**В коллекцию включены следующие запросы:**

- POST-запрос на добавление товара в корзину: /cart/product

- PUT-запрос на изменение количества товаров в корзине: /cart

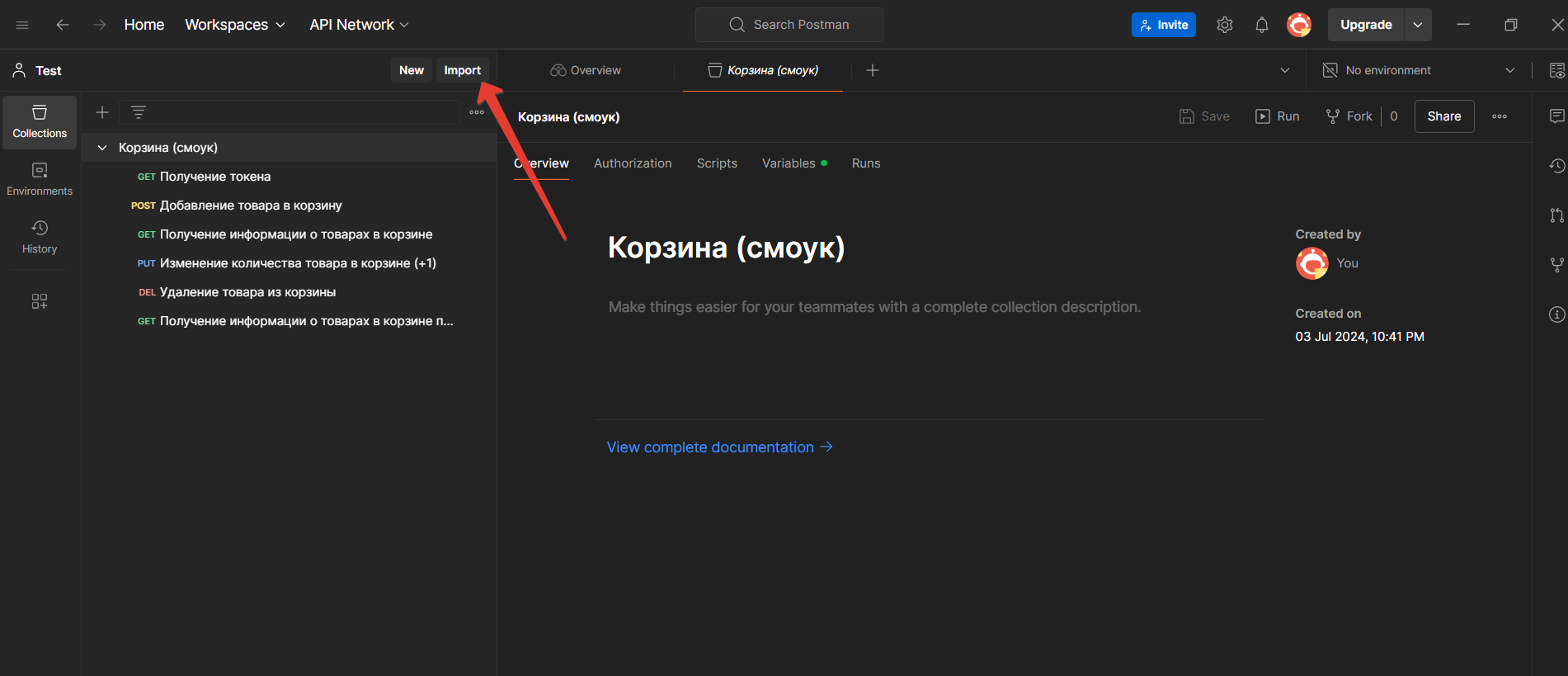
- DELETE-запрос на удаление товаров из корзины: /cart

