[GET] Find products by category

SeverityPriorityBehaviorTypeO Normal↑ HighPositiveFunctionalLayerIs FlakyMilestoneAutomationNot setNo-Manual

Status Is Muted

Actual No

Pre-conditions

Headers request:

Content-Type: application/json Accept: application/json

Parameters

category: Laptops, Watches, Phones

Steps to reproduce

Step	1		
Action	Send the GET request https://qa.demoshopping.ru/products/FindByCategory?category={string}		
Input data	category (from Parameters)		
Expected result	Request has been successfully sent		
Step	2		
Action	Check the status code		
Input data			
Expected result	HTTP Status: 200 OK		
Step	3		
Action	Check the response body		
Input data	Requirements: https://qa.demoshopping.ru/api-docs/#/Products/get_products_FindByCategory		
Expected result	The response body is represented as an array with objects: [{ "id": 0, "name": "string", "description": "string", "price": 0, "category": "string", "manufacturer": "string", "imageUrl": "string", "freeShipping": true }]		

[POST] Error code 500: Add product to the Cart with non-existent product ID

SeverityPriorityBehaviorTypeO NormalO MediumNegativeFunctional

Layer Is Flaky Milestone Automation

Not set No - Manual

Status Is Muted

Actual No

Pre-conditions

Headers request:

Content-Type: application/json Accept: application/json Authorization: Bearer {token}

Steps to reproduce

Step	1		
Action	Send the POST request https://qa.demoshopping.ru/cart		
Input data	Requirements: https://qa.demoshopping.ru/api-docs/#/Cart/post_cart		
	Postman: Body > Raw > JSON		
	<pre>productId = integer // {non-existent ID} quantity = integer</pre>		
	Test sample body:		
	1 "productId": 1111111111,		
	"quantity": 5 }		
Expected result	Request has been successfully sent.		
Step	2		
Action	Check the status code		
Innut data			
Input data			
Expected result	HTTP Status: 500 Internal Server Error.		
Ston	٦		
Step	3		
Action	Check the response body		
Input data			
Expected result	The message "Ошибка сервера при добавлении товара в корзину" has been received.		

[PUT] Full product update by ID

SeverityPriorityBehaviorTypeO Normal↑ HighPositiveFunctionalLayerIs FlakyMilestoneAutomationNot setNo-Manual

Status Is Muted

Actual No

Pre-conditions

Headers request:

Content-Type: application/json Accept: application/json

Steps to reproduce

Step 1

Action Send the PUT request https://qa.demoshopping.ru/products/id/{productId}

Requirements: https://qa.demoshopping.ru/api-docs/#/Products/put_products_id_productld

Postman: Body > Raw > JSON

- product_id = integer
- name = string
- description = string
- price = number
- category = string
- manufacturer = string
- imageUrl = string
- freeShipping = boolean

Input data

Test sample body:

"product_id": 11643,

"name": "APPLE WATCH XL",

"description": "NEW GENERATION OF APPLE WATCHES",

"price": 199.99,

"category": "Watches",
"manufacturer": "Apple",

"imageUrl": "https://m.media-amazon.com/images/I/810ji95ZzVS._AC_SX679_.jpg",

"freeShipping": false

}

Expected result Request has been successfully sent.

Step 2

Action Check the status code

Input data

Expected result HTTP Status: 200 OK.

Step 3

Action Check the response body

Input data

Expected result The message "Товар обновлён" has been received.

[PATCH] Change product quantity in order

Severity	Priority	Behavior	Туре
O Normal	↑ High	Positive	Functional
Layer	Is Flaky	Milestone	Automation
Not set	No	-	Manual
Status	Is Muted		

Pre-conditions

Actual

Send requests to add the product to the Cart and create the Order:

No

```
body:
{
"productId": {productId},
"quantity": 1
}
2. [POST] https://qa.demoshopping.ru/orders
take data from response: {orderId}
Headers request:
```

1. [POST] https://qa.demoshopping.ru/cart

Content-Type: application/json Accept: application/json Authorization: Bearer {token}

Steps to reproduce

Step	1
Action	Send the PATCH request https://qa.demoshopping.ru/orders/{orderId}/products/{productId}
	Requirements: Raw > JSON quantity = integer Test sample:
Input data	Path: {productId} = 11643 {orderId} = 2362
	Body: { "quantity": 2 }

Expected result Request has been successfully sent.

```
Step 2

Action Check the status code

Input data

Expected result HTTP Status: 200 OK.
```

Step 3

Action Check the response body

Input data

Expected result The message "Количество товара обновлено в заказе" has been received.

[DELETE] Error code 403: Delete the product that cannot be deleted

SeverityPriorityBehaviorType○ Normal↑ HighNegativeOther

Layer Is Flaky Milestone Automation

Not set No - Manual

Status Is Muted

Actual No

Steps to reproduce

Step 1

Action Send the DELETE request https://qa.demoshopping.ru/products/id/{undeletableProductId}

Requirements: https://qa.demoshopping.ru/api-docs/#/Products/delete_products_id_productId

Input data Test sample:

Path:

 $\{undeletableProductId\} = 15$

Expected result Request has been successfully sent.

Step 2

Action Check the status code

Input data

Expected result HTTP Status: 403 Forbidden

Step 3

Action Check the response body

Input data

Expected result The message "Удаление этого товара запрещено" has been received.