GET Product search by ID

GET: Find a product in the catalog by ID

Priority	Behavior	Туре	Layer
↑ High	Positive	Functional	API
Is Flaky	Milestone	Automation	Status
No	-	Manual	Actual

Is Muted

No

Pre-conditions

GET method endpoint: {{base_URL}}/products/id/:productId

Qa-Demoshopping Variables: base_URL, Value: https://qa.demoshopping.ru

Authorization: Inherit auth from parent: {{token}} **Path Variables -** Key: productld, Value: {{product_id}}

Environments: QA is active

Post-conditions

Post-response Script:

```
pm.test("Status code is 200", function () {
pm.response.to.have.status(200);
});
pm.test("Ответ является массивом продуктов с корректной структурой", function () {
let responseData = pm.response.json();
// Проверка, что каждый объект в массиве имеет правильную структуру
if (Array.isArray(responseData)) {
responseData.forEach((item) => {
pm.expect(item).to.have.property('product_id').that.is.a('number');
pm.expect(item).to.have.property('name').that.is.a('string');
pm.expect(item).to.have.property('description').that.is.a('string');
pm.expect(item).to.have.property('price').that.is.a('string');
pm.expect(item).to.have.property('category').that.is.a('string');
pm.expect(item).to.have.property('manufacturer').that.is.a('string');
pm.expect(item).to.have.property('imageUrl').that.is.a('string');
pm.expect(item).to.have.property('freeShipping').that.is.oneOf([0, 1]);
});
});
pm.test("Время ответа меньше, чем 2000ms", function () {
pm.expect(pm.response.responseTime).to.be.below(2000);
})
```

Tags

GET

Parameters

OS: Linux Mint 21.3 Cinnamon

Browser: Mozilla Firefox for Linux 134.0.1 (64-bit)

Postman: 11.28.2

Steps to reproduce

Step 1

Action Send a GET request.

Input data	<pre>GET method endpoint: {{base_URL}}/products/id/:productId Path Variables - Key: productId, Value: {{product_id}}</pre>
Expected result	The request was successfully sent to the server
Step	2
Action	Verify the response code.
Input data	
Expected result	Received code: 200 OK
Step	3
Action	Verify the response in the server body.
Input data	
Expected result	Example of the response received: { "id": 0, "name": "string", "description": "string", "price": 0, "category": "string", "manufacturer": "string", "imageUrl": "string", "freeShipping": true }

POST Add a new product

POST: Successfully added a new product to the product catalog.

PriorityBehaviorTypeLayer↑ HighPositiveFunctionalAPIIs FlakyMilestoneAutomationStatusNo-ManualActual

Is Muted

Nο

Pre-conditions

POST method endpoint: {{base_URL}}/add-product

Qa-Demoshopping Variables: base_URL, Value: https://qa.demoshopping.ru

Authorization: Inherit auth from parent: {{token}}

Headers - Content-Type: application/json

Environments: QA is active

Post-conditions

Post-response Script:

```
let responseText = pm.response.text(); // Получение текста ответа
let regex = /ID:s*(d+)/; // Регулярное выражение
let match = responseText.match(regex);
if (match) {
let productId = match[1]; // Извлечение числа
console.log("Extracted Product ID:", productId);
pm.environment.set("product id", productld); // Сохранение в переменной окружения
} else {
console.log("ID not found in response");
pm.test("Статус код равен 200", function () {
pm.response.to.have.status(200);
});
pm.test("Тело ответа содержит информацию о добавлении товара", function () {
pm.expect(pm.response.text()).to.include("Продукт успешно добавлен с ID:");
});
pm.test("Время ответа меньше, чем 2000ms", function () {
pm.expect(pm.response.responseTime).to.be.below(2000);
});
```

Tags

POST

Parameters

OS: Linux Mint 21.3 Cinnamon

Browser: Mozilla Firefox for Linux 134.0.1 (64-bit)

Postman: 11.28.2

Steps to reproduce

Step 1

Action Send a POST request.

```
POST method endpoint: {{base_URL}}/add-product
                                Body:
                                {
"name": "{{$randomProductName}}",
"description": "{{$randomPhrase}}",
"price": {{$randomPrice}},
Input data
                                "category": "{{$randomWord}}",
"manufacturer": "{{$randomCompanyName}}",
                                "imageUrl": "{{$randomUrl}}",
                                "freeShipping": {{$randomBoolean}}
Expected result
                                The request was successfully sent to the server
                                2
Step
Action
                                Verify the response code.
Input data
Expected result
                                Received code: 200 OK
                                3
Step
Action
                                Verify the response in the server body.
Input data
Expected result
                                Response "Продукт успешно добавлен с ID" is received.
```

PUT Full product update by ID

PUT: Full update of product information by ID

PriorityBehaviorTypeLayer↑ HighPositiveFunctionalAPIIs FlakyMilestoneAutomationStatusNo-ManualActual

