GET - Отображение списка всех продуктов

Используемые тестовые данные Swagger - https://qa.demoshopping.ru/api-docs/#/Products/get_products

PriorityBehaviorTypeLayer↑ HighPositiveFunctionalNot setIs FlakyMilestoneAutomationStatusNo-ManualActual

Is Muted

No

Tags

GET

Parameters

baseURL: https://qa.demoshopping.ru

Environments: Windows 10 Home version 22H2, Postman for Windows Version 11.28.4, Environments

Postman - QA server

Step	1
Action	Отправить GET запрос.
Input data	{{baseURL}}/products
Expected result	Запрос успешно отправлен на сервер.
Step	2
Action	Проверить код состояния.
Input data	
Expected result	Ответ запроса 200 ОК.
Step	3
Action	Проверить тело ответа от сервера.
Input data	

Тело ответа в формате JSON (сформированный объект JSON, в массиве) возвращается от сервера и буд ет иметь следующий вид:

```
Expected result
```

```
{
    "id": number,
    "name": "string",
    "description": "string",
    "price": number,
    "category": "string",
    "manufacturer": "string",
    "imageUrl": "string",
    "freeShipping": boolean
}
```

Каждое поле JSON объекта соответствует аргументу метода сервиса.

Step	4
Action	Проверить структуру ответа.
Input data	
Expected result	Cxema JSON отображена корректно, значения полей соответствуют ожидаемым значениям из тестовых данных.
Step	5
Action	Проверить что Content-Type в Headers - application/json.
Input data	
Expected result	В ответе запроса Content-Type в Headers - application/json.

POST - Добавление нового продукта в каталог

Используемые тестовые данные Swagger - https://qa.demoshopping.ru/api-docs/#/Products/get_products

PriorityBehaviorTypeLayer↑ HighPositiveFunctionalNot setIs FlakyMilestoneAutomationStatusNo-ManualActual

Is Muted

No

Tags

POST

Parameters

baseURL: https://qa.demoshopping.ru Environments: Windows 10 Home version 22H2, Postman for Windows Version 11.28.4, Environments Postman - QA server

Step	1
Action	В запросе POST во вкладке "Body" написать body запроса в JSON с корректными значениями для всех параметров.
Input data	{ "name": "Example Product", "description": "This is an example product.", "price": 100, "category": "Electronics", "manufacturer": "Example Manufacturer", "imageUrl": "http://example.com/image.jpg", "freeShipping": true } ,
Expected result	Данные успешно записаны и отображаются в body в JSON.
Step	2
Action	Отправить POST запрос.
Input data	{{baseURL}}/add-product
Expected result	Запрос успешно отправлен на сервер.

Step	3
Action	Проверить код состояния.
Input data	
Expected result	Ответ запроса 200 ОК.
Step	4
Action	Проверить тело ответа от сервера.
Input data	
Expected result	Тело ответа в формате HTML. С сообщением "Продукт успешно добавлен с ID: (ID товара)".
Step	5
Action	Проверить что Content-Type в Headers -text/html.
Input data	
Expected result	В ответе запроса Content-Type в Headers - text/html.
Step	6
Action	Проверить наличие переменной в запросе GET. Params>Path Variables>{{prodictld}}.
Input data	
Expected result	В Path Variables есть переменная {{prodictId}}.
Step	7
Action	Отправить запрос GET - поиск товара по ID.
Action Input data	Отправить запрос GET - поиск товара по ID. {{baseURL}}/products/id/:productId
Input data Expected	{{baseURL}}/products/id/:productId
Input data Expected result	{{baseURL}}/products/id/:productId Запрос успешно отправлен на сервер.
Input data Expected result Step	{{baseURL}}/products/id/:productId Запрос успешно отправлен на сервер. 8

}

PUT - Полное обновление информации о продукте

Priority	Behavior	Туре	Layer
↑ High	Positive	Functional	Not set
Is Flaky	Milestone	Automation	Status
No	-	Manual	Actual

Is Muted

No

Tags

PUT

Parameters

baseURL: https://qa.demoshopping.ru

Environments: Windows 10 Home version 22H2, Postman for Windows Version 11.28.4, Environments

Postman - QA server

Step	1	
Action	Проверить наличие переменной в Params>Path Variables>{{prodictId}}.	
Input data		
Expected result	В Path Variables есть переменная {{prodictld}}.	
Step	2	
Action	В запросе POST во вкладке "Body" написать body запроса в JSON с корректными значениями для всех параметров.	
Input data	{ "name": "test", "description": "test", "price": 150, "category": "string", "manufacturer": "string", "imageUrl": "string", "freeShipping": true } , где "name" - string "description" - string "price" - number "category" - string "manufacturer" - string "manufacturer" - string "imageUrl" - string "imageUrl" - string "freeShipping" - boolean	
Expected result	Данные успешно записаны и отображаются в body в JSON.	
Step	3	

