API Test Document

- **Test Scenario**: Verify that the **Dog API** returns a random dog image successfully

Test Case 1: Verify the response status code is 200

Method: GET

Endpoint: https://dog.ceo/api/breeds/image/random

Test Steps:

1. Open API testing tool (e.g., Postman).
2. Set method to GET.
3. Enter the endpoint: https://dog.ceo/api/breeds/image/random.
4. Send the request.
5. Verify that the response status code

Expected Result: Status code should be 200 Ok

Test Case 2: Verify the response content type is JSON

Method: GET

Endpoint: https://dog.ceo/api/breeds/image/random

Test Steps:

1. Send a GET request to https://dog.ceo/api/breeds/image/random.
2. In the response headers, locate Content-Type.
3. Confirm that Content-Type = application/json.

Expected Result: Content-Type header should be application/json

Test Case 3: Verify that the response body contains an image URL

Method: GET

Endpoint: https://dog.ceo/api/breeds/image/random

Test Steps:

1. Send a GET request to the endpoint.
2. Inspect the response body.
3. Check a key named "message" in the response body

Expected Result: JSON response should contain a key "message" with a valid image URL ending in .jpg or .jpeg or .png

---

Test Case 4: Verify that the response does not contain a valid image URL for an invalid endpoint

Method: GET

Endpoint: https://dog.ceo/api/breeds/image/invalid

Test Steps:

1. Send a GET request to the invalid endpoint.
2. Verify that: The status code is 404

Expected Result: Response status code should be 404 and JSON response shouldn’t contain a key "message" with a valid image URL ending in .jpg, .jpeg, or .png

---

- **Test Scenario:** Verify that Objects added and deleted using **Objects API**

Test Case 1: Add an object with valid data

Method: POST

Endpoint: https://api.restful-api.dev/objects

Test Steps:

1. Set method to POST.
2. Set endpoint: https://api.restful-api.dev/objects
3. Check the status code after send the request

Expected Result: Status Code: 200 or 201

Test case 2: Verify the added object is stored correctly

Precondition: A valid object has been added.

Method: GET

Endpoint: https://api.restful-api.dev/objects/{id}

Test Steps:

1. After creating the object, copy the returned id.
2. Send a GET request to https://api.restful-api.dev/objects/{id}.
3. Verify the status code is 200.
4. Confirm that the returned data matches the original input

Expected Result: Status Code: 200

Test case 3: Validate that the Content-Type of the response is application/json.

Method: GET

Endpoint: https://api.restful-api.dev/objects/{id}

Test Steps

1. Send a GET request to https://api.restful-api.dev/objects/{id}.
2. Inspect the response headers.
3. Check that Content-Type = application/json

Expected Result: Header: Content-Type: application/json

Test case 4: Verify behavior with an invalid object ID.

Method: GET

Endpoint: https://api.restful-api.dev/objects/invalid123

Test Steps:

1. Send a GET request to the invalid object ID.
2. Verify that the status code is 404.

Expected Result: Status Code: 404

Test Case 5: Delete the object using its ID

Precondition: Object exists with a valid ID.

Method: DELETE

Endpoint: https://api.restful-api.dev/objects/{id}

Test Steps:

1. Use the id of the created object.
2. Send a DELETE request to https://api.restful-api.dev/objects/{id}.
3. Verify the status code is 200

Expected Result: Status Code: 200 and Message: Object with id = {id} has been deleted

---

Test Case 6: Validate the product response against JSON schema

Method: GET

Endpoint: https://api.restful-api.dev/objects/

Test Steps:

- Use a tool like Postman, Swagger, or a testing library like REST Assured with schema validation.
- Send a GET request to https://api.restful-api.dev/objects/.
- Apply the JSON Schema (products-schema.json) to validate the structure.
- Ensure the schema validation passes with no errors.

Schema File: products-schema.json

Expected Result: Schema validation passes with no error