Is Muted

No

Pre-conditions

PUT method endpoint: {{base_URL}}/products/id/:productId

Qa-Demoshopping Variables: base_URL, Value: https://qa.demoshopping.ru

Authorization: Inherit auth from parent: {{token}}

Headers - Content-Type: application/json

Environments: QA is active

Post-conditions

Post-response Script:

```
pm.test("Статус код равен 200", function () {
pm.response.to.have.status(200);
});

<br/>
br>
pm.test("Время ответа меньше, чем 2000ms", function () {
pm.expect(pm.response.responseTime).to.be.below(2000);
});
```

Tags

PUT

Parameters

OS: Linux Mint 21.3 Cinnamon

Browser: Mozilla Firefox for Linux 134.0.1 (64-bit)

Postman: 11.28.2

Expected result

Steps to reproduce

Step	1
Action	Send a PUT request.
	PUT method endpoint: {{base_URL}}/products/id/:productId
Input data	<pre>Body: { "name": "{{\$randomProductName}}", "description": "{{\$randomPhrase}}", "price": {{\$randomPrice}}, "category": "{{\$randomWord}}", "manufacturer": "{{\$randomCompanyName}}", "imageUrl": "{{\$randomUrl}}", "freeShipping": {{\$randomBoolean}} }</pre>

The request was successfully sent to the server

Step	2
Action	Verify the response code.
Input data	
Expected result	Received code: 200 OK
Step	3
Action	Verify the response in the server body.
Input data	
Expected result	Response "Товар обновлён" is received.

PATCH Partial product update by ID

PATCH: Partial update of product information by ID

Priority	Behavior	Туре	Layer
↑ High	Positive	Functional	API
ls Flaky	Milestone	Automation	Status
No		Manual	Actual

Is Muted

No

Pre-conditions

PATCH method endpoint: {{base_URL}}/products/id/:productId

Qa-Demoshopping Variables: base_URL, Value: https://qa.demoshopping.ru

Authorization: Inherit auth from parent: {{token}} **Path Variables -** Key: productId, Value: {{product_id}}

Headers - Content-Type: application/json

Environments: QA is active

Post-conditions

Post-response Script:

```
pm.test("Статус код равен 200", function () {
pm.response.to.have.status(200);
});
<br/>
<br/>
pm.test("Время ответа меньше, чем 2000ms", function () {
pm.expect(pm.response.responseTime).to.be.below(2000);
});
```

Tags

PATCH

Parameters

OS: Linux Mint 21.3 Cinnamon

Browser: Mozilla Firefox for Linux 134.0.1 (64-bit)

Postman: 11.28.2

Expected result

Steps to reproduce

Step	1	
Action	Send a PATCH request.	
	PATCH method endpoint: {{base_URL}}/products/id/:productId Path Variables - Key: productId, Value: {{product_id}}	
Input data	<pre>Body: { "name": "<string>", "description": "<string>", "price": {{\$randomPrice}}, "category": "<string>", "manufacturer": "<string>", "imageUrl": "<uri>", "freeShipping": {{\$randomBoolean}} }</uri></string></string></string></string></pre>	

The request was successfully sent to the server

Step	2
Action	Verify the response code.
Input data	
Expected result	Received code: 200 OK
Step	3
Action	Verify the response in the server body.
Input data	
Expected result	Response "Товар частично обновлён" is received.

DELETE Remove product by ID

DELETE remove a product from the catalog by ID

PriorityBehaviorTypeLayer↑ HighPositiveFunctionalAPIIs FlakyMilestoneAutomationStatusNo-ManualActual

Is Muted

No

Pre-conditions

DELETE method endpoint: {{base_URL}}/products/id/:productId

Qa-Demoshopping Variables: base_URL, Value: https://qa.demoshopping.ru

Authorization: Inherit auth from parent: {{token}}
Path Variables - Key: productId, Value: {{product_id}}

Environments: QA is active

Post-conditions

Post-response Script:

```
pm.test("Статус код равен 200", function () {
pm.response.to.have.status(200);
});
<br/>
<br/>
pm.test("Время ответа меньше, чем 2000ms", function () {
pm.expect(pm.response.responseTime).to.be.below(2000);
});
```

Tags

DELETE

Parameters

OS: Linux Mint 21.3 Cinnamon

Browser: Mozilla Firefox for Linux 134.0.1 (64-bit)

Postman: 11.28.2

Steps to reproduce

Step	1
Action	Send a DELETE request.
Input data	
Expected result	The request was successfully sent to the server
Step	2
Action	Verify the response code.
Input data	
Expected result	Received code: 200 OK

Step	3
Action	Verify the response in the server body.
Input data	
Expected result	Response "Товар удалён" is received.