Practice 1

1. (20') For Get All Tools endpoint, apply query parameter, category = electric-generators. Create test scripts to assert the returned tool category is electric-generators. Try both assertions passed and failed.
2. (10') Manually take the value of electric generators and store it in a collection variable called category.
3. Replace all occurrences of electric-generators with the collection variable, category.
   * (10') query parameter in the url of Get All Tools endpoint
   * (10') the test script for checking response category value, which you complete in step1.

Practice 2

1. (10') Use the saved global variable toolId in the request Get a Single Tool. Open the Postman console to verify the data that has been sent.
2. (10') Use the saved global variable toolId in the request Create an Order.

Practice 3

For the Get a Single Tool Endpoint, create test script to

1. (10') Check if the toolId in response matches the request.
2. (20') Check if the current-stock is greater than 0. Please try to search the correct syntax on google.

Practice 1

1. For Get All Tools endpoint, apply query parameter, category = electric-generators. Create test scripts to assert the returned tool category is electric-generators. Try both assertions passed and Failed.

Tested with query parameters, category = electric-generators.

A screenshot of a computer

Description automatically generated

Test Scripts for Passed Assertion:

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

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

});

pm.test("All tools belong to the electric-generators category", function () {

let responseJson = pm.response.json();

responseJson.forEach(tool => {

pm.expect(tool.category).to.eql("electric-generators");

});

});

Output:

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

Test for Failed Assertion:

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

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

});

pm.test("All tools belong to an incorrect category", function () {

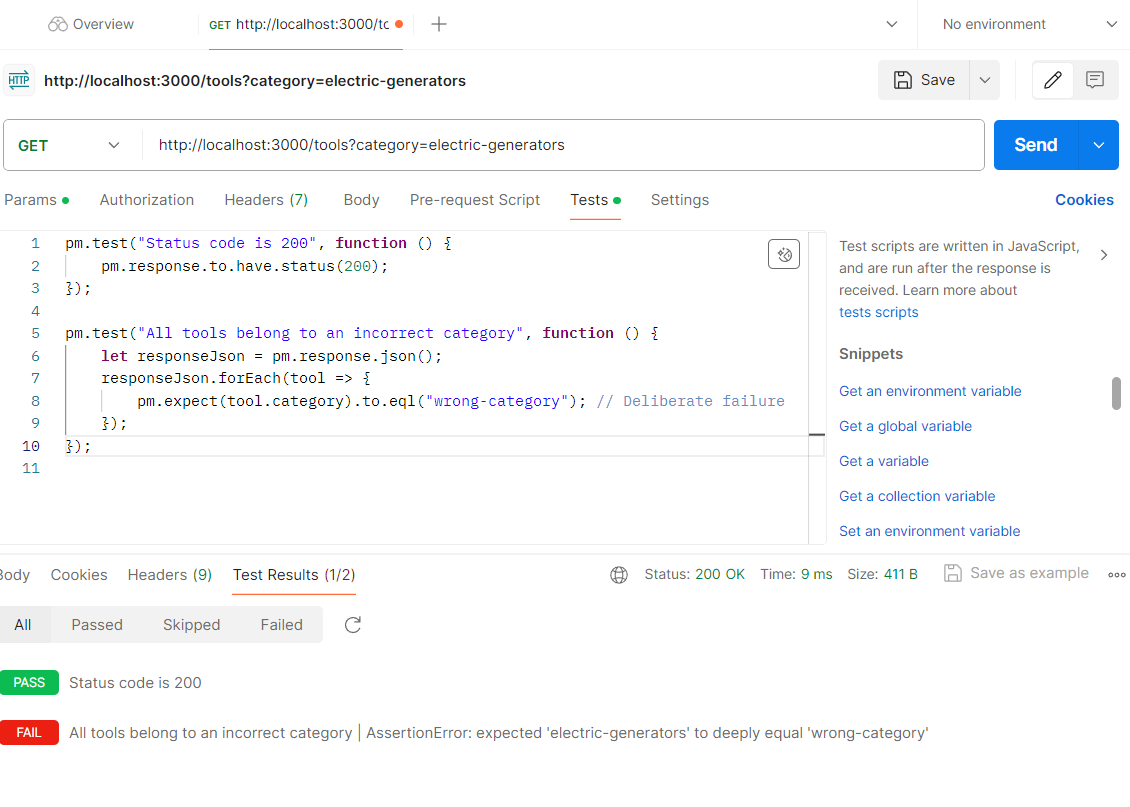
let responseJson = pm.response.json();

responseJson.forEach(tool => {

pm.expect(tool.category).to.eql("wrong-category"); // Deliberate failure

});

});



2. Manually take the value of electric generators and store it in a collection variable called category.

Setting an Collection variable:

A screenshot of a computer

Description automatically generated

3. Replace all occurrences of electric-generators with the collection variable, category.

Query parameter in the url of Get All Tools endpoint

A screenshot of a computer

Description automatically generated

The test script for checking response category value, which you complete in step1

Test Script:

// Get the collection variable

let expectedCategory = pm.collectionVariables.get("category");

// Log the expected category for debugging

console.log("Expected Category:", expectedCategory);

// Validate the status code

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

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

});

// Parse the response body

let responseJson = pm.response.json();

// Test if all items in the response have the correct category

pm.test("All tools belong to the expected category", function () {

responseJson.forEach(tool => {

pm.expect(tool.category).to.eql(expectedCategory, `Tool with ID ${tool.id} does not match the expected category`);

});

});

Output:

A screenshot of a computer

Description automatically generated

Practice 2

1. Use the saved global variable toolId in the request Get a Single Tool. Open the Postman console to verify the data that has been sent.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

Console

A screenshot of a computer

Description automatically generated

2.Use the saved global variable toolId in the request Create an Order.

A screenshot of a computer

Description automatically generated

Practice 3

1. Test Script to Check if the toolId in response matches the request.

// Retrieve the toolId global variable

const requestedToolId = pm.globals.get("toolId");

// Parse the response body

pm.test("Response should match the global toolId", function () {

const response = pm.response.json();

// Ensure response contains toolId

pm.expect(response).to.have.property('toolId');

// Verify the toolId in the response matches the global variable

pm.expect(response.toolId).to.eql(requestedToolId);

});

// Test for a 404 scenario

if (pm.response.code === 404) {

pm.test("Tool not found error message", function () {

const response = pm.response.json();

pm.expect(response.message).to.eql("Tool not found");

});

}

A screenshot of a computer

Description automatically generated

A screenshot of a chat

Description automatically generated

2.Check if the current-stock is greater than 0. Please try to search the correct syntax on google.

A screenshot of a computer

Description automatically generated

Test Results

A screenshot of a computer

Description automatically generated

Manually Changing Stock of Tool

A screenshot of a computer

Description automatically generated