**API Manual Testing**

Project: Pet API

Created by: Marcela Jataí

Date of creation: 25/03/2022

Last Update: 25/03/2022

To the API testing, I’ll be using the application “Postman” which I think is the better one to test APIs in the present.

1st step: Download Postman and create a valid account to access the application.

2nd step: Create a new Collection. Name it as “Pet API”

3rd step: Create a new request

**Test 1 – Method: Post - (201 – created)**

1. On the “Name” input field: name your test as Post New Pet
2. On the URL input field: <http://testchallenge.com/newPet>
3. Check the headers:

- Key: Content-Type

- Value: application/json.

1. On the “body”: change to “raw” and change the file type to JSON

{

“name”: “Bob”,

“petType”: “cat”,

“age”: 5,

“race”: “not defined”

}

1. On “tests”: add the following script to validate the status and the data type and max lenght

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);

});

1. Click “Send”

**Expected results:**

On the test body result it will show the new input you just add to the API, like shown below:

{

“Id”: “121314”,

“name”: “Bob”,

“petType”: “cat”,

“age”: 5,

“race”: “not defined”

}

The status code of the test will be “201 – Created”

On the “Test Results” tab it will show:

- “Pass” + “Status Code is 201”

- “Pass” + Schema is valid

**Test 2 – Method: Post - (400 – bad request)**

1. On the “Name” input field: name your test as Post New Pet
2. On the URL input field: <http://testchallenge.com/newPet>
3. Check the headers:

- Key: Content-Type

- Value: application/json.

1. On the “body”: change to “raw” and change the file type to JSON. To simulate a bad request, which is a cliente-side mistake, we can try by:

- Type more than 100 characters on the “name”

- Type more the 50 charactes on the “petType”

- Insert a letter or special character on “age”

- Type more than 50 charactes on the “race”

Ex:

{

“name”: “Belle”,

“petType”: “Dog”,

“age”: “seven”,

“race”: “Husky”

}

1. On “tests”: add the following script to validate the status

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);

});

1. Click “Send”

**Expected results:**

The status code of the test will be “400 – Bad Request”

On the “Test Results” tab it will show:

- “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 – Method: Post - (404 – Not Found)**

1. On the “Name” input field: name your test as Post New Pet
2. On the URL input field: <http://testchallenge.com/newPet>r (to simulate a 404 error, introducing an aditional letter on the end point, so it can return the not found error)
3. Check the headers:

- Key: Content-Type

- Value: application/json.

1. On the “body”: change to “raw” and change the file type to JSON

{

“name”: “Bob”,

“petType”: “cat”,

“age”: 5,

“race”: “not defined”

}

1. On “tests”: add the following script to validate the status and the data type and max lenght

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);

});

1. Click “Send”

**Expected results:**

The status code of the test will be “404 - Not Found”

On the “Test Results” tab it will show:

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

**Test 4 – Method: Post - (500 – Internal Server Error)**

This error is returned when something is wrong inside 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 – Method: Get - (200 – OK)**

1. On the “Name” input field: name your test as Get New Pet
2. On the URL input field: <http://testchallenge.com/newPet/121314> (We’ll check if Bob is really here)
3. Check the headers:

- Key: Content-Type

- Value: application/json.

1. Click Send
2. On “tests”: add the following script to validate the status and the data type and max lenght

pm.test(“Status code is 200”, function() {

pm.response.to.have.status(200);

}) ;

**Expected results:**

On the test body result it will show the new input you just add to the API, like shown below:

{

“Id”: “121314”,

“name”: “Bob”,

“petType”: “cat”,

“age”: 5,

“race”: “not defined”

}

The status code of the test will be “200 – OK”

On the “Test Results” tab it will show:

- “Pass” + “Status Code is 200”

**Test 6 – Method: Get - (400 – Bad Request)**

1. On the “Name” input field: name your test as Get New Pet
2. On the URL input field: <http://testchallenge.com/newPet/121314>% (We’ll change de end point to and unexisting one with a diferente character)
3. Check the headers:

- Key: Content-Type

- Value: application/json.

1. Click Send
2. On “tests”: add the following script to validate the status and the data type and max lenght

pm.test(“Status code is 200”, function() {

pm.response.to.have.status(200);

}) ;

**Expected results:**

The status code of the test will be “400 – Bad Request”

On the “Test Results” tab it will show:

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

**Test 7 – Method: Get - (404 – Not Found)**

1. On the “Name” input field: name your test as Get New Pet
2. On the URL input field: <http://testchallenge.com/newPet/121314>5 (We’ll change de end point to and unexisting one)
3. Check the headers:

- Key: Content-Type

- Value: application/json.

1. Click Send
2. On “tests”: add the following script to validate the status and the data type and max lenght

pm.test(“Status code is 200”, function() {

pm.response.to.have.status(200);

}) ;

**Expected results:**

The status code of the test will be “400 – Not Found”

On the “Test Results” tab it will show:

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

**Test 8 – Method: Get - (500 – Internal Server Error)**

This error is returned when something is wrong inside 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.