GET: Search for a product by ID

Severity **Priority Behavior Type** Positive Major ↑ High Smoke

Is Flaky Milestone **Automation** Layer

Manual

No Status Is Muted

Actual No

Pre-conditions

Not set

Headers request: Content-type: application/json. Accept: application/json.

Parameters

Product ID correct: 2315

Steps to reproduce

Step	1
Action	Send GET request: https://qa.demoshopping.ru/products/id/{{Product ID correct}}
Input data	
Expected result	The request is sent to the server
Step	2
Action	Check status code
Input data	
Expected result	HTTP status: 200
Step	3
Action	Check that response body contains data about the product (product id, name, description, price, category, m anufacturer, imageUrl, freeShipping).
Input data	

Input data

The response body contains the below information:

```
{
    "product_id": 2315,
        "" "string",
                 "name": "string",
                "description": "string",
Expected
                 "price": "0.00",
"category": "string",
result
                 "manufacturer": "string",
                 "imageUrl": "string",
                 "freeShipping": 1
```

}



POST: Adding a new product

SeverityPriorityBehaviorType♠ Critical↑ HighPositiveSmoke

Layer Is Flaky Milestone Automation

Manual

Status Is Muted

No

Actual No

Pre-conditions

Not set

Headers request: Content-type: application/json. Accept: application/json.

Parameters

Url: https://qa.demoshopping.ru

Step	1
Action	Send POST request <u>{{Url}}/</u> add-product with specifying product parameters in the request body
Input data	{ "name": "{{\$randomProductName}}", "description": "{{\$randomProductAdjective}}", "price":{{\$randomInt}}, "category": "Electronics", "manufacturer": "Example Manufacturer", "imageUrl": "{{\$randomImageUrl}}", "freeShipping": true }
Expected resul	t The request is sent to the server
Step	2
Action	Check status code
Input data	
Expected resul	t HTTP status: 200
Step	3
Action	Check that response body contains data about the new product with id
Input data	
Expected resul	t The response body: "Продукт успешно добавлен с ID: 2405"

PUT: Complete product update by ID

Priority Severity **Behavior** Type Positive Smoke Major ↑ High Layer Is Flaky Milestone **Automation** Not set Manual No Status Is Muted Actual No

Pre-conditions

Headers request: Content-type: application/json. Accept: application/json.

Parameters

Product ID correct: 2315

Url: https://qa.demoshopping.ru

Step	1
Action	Send PUT request ${\{Url\}}/products/id/{\{Product\ ID\ correct\}\}}$ with specifying product parameters in the request body
Input data	{ "name": "string", "description": "string", "price": 0, "category": "string", "manufacturer": "string", "imageUrl": "string", "freeShipping": true }
Expected result	The request is sent to the server
Step	2
Action	Check status code
Input data	
Expected result	HTTP status: 200
Step	3
Action	Check that response body contains data about complete product update
Input data	
Expected result	The response body: "Товар обновлен"

DELETE: deleting a product by ID

Severity **Priority Behavior** Type ↑ High Positive Smoke Major Layer Is Flaky Milestone **Automation** Not set Manual No Status Is Muted Actual No

Pre-conditions

Headers request: Content-type: application/json. Accept: application/json.

Parameters

Product ID correct: 2315

Url: https://qa.demoshopping.ru

Step	1
Action	Send DELETE request {{Url}}/products/id/{{Product ID correct}}
Input data	
Expected result	The request is sent to the server
Step	2
Action	Check status code
Input data	
Expected result	HTTP status: 200
Step	3
Action	Check that response body contains data about the deleting of the product
Input data	
Expected result	Response body: "Товар успешно удален"

PATCH: Partial product update by ID

SeverityPriorityBehaviorType^ Major^ HighPositiveSmoke

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.

Parameters

Url: https://qa.demoshopping.ru

Product ID correct: 2315

Step	1
Action	Send PATCH request ${\{Url\}}/products/id/\{\{Product\ ID\ correct\}\}\ with specifying product\ parameters in the request body$
Input data	{ "name": "string", "description": "string", "price": 0, "category": "string", "manufacturer": "string", "imageUrl": "string", "freeShipping": true }
Expected result	The request is sent to the server
Step	2
Action	Check status code
Input data	
Expected result	HTTP status: 200
Step	3
Action	Check that response body contains data about partial product update
Input data	
Expected result	The response body: "Товар частично обновлен"