**Pokémon API using Postman**

<https://pokeapi.co/api/v2/berry/45>

**Question:**

The final project for this course is to build a suite of tests for the Pokémon API using Postman.

Here is what you have to do:

- Create a new collection in Postman

- Create GET requests for "Berries" and "Pokémon"

- Create as many test cases as you can think of

- Make sure your tests pass!

See the Project video for additional details on what to do. I have provided my solution for you to compare in the final video.

Take the test script that you have created and post it for others to see what you have learned.

{

"**firmness**": {

"name": "soft",

"url": "https://pokeapi.co/api/v2/berry-firmness/2/"

},

"**flavors**": [

{

"flavor": {

"name": "spicy",

"url": "https://pokeapi.co/api/v2/berry-flavor/1/"

},

"potency": 0

},

{

"flavor": {

"name": "dry",

"url": "https://pokeapi.co/api/v2/berry-flavor/2/"

},

"potency": 0

},

{

"flavor": {

"name": "sweet",

"url": "https://pokeapi.co/api/v2/berry-flavor/3/"

},

"potency": 10

},

{

"flavor": {

"name": "bitter",

"url": "https://pokeapi.co/api/v2/berry-flavor/4/"

},

"potency": 0

},

{

"flavor": {

"name": "sour",

"url": "https://pokeapi.co/api/v2/berry-flavor/5/"

},

"potency": 15

}

],

"**growth\_time**": 18,

"**id**": 45,

"**item**": {

"name": "payapa-berry",

"url": "https://pokeapi.co/api/v2/item/170/"

},

"**max\_harvest**": 5,

"**name**": "payapa",

"**natural\_gift\_power**": 60,

"**natural\_gift\_type**": {

"name": "psychic",

"url": "https://pokeapi.co/api/v2/type/14/"

},

"**size**": 252,

"**smoothness**": 30,

"**soil\_dryness**": 6

}

// To check Status Code is 200

pm.**test**("Status code is 200", **function** () {

    pm.response.to.have.status(200);

});

// To check Status Code is 'OK'

pm.**test**("Status code name has string", **function** () {

    pm.response.to.have.status("OK");

});

// Check the response time. Here it takes 109ms (frequently changes). so 200ms is fine

pm.**test**("Response time is less than 200ms", **function** () {

    pm.expect(pm.response.responseTime).to.be.below(200);

});

// Check response body contains string -> This won't work properly. Not gonna use it.

pm.**test**("Body matches string", **function** () {

    pm.expect(pm.response.text()).to.include("6");

});

// Best way to check string with value: Response body JSON value check. So, in this way we will check all.

// soil\_dryness

var jsonData**=** pm.response.json();

pm.**test**("soil\_dryness is Correct", **function**(){

pm.expect(jsonData.soil\_dryness).to.eql(6);

})

// smoothness

pm.**test**("smoothness is Correct", **function**(){

    pm.expect(jsonData.smoothness).to.eql(30);

});

// size

pm.**test**("size is Correct", **function**(){

    pm.expect(jsonData.size).to.eql(252);

});

// natural\_gift\_power

pm.**test**("natural\_gift\_power is Correct", **function**(){

    pm.expect(jsonData.natural\_gift\_power).to.eql(60);

});

// name

pm.**test**("name is Correct", **function**(){

pm.expect(jsonData.name).to.eql('payapa');

});

// max\_harvest

pm.**test**("max\_harvest is Correct", **function**(){

    pm.expect(jsonData.max\_harvest).to.eql(5);

});

// id

pm.**test**("id is Correct", **function**(){

pm.expect(jsonData.id).to.eql(45);

});

// growth\_time

pm.**test**("growth\_time is Correct", **function**(){

    pm.expect(jsonData.growth\_time).to.eql(18);

});

// firmness.name

pm.**test**("firmness name is Correct", **function**(){

    pm.expect(jsonData.firmness.name).to.eql('soft');

});

// firmness.url

pm.**test**("firmness url is Correct", **function**(){

    pm.expect(jsonData.firmness.url).to.eql('https://pokeapi.co/api/v2/berry-firmness/2/');

});

// item.name

pm.**test**("item name is Correct", **function**(){

    pm.expect(jsonData.item.name).to.eql('payapa-berry');

});

// item.url

pm.**test**("item name is Correct", **function**(){

    pm.expect(jsonData.item.url).to.eql('https://pokeapi.co/api/v2/item/170/');

});

// natural\_gift\_type.name

pm.**test**("natural\_gift\_type name is Correct", **function**(){

    pm.expect(jsonData.natural\_gift\_type.name).to.eql('psychic');

});

