2. **Verify Status Code**:  
   Ensure the status code is 200.

**Expected Result**:  
The booking details should be returned successfully with a 200 status code.

**Test Case Name: Update Booking**

**Objective**:  
To update an existing booking with new details.

**Test Prerequisites**:

* Ensure the token from the "Create Token" test is available.
* Ensure the bookingId from the "Create Booking" test is available.

**Test Steps**:

1. **Send PUT Request**:  
   Send a PUT request to the /booking/{bookingId} endpoint, providing updated booking details (firstname, lastname, additionalneeds, etc.) along with the token in the headers.
2. **Verify Status Code**:  
   Ensure the status code is 200.

**Expected Result**:  
The booking is successfully updated, returning a 200 status code.

**Test Case Name: Get Booking After Update**

**Objective**:  
To verify the booking was successfully updated by retrieving the updated booking details.

**Test Prerequisites**:

* Ensure the bookingId from the "Create Booking" test is available.

**Test Steps**:

1. **Send GET Request**:  
   Send a GET request to the /booking/{bookingId} endpoint to retrieve the updated booking details.
2. **Verify Status Code**:  
   Ensure the status code is 200.

**Expected Result**:  
The updated booking details are returned successfully with a 200 status code.

**Test Case Name: Delete Booking**

**Objective**:  
To delete the created booking and verify successful deletion.

**Test Prerequisites**:

* Ensure the token from the "Create Token" test is available.
* Ensure the bookingId from the "Create Booking" test is available.

**Test Steps**:

1. **Send DELETE Request**:  
   Send a DELETE request to the /booking/{bookingId} endpoint with the token in the headers.
2. **Verify Status Code**:  
   Ensure the status code is 201, indicating successful deletion.

**Expected Result**:  
The booking is successfully deleted, returning a 201 status code.

**Restful-api :** [**https://restful-api.dev/**](https://restful-api.dev/)

**Test Case Name: List Of All Objects**

**Objective**:  
To verify that the API successfully returns a list of all objects.

**Test Prerequisites**:

* Ensure the API is running and accessible at https://api.restful-api.dev/.

**Test Steps**:

1. **Send GET Request**:  
   Send a GET request to the /objects endpoint.
2. **Verify Status Code**:  
   Ensure that the response status code is 200.
3. **Log Response**:  
   Print the response body to the console for verification.

**Expected Result**:  
The API should return a 200 status code and a list of objects.

**Test Case Name: Get Object By ID**

**Objective**:  
To verify that the API successfully returns an object by its ID.

**Test Prerequisites**:

* Ensure the API is running and accessible at https://api.restful-api.dev/.

**Test Steps**:

1. **Send GET Request**:  
   Send a GET request to the /objects/1 endpoint.
2. **Verify Status Code**:  
   Ensure that the response status code is 200.
3. **Log Response**:  
   Print the response body to the console.

**Expected Result**:  
The API should return a 200 status code and the object with ID 1.

**Test Case Name: Create Object**

**Objective**:  
To verify that the API successfully creates a new object.

**Test Prerequisites**:

* Ensure the API is running and accessible at https://api.restful-api.dev/.

**Test Steps**:

1. **Send POST Request**:  
   Send a POST request to the /objects endpoint with the new object data (e.g., name, year, price).
2. **Verify Status Code**:  
   Ensure that the response status code is 200.
3. **Log Response**:  
   Print the response body to the console and store the object ID for use in subsequent tests.

**Expected Result**:  
The object is successfully created, returning a 200 status code and the object’s id.

**Test Case Name: Get Object By ID Created**

**Objective**:  
To verify that the newly created object can be retrieved by its ID.

**Test Prerequisites**:

* Ensure the id from the "Create Object" test is available.

**Test Steps**:

1. **Send GET Request**:  
   Send a GET request to the /objects/{id} endpoint with the object ID.
2. **Verify Status Code**:  
   Ensure that the response status code is 200.
3. **Log Response**:  
   Print the response body to the console.

**Expected Result**:  
The API should return a 200 status code and the object details.

**Test Case Name: Update Object**

**Objective**:  
To verify that the API successfully updates an existing object.

**Test Prerequisites**:

* Ensure the id from the "Create Object" test is available.

**Test Steps**:

1. **Send PUT Request**:  
   Send a PUT request to the /objects/{id} endpoint with updated object data.
2. **Verify Status Code**:  
   Ensure that the response status code is 200.
3. **Log Response**:  
   Print the response body to the console.

**Expected Result**:  
The object is successfully updated, and the response status code is 200.

**Test Case Name: Get Object After Update**

**Objective**:  
To verify that the updated object details are correctly retrieved.

**Test Prerequisites**:

* Ensure the id from the "Create Object" and "Update Object" tests is available.

**Test Steps**:

1. **Send GET Request**:  
   Send a GET request to the /objects/{id} endpoint with the object ID.
2. **Verify Status Code**:  
   Ensure that the response status code is 200.
3. **Log Response**:  
   Print the response body to the console.

**Expected Result**:  
The API should return the updated object with a 200 status code.

**Test Case Name: Partially Update Object**

**Objective**:  
To verify that the API successfully performs a partial update of an object.

**Test Prerequisites**:

* Ensure the id from the "Create Object" test is available.

**Test Steps**:

1. **Send PATCH Request**:  
   Send a PATCH request to the /objects/{id} endpoint with partial object data.
2. **Verify Status Code**:  
   Ensure that the response status code is 200.
3. **Log Response**:  
   Print the response body to the console.

**Expected Result**:  
The object is partially updated successfully, returning a 200 status code.

**Test Case Name: Get Object After Partially Update**

**Objective**:  
To verify that the object details reflect the partial update.

**Test Prerequisites**:

* Ensure the id from the "Create Object" and "Partially Update Object" tests is available.

**Test Steps**:

1. **Send GET Request**:  
   Send a GET request to the /objects/{id} endpoint with the object ID.
2. **Verify Status Code**:  
   Ensure that the response status code is 200.
3. **Log Response**:  
   Print the response body to the console.

**Expected Result**:  
The API should return the partially updated object with a 200 status code.

**Test Case Name: Delete Object**

**Objective**:  
To verify that the API successfully deletes an object.

**Test Prerequisites**:

* Ensure the id from the "Create Object" test is available.

**Test Steps**:

1. **Send DELETE Request**:  
   Send a DELETE request to the /objects/{id} endpoint with the object ID.
2. **Verify Status Code**:  
   Ensure that the response status code is 200.
3. **Log Response**:  
   Print the response body to the console.

**Expected Result**:  
The object is successfully deleted, and the response status code is 200.