GET. Поиск товаров с бесплатной доставкой

Требования к запросу: https://ga.demoshopping.ru/api-docs/#/Products/get_products_FindByShipping

PriorityBehaviorTypeLayer↑ HighPositiveFunctionalNot setIs FlakyMilestoneAutomationStatusNo-ManualActual

Is Muted

No

Pre-conditions

Требования к заголовкам:

Authorization: Inherit auth from Parent, Bearer Token {{vault:authorization-secret}}}

Cache-Control: no-cache

Postman-Token: <calculated when request is sent>

Host: <calculated when request is sent> **User-Agent:** PostmanRuntime/7.43.0

Accept: */*

Accept-Encoding: gzip, deflate, br

Connection: keep-alive

Parameters

Операционная система: Windows 11 Pro 24H2 26100.2894

Postman: Desktop App v11.28.4

Steps to reproduce

Step	1
Action	Отправить GET-запрос на URL https://qa.demoshopping.ru/products/FindByShipping , предварительно указ ав дополнительные параметры
Input data	Параметры (Query Params): ключ (key) — freeShipping значение (value) — true
Expected result	Запрос успешно отправлен на сервер
Step	2
Action	Проверить код состояния
Input data	
Expected result	HTTP Status: 200 OK
Step	3
Action	Проверить тело ответа от сервера

Input data

Тело ответа возвращается в формате JSON и имеет следующий вид:

```
Expected result
```

```
{
    "id": integer,
    "name": "string",
    "description": "string",
    "price": integer,
    "category": "string",
    "manufacturer": "string",
    "imageUrl": "string",
    "freeShipping": true
    }
}
```

Step 4

Action Убедиться, что в теле ответа представлены только товары с бесплатной доставкой

Input data

Expected result

Ключи "freeShipping" имеют значения true. Значения false отсутствуют в ответе.

POST. Успешная регистрация нового пользователя

Требования к запросу: https://qa.demoshopping.ru/api-docs/#/Users/post_register

PriorityBehaviorTypeLayer↑ HighPositiveFunctionalNot setIs FlakyMilestoneAutomationStatusNo-ManualActual

Is Muted

No

Pre-conditions

Требования к заголовкам:

Authorization: Inherit auth from Parent, Bearer Token {{vault:authorization-secret}}

Cache-Control: no-cache

Postman-Token: <calculated when request is sent>

Content-Type: application/json

Content-Length: <calculated when request is sent>

Host: <calculated when request is sent> **User-Agent:** PostmanRuntime/7.43.0

Accept: */*

Accept-Encoding: gzip, deflate, br

Connection: keep-alive

Parameters

Операционная система: Windows 11 Pro 24H2 26100.2894

Postman: Desktop App v11.28.4

Steps to reproduce

Step 1

Action Отправить POST-запрос на URL https://qa.demoshopping.ru/register, предварительно добавив скрипт в

данный запрос

```
Postman: Body → Raw → JSON
                "username": "{{valrandomUsername}}",
                "password": "{{valrandomPassword}}"
               Postman: Body → Scripts → Pre-request
               // Функция для генерации случайного логина
               function generateUsername() {
                 const length = Math.floor(Math.random() * (15 - 3 + 1)) + 3; // Длина от 3 до 15
                 const chars = "abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789 "; // Допуст
               имые символы
                 let username = "";
                 for (let i = 0; i < length; i++) {
                    username += chars.charAt(Math.floor(Math.random() * chars.length));
                 return username;
               }
               // Функция для генерации случайного пароля
               function generatePassword() {
                 const minLength = 8; // Минимальная длина пароля
                 const additionalLength = Math.floor(Math.random() * 10); // Дополнительная случайная длина (напри
               мер, от 0 до 10)
                 const length = minLength + additionalLength; // Итоговая длина пароля
                 const letters = "abcdefqhijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ"; // Буквы
                 const digits = "0123456789"; // Цифры
                 const allChars = letters + digits; // Все допустимые символы
                 let password = "";
                 let hasLetter = false;
                 let hasDigit = false;
Input data
                 // Генерация символов пароля
                 for (let i = 0; i < length; i++) {
                    const char = allChars.charAt(Math.floor(Math.random() * allChars.length));
                    password += char;
                   // Проверка на наличие хотя бы одной буквы и цифры
                    if (letters.includes(char)) hasLetter = true;
                    if (digits.includes(char)) hasDigit = true;
                 // Гарантируем наличие хотя бы одной буквы и одной цифры
                 if (!hasLetter) {
                    password = password.substring(1) + letters.charAt(Math.floor(Math.random() * letters.length));
                 if (!hasDigit) {
                    password = password.substring(0, password.length - 1) + digits.charAt(Math.floor(Math.random() * di
               gits.length));
                 // Перемешивание символов для случайного порядка
                 password = password.split("").sort(() => Math.random() - 0.5).join("");
                 return password;
               // Генерация логина и пароля
               let valrandomUsername = generateUsername();
               let valrandomPassword = generatePassword();
               // Сохранение данных в переменных окружения
               pm.environment.set("valrandomUsername", valrandomUsername); pm.environment.set("valrandomPassword", valrandomPassword);
               console.log("Сгенерированные данные:");
               console.log("Логин:", valrandomUsername);
               console.log("Пароль:", valrandomPassword);
Expected
               Запрос успешно отправлен на сервер
result
Step
               2
Action
               Проверить код состояния
Input data
Expected
               HTTP Status: 200 OK
result
```

Step	3
Action	Проверить тело ответа от сервера
Input data	
Expected result	Тело ответа возвращается в формате JSON и имеет следующий вид: { "message": "Регистрация выполнена успешно" }

PUT. Попытка обновления по ID данных товара с вводом значения из спецсимволов для ключа "price"

Требования к запросу: https://ga.demoshopping.ru/api-docs/#/Products/put_products id_productId

PriorityBehaviorTypeLayer↑ HighNegativeFunctionalNot setIs FlakyMilestoneAutomationStatusNo-ManualActual

Is Muted

No

Pre-conditions

Значение {{productId}}} добавлено в переменные окружения / коллекции.