// natural\_gift\_type.url

pm.**test**("natural\_gift\_type name is Correct", **function**(){

    pm.expect(jsonData.natural\_gift\_type.url).to.eql('https://pokeapi.co/api/v2/type/14/');

});

// flavors -> Complicated One -> It is an array

// flavor -> 1 ok

pm.**test**("flavors[0].flavor.name is Correct", **function**(){

    pm.expect(jsonData.flavors[0].flavor.name).to.eql('spicy');

});

pm.**test**("flavors[0].flavor.url is Correct", **function**(){

    pm.expect(jsonData.flavors[0].flavor.url).to.eql('https://pokeapi.co/api/v2/berry-flavor/1/');

});

pm.**test**("flavors[0].potency is Correct", **function**(){

    pm.expect(jsonData.flavors[0].potency).to.eql(0);

});

// flavor -> 2

pm.**test**("flavors[1].flavor.name is Correct", **function**(){

    pm.expect(jsonData.flavors[1].flavor.name).to.eql('dry');

});

pm.**test**("flavors[1].flavor.url is Correct", **function**(){

    pm.expect(jsonData.flavors[1].flavor.url).to.eql('https://pokeapi.co/api/v2/berry-flavor/2/');

});

pm.**test**("flavors[1].potency is Correct", **function**(){

    pm.expect(jsonData.flavors[1].potency).to.eql(0);

});

// flavor -> 3

pm.**test**("flavors[2].flavor.name is Correct", **function**(){

    pm.expect(jsonData.flavors[2].flavor.name).to.eql('sweet');

});

pm.**test**("flavors[2].flavor.url is Correct", **function**(){

    pm.expect(jsonData.flavors[2].flavor.url).to.eql('https://pokeapi.co/api/v2/berry-flavor/3/');

});

pm.**test**("flavors[2].potency is Correct", **function**(){

    pm.expect(jsonData.flavors[2].potency).to.eql(10);

});

// flavor -> 4

pm.**test**("flavors[3].flavor.name is Correct", **function**(){

    pm.expect(jsonData.flavors[3].flavor.name).to.eql('bitter');

});

pm.**test**("flavors[3].flavor.url is Correct", **function**(){

    pm.expect(jsonData.flavors[3].flavor.url).to.eql('https://pokeapi.co/api/v2/berry-flavor/4/');

});

pm.**test**("flavors[3].potency is Correct", **function**(){

    pm.expect(jsonData.flavors[3].potency).to.eql(0);

});

// flavor -> 5

pm.**test**("flavors[4].flavor.name is Correct", **function**(){

    pm.expect(jsonData.flavors[4].flavor.name).to.eql('sour');

});

pm.**test**("flavors[4].flavor.url is Correct", **function**(){

    pm.expect(jsonData.flavors[4].flavor.url).to.eql('https://pokeapi.co/api/v2/berry-flavor/5/');

});

pm.**test**("flavors[4].potency is Correct", **function**(){

    pm.expect(jsonData.flavors[4].potency).to.eql(15);

});

**// Response Headers**

**Date**: Wed, 19 Oct 2022 12:33:37 GMT

**Content**-**Type**: application/json; charset=utf-8

**Transfer-Encoding:** chunked

**Connection**: keep-alive

**access-control-allow-origin:** \*

**Cache-Control:** public, max-age=86400, s-maxage=86400

**etag**: W/"338-u4i3kNqhbkLJY9IDSK90mFwNOMQ"

function-execution-id: yt53a5q4ae8o

strict-transport-security: max-age=31556926

x-cloud-trace-context: 719825c26c3e09b610ae68b04d2856f2;o=1

x-country-code: SG

x-powered-by: Express

x-served-by: cache-qpg1257-QPG

x-cache: HIT

x-cache-hits: 1

x-timer: S1666161315.949701,VS0,VE1

vary: Accept-Encoding,cookie,need-authorization, x-fh-requested-host, accept-encoding

alt-svc: h3=":443"; ma=86400, h3-29=":443"; ma=86400

CF-Cache-Status: HIT

Age: 21503

Server-Timing: cf-q-config;dur=5.0000016926788e-06

Report-To: {"endpoints":[{"url":"https:\/\/a.nel.cloudflare.com\/report\/v3?s=WDRrRD5rrn5Wsu7K2%2Fdc9Z6T9pOGsKJhXDAlpkuKFAI%2B52FPwIoBZq7H5Pu3qYSpGUsAtxJuILAHLpIggbyuqLl6nyR3sCnj7428t9eGtKaKIs7r1HHyjrMDU%2FML"}],"group":"cf-nel","max\_age":604800}

NEL: {"success\_fraction":0,"report\_to":"cf-nel","max\_age":604800}

Server: cloudflare

CF-RAY: 75c98b108dba469d-SIN

Content-Encoding: br