API Manual Testing

Project: Pet API

Created by: Marcela Jataí

Date of creation: 25/03/2022

Last Update: 27/03/2022

Prerequisite: Postman and valid active account on Postman

General instructions:

1. Create a new collection and name it as “Pet Store”
2. Create a new Request

Test 1: Post/Status Code 201

1. Method: Post
2. Name: New Pet
3. Enter request URL: http://testchallenge.com/newPet
4. Headers:
   - Key: Content-Type
   - Value: application/json
5. Body:
   - Raw -> document type: JSON

```
{
“name”: “Bob”,
“petType”: “cat”,
“age”: 5,
“race”: “not defined”
}
```

6. Tests:

```
pm.test(“Status code is 201”, function() {
        pm.response.to.have.status(201);
}) ;
const schema = {
        "type": "object",
        "properties": {
          "name": {
            "type": "string",
                        "maxLenght": 100
```

```
           },
           "petType": {
              "type": "string",
                            "maxLenght": 50
           },
           "age": {
              "type": "integer"
           },
           "race": {
              "type": "string",
                            "maxLenght": 100
           }
        },
        "required": [
           "age",
           "name",
           "petType",
           "race"
        ]
};

pm.test("Validating schema", () => {
        pm.response.to.have.jsonSchema(schema);
});
```

7. Click “send”

<u>Expected results:</u>

Response:

```
{
“Id”: “121314”,
“name”: “Bob”,
“petType”: “cat”,
“age”: 5,
“race”: “not defined”
}
```

Status Code: 201 – Created

Test Results:

- Pass + “Status Code is 201”

- Pass + “Schema is valid”

Test 2: Post/Status Code 400

1. Method: Post
2. Name: New Pet
3. Enter request URL: http://testchallenge.com/newPet
4. Headers:
   - Key: Content-Type
   - Value: application/json
5. Body:
   - Raw -> document type: JSON

```
{
“name”: “Belle”,
“petType”: “Dog”,
“age”: “seven”,
“race”: “Husky”
}
```

6. Tests:

```
pm.test(“Status code is 201”, function() {
         pm.response.to.have.status(201);
}) ;
const schema = {
        "type": "object",
        "properties": {
           "name": {
              "type": "string",
                            "maxLenght": 100
           },
           "petType": {
              "type": "string",
                            "maxLenght": 50
           },
           "age": {
              "type": "integer"
           },
           "race": {
              "type": "string",
                            "maxLenght": 100
           }
        },
        "required": [
           "age",
           "name",
                "petType",
                "race"
            ]
    };

    pm.test("Validating schema", () => {
            pm.response.to.have.jsonSchema(schema);
    });
```

7. Click “send”

<u>**Expected results:**</u>

Status Code: 400 – Bad Request

Test Results:

- “Fail” + “Status Code is 400 | AssertionError: expected response to be 201 but got 400”

- “Fail” + Validating Schema | AssertionError: expected data to satisfy schema but found the following erros: data.age should be integer

Test 3: Post/Status Code 404

1. Method: Post
2. Name: New Pet
3. Enter request URL: http://testchallenge.com/newPert
4. Headers:
   - Key: Content-Type
   - Value: application/json
5. Body:
   - Raw -> document type: JSON

```
{
“name”: “Belle”,
“petType”: “Dog”,
“age”: “seven”,
“race”: “Husky”
}
```

6. Tests:

```
pm.test(“Status code is 201”, function() {
        pm.response.to.have.status(201);
}) ;
const schema = {
        "type": "object",
        "properties": {
           "name": {
              "type": "string",
                            "maxLenght": 100
           },
           "petType": {
              "type": "string",
                            "maxLenght": 50
           },
           "age": {
              "type": "integer"
           },
           "race": {
              "type": "string",
                            "maxLenght": 100
           }
        },
        "required": [
           "age",
           "name",
           "petType",
           "race"
        ]
};

pm.test("Validating schema", () => {
        pm.response.to.have.jsonSchema(schema);
});
```

7. Click “send”

Expected results:

Status Code: 404 – Not Found

Test Results:

- “Fail” + “Status Code is 404 | AssertionError: expected response to be 201 but got 404”

Test 4: Post/Status Code 500

1. Method: Post
2. Name: New Pet

3. Enter request URL: http://testchallenge.com/newPet

   This error is returned when something is wrong with the API. If this error shows up, I'd ask for the developer to help me and show that I got this error and I'd ask to see and analyze the API log with me.

Test 5: Get/Status Code 200

1. Method: Get
2. Name: Search Pet
3. Enter request URL: http://testchallenge.com/newPet/121314
4. Headers:
   - Key: Content-Type
   - Value: application/json

5. Tests:

```
pm.test("Status code is 200", function() {
        pm.response.to.have.status(200);
}) ;
```

<u>Expected results:</u>

Status Code: 200 – OK

Test Results:

- "Pass" + "Status Code is 200"

Test 6: Get/Status Code 400

1. Method: Get
2. Name: Search Pet
3. Enter request URL: http://testchallenge.com/newPet/1213154 (invalid ID)
4. Headers:
   - Key: Content-Type
   - Value: application/json

5. Tests:

```
pm.test("Status code is 200", function() {
        pm.response.to.have.status(200);
}) ;
```

<u>Expected results:</u>

Status Code: 400 – Bad Request

Test Results:

- "Fail" + "Status Code is 400" | AssertionError: expected response to be 200 but got 400

Test 6: Get/Status Code 404

1. Method: Get
2. Name: Search Pet
3. Enter request URL: http://testchallenge.com/newPet/221314 (Unexisting ID)
4. Headers:
   - Key: Content-Type
   - Value: application/json

5. Tests:

```
pm.test("Status code is 200", function() {
        pm.response.to.have.status(200);
}) ;
```

<u>Expected results:</u>

Status Code: 404 – Not Found

Test Results:

- "Fail" + "Status Code is 404" | AssertionError: expected response to be 200 but got 404

Test 8: Get/Status Code 500

1. Method: Get
2. Name: Search Pet
3. Enter request URL: http://testchallenge.com/newPet/121314

This error is returned when something is wrong with the API. If this error shows up, I’d ask for the developer to help me and show that I got this error and I’d ask to see and analyze the API log with me.