Требования к заголовкам:

Authorization: Inherit auth from Parent, Bearer Token {{vault:authorization-secret}}}

Cache-Control: no-cache

Postman-Token: <calculated when request is sent>

Content-Type: application/json

Content-Length: <calculated when request is sent>

Host: <calculated when request is sent> **User-Agent:** PostmanRuntime/7.43.0

Accept: */*

Accept-Encoding: gzip, deflate, br

Connection: keep-alive

Parameters

Операционная система: Windows 11 Pro 24H2 26100.2894

Postman: Desktop App v11.28.4

Steps to reproduce

Step	1
Action	Отправить PUT-запрос на URL https://qa.demoshopping.ru/products/id/{{productId}} }
Input data	Postman: Body → Raw → JSON { "name": "King Charles III", "description": "Ochen star", "price": "***+++" , "category": "United Kingdom", "manufacturer": "House of Windsor", "imageUrl": "https://static01.nytimes.com/newsgraphics/2023-04-04-coronation-faq/cca9949776c5b6851839b d363864e113d4078b21/_big_assets/images/king-head-460_x2.png", "freeShipping": false }
Expecte result	ed Запрос успешно отправлен на сервер

Step 2

Action Проверить код состояния

Input data	
Expecte result	HTTP Status: 400 Bad Request
Step	3
Action	Проверить тело ответа от сервера
Input data	
Expecte result	ed ^{Тело} ответа возвращается в формате HTML и имеет следующий вид: Неправильный тип значения для 'price'

РАТСН. Обновление количества товара в корзине авторизованного пользователя на случайное число единиц

Требования к запросу: https://qa.demoshopping.ru/api-docs/#/Cart/patch_cart_cart/tem/d

PriorityBehaviorTypeLayer↑ HighPositiveFunctionalNot setIs FlakyMilestoneAutomationStatusNo-ManualActual

Is Muted

No

Pre-conditions

В корзину добавлен любой товар. Значение {{cartItemId}} добавлено в переменные окружения / коллекции.

Требования к заголовкам:

Authorization: Inherit auth from Parent, Bearer Token {{vault:authorization-secret}}

Cache-Control: no-cache

Postman-Token: <calculated when request is sent>

Content-Type: application/json

Content-Length: <calculated when request is sent>

Host: <calculated when request is sent> **User-Agent:** PostmanRuntime/7.43.0

Accept: */*

Accept-Encoding: gzip, deflate, br

Connection: keep-alive

Parameters

Операционная система: Windows 11 Pro 24H2 26100.2894

Postman: Desktop App v11.28.4

Steps to reproduce

Step	1
Action	Отправить PATCH-запрос на URL https://qa.demoshopping.ru/cart/{{cartItemId}}
Input data	Postman: Body → Raw → JSON { "quantity": {{randomQuantity}}} } Postman: Body → Scripts → Pre-request // Генерация случайного количества товара (от 1 до 999) let randomQuantity = Math.floor(Math.random() * (999 - 1 + 1)) + 1; // Генерация случайного productId (от 1 до 50) let productId_original = Math.floor(Math.random() * (50 - 1 + 1)) + 1; // Сохранение значений в переменные окружения pm.environment.set("productId_original", productId_original); pm.environment.set("randomQuantity", randomQuantity); // Логирование значений для проверки console.log("Случайный productId:", productId_original); console.log("Случайное количество:", randomQuantity);
Expected result	Запрос успешно отправлен на сервер

Step

Action	Проверить код состояния
Input data	
Expected result	HTTP Status: 200 OK
Step	3
Action	Проверить тело ответа от сервера
Input data	
	Тело ответа возвращается в формате JSON и имеет следующий вид:
Expected result	{ "message": "Количество товара в корзине обновлено" }

DELETE. Попытка удаления товара из заказа неавторизованным пользователем

Требования к запросу: https://qa.demoshopping.ru/api-docs/#/Orders/delete_ordersorderld productId

PriorityBehaviorTypeLayer↑ HighNegativeFunctionalNot setIs FlakyMilestoneAutomationStatusNo-ManualActual

Is Muted

No

Pre-conditions

Из товаров в корзине пользователя создан заказ./Значения {{order_ld}} и {{order_product_ld}} добавлены в
переменные окружения / коллекции.//span>

Во вкладке Authorization выбраны следующие параметры: Auth Type → No Auth.

Требования к заголовкам:

Cache-Control: no-cache

Postman-Token: <calculated when request is sent>

Host: <calculated when request is sent> **User-Agent:** PostmanRuntime/7.43.0

Accept: */*

Accept-Encoding: gzip, deflate, br

Connection: keep-alive

Parameters

Операционная система: Windows 11 Pro 24H2 26100.2894

Postman: Desktop App v11.28.4

Steps to reproduce

Step	1
Action	Отправить DELETE-запрос на URL

Step

3

Action	Проверить тело ответа от сервера
Input data	
Expected result	Тело ответа возвращается в формате HTML и имеет следующий вид: Токен не предоставлен.