Action	Отправить PUT запрос.
Input data	{{baseURL}}/products/id/:productId
Expected result	Запрос успешно отправлен на сервер.
Step	4
Action	Проверить код состояния.
Input data	
Expected result	Ответ запроса 200 ОК.
Step	5
Action	Проверить тело ответа от сервера.
Input data	
Expected result	Тело ответа в формате HTML. С сообщением "Товар обновлён".
Step	6
Action	Проверить что Content-Type в Headers - text/html.
Input data	
Expected result	В ответе запроса Content-Type в Headers - text/html.
Step	7
Action	Проверить наличие переменной в запросе GET. Params>Path Variables>{{prodictId}}.
Input data	
Expected result	В Path Variables есть переменная {{prodictld}}.
Step	8
Action	Отправить запрос GET - поиск товара по ID.
Input data	{{baseURL}}/products/id/:productId
Expected result	Запрос успешно отправлен на сервер.
Step	9

Action	Проверить код состояния.	
Input data		
Expected result	Ответ запроса 200 ОК.	
Step	10	
Action	Проверить тело ответа от сервера.	
Input data		
Expected result	Тело ответа в формате JSON. Отображается вся информация о обновленном товаре с корректными зна чениями параметров. { "product_id": number, "name": string, "price": number, "category": string, "manufacturer": string, "imageUrl": string, "freeShipping": boolean }	

РАТСН - Частичное обновление информации о названии продукта

Используемые тестовые данные Swagger - https://qa.demoshopping.ru/api-docs/#/Products/get_products

Priority	Behavior	Туре	Layer
↑ High	Positive	Functional	Not set
In Fining			
Is Flaky	Milestone	Automation	Status

Is Muted

No

Tags

PATCH

Parameters

baseURL: https://qa.demoshopping.ru

Environments: Windows 10 Home version 22H2, Postman for Windows Version 11.28.4, Environments

Postman - QA server

Step	1
Action	Проверить наличие переменной в Params>Path Variables>{{prodictId}}.
Input data	
Expected result	B Path Variables есть переменная {{prodictId}}.
Step	2
Action	В запросе PATCH во вкладке "Body" написать body запроса в JSON с корректными значениями для ну жных параметров.
Input data	{ "name": "test" }
Expected result	Данные успешно записаны и отображаются в body в JSON.
Step	3
Action	Отправить РАТСН запрос.
Input data	{{baseURL}}/products/id/:productId
Expected result	Запрос успешно отправлен на сервер.

Step	4
Action	Проверить код состояния.
Input data	
Expected result	Ответ запроса 200 ОК.
Step	5
Action	Проверить тело ответа от сервера.
Input data	
Expected result	Тело ответа в формате HTML. С сообщением "Товар частично обновлён".
Step	6
Action	Проверить что Content-Type в Headers - text/html.
Input data	
Expected result	В ответе запроса Content-Type в Headers - text/html.
Step	7
Action	Проверить наличие переменной в запросе GET. Params>Path Variables>{{prodictId}}.
Input data	
Expected result	В Path Variables есть переменная {{prodictId}}.
Step	8
Action	Отправить запрос GET - поиск товара по ID.
Input data	
Expected result	Запрос успешно отправлен на сервер.
Step	9
Action	Проверить код состояния.
Input data	
Expected result	Ответ запроса 200 ОК.

DELETE - Удаление товара из каталога

Используемые тестовые данные Swagger - https://qa.demoshopping.ru/api-docs/#/Products/get_products

PriorityBehaviorTypeLayer↑ HighPositiveFunctionalNot setIs FlakyMilestoneAutomationStatusNo-ManualActual

Is Muted

No

Tags

DELETE

Parameters

baseURL: https://qa.demoshopping.ru

Environments: Windows 10 Home version 22H2, Postman for Windows Version 11.28.4, Environments

Postman - QA server

Step	1
Action	Проверить наличие переменной в запросе DELETE. Params>Path Variables>{{prodictId}}.
Input data	
Expected result	B Path Variables есть переменная {{prodictId}}.
Step	2
Action	Отправить DELETE запрос.
Input data	{{baseURL}}/products/id/:productId
Expected result	Запрос успешно отправлен на сервер.
Step	3
Action	Проверить код состояния.
Input data	
Expected result	Ответ запроса 200 ОК.
Step	4
Action	Проверить тело ответа от сервера.
Input data	

Expected result	тело ответа в формате ттиви о сосощением товар удалент
Step	5
Action	Проверить что Content-Type в Headers - text/html.
Input data	
Expected result	В ответе запроса Content-Type в Headers - text/html.
Step	6
Action	Проверить наличие переменной в запросе GET. Params>Path Variables>{{prodictId}}.
Input data	
Expected result	В Path Variables есть переменная {{prodictId}}.
Step	7
Action	Отправить запрос GET - поиск товара по ID.
Input data	{{baseURL}}/products/id/:productId
Expected result	Запрос успешно отправлен на сервер.
Step	8
Action	Проверить код состояния.
Input data	
Expected result	Ответ запроса 404 Not Found.
Step	9
Action	Проверить тело ответа от сервера.
Input data	
Expected result	Тело ответа в формате HTML. С сообщением "Товар с таким ID не найден".

Expected result Тело ответа в формате HTML. С сообщением "Товар удалён".