A close-up of a logo

Description automatically generated

**Assignment: Implement Request Chaining for Restful Booker API in Makaia Framework**

**Objective**

Perform Create, Update, and Get All booking IDs operations using request chaining with the Restful Booker API in the Makaia Framework.

**Prerequisites**

Makaia Framework setup

Restful Booker API access

**Steps**

**1. Setup Configuration Properties**

Open the config\_api.properties file.

Add the Restful Booker API base URL:

server = <https://restful-booker.herokuapp.com>

**2. Create a New Java Class for Booking Operations**

Create a new Java class named TC\_BookingOperations.java in the **tests.rest** package.

**3. Implement the Chaining of Requests**

**Setup Base URI:**

Configure RestAssured with the base URI from the **config\_api.properties** file.

**Create a Booking:**

Define the payload for creating a booking, which includes fields like firstname, lastname, totalprice, depositpaid, bookingdates, and additionalneeds.

Send a POST request to the /booking endpoint.

Extract the **bookingid** from the response.

**Update the Booking:**

Define the payload for updating the booking, which should include updated fields such as firstname, lastname, totalprice, depositpaid, bookingdates, and additionalneeds.

Send a PUT request to the /booking/{bookingid} endpoint using the extracted **bookingid**.

Verify the response to ensure the booking has been updated correctly.

**Get All Booking IDs:**

Send a GET request to the /booking endpoint to retrieve all booking IDs.

Verify the response status code to ensure it returns 200 and that the list of booking IDs is correctly retrieved.

**Detailed Steps**

**Setup Base URI:**

Add the server URL in the config\_api.properties file.

In your test class (TC\_BookingOperations.java), configure the base URI using RestAssured.

**Create a Booking:**

Define the JSON payload for creating a booking with necessary details.

Send a POST request to the /booking endpoint.

Extract the bookingid from the response for use in subsequent requests.

**Update the Booking:**

Define the JSON payload for updating the booking with updated details.

Send a PUT request to the /booking/{bookingid} endpoint using the bookingid extracted from the previous response.

Verify that the booking has been updated by checking the response content.

**Get All Booking IDs:**

Send a GET request to the /booking endpoint to retrieve all booking IDs.

Verify the response status code to ensure it is 200 and the list of booking IDs is returned.

**Summary**

**Configuration:** Set the base URL in config\_api.properties.

**Create Booking:** Send a POST request with booking details and extract bookingid.

**Update Booking:** Send a PUT request with updated booking details using the bookingid.

**Get All Booking IDs:** Send a GET request to retrieve and verify all booking IDs.

Top of Form

Bottom of Form