- GET-запрос на получение информации о товарах в корзине: /cart

**Инструкция по работе с коллекцией:**

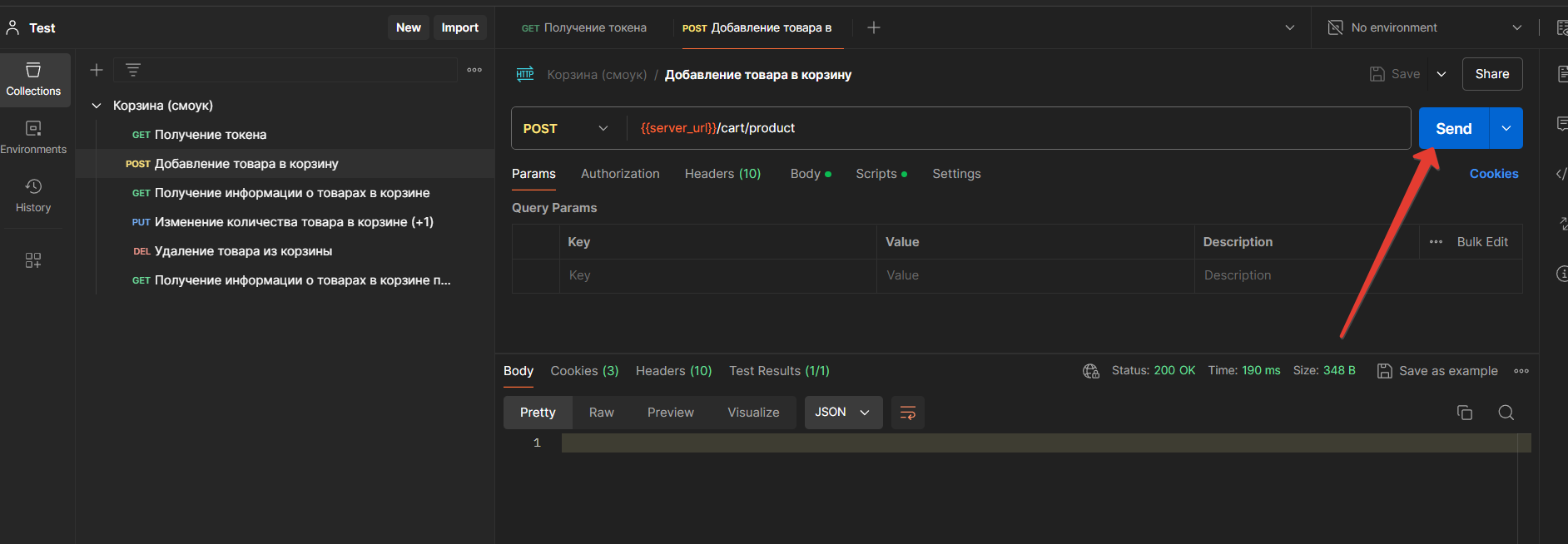
Для начала работы с коллекцией запросов по корзине, ее надо импортировать в Postman, предварительно установленный на компьютер.

Для этого в рабочем пространстве сверху надо нажать на кнопку import, где нужно будет выбрать коллекция сохраненную в формате json.

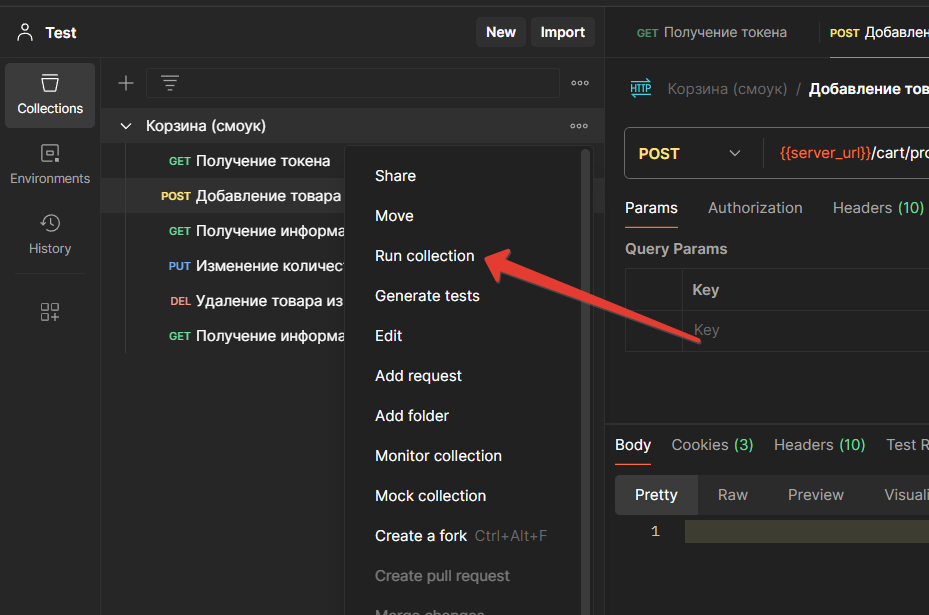


Слева представлены запросы, приведенные выше, объединенные в набор для смоук-тестирования.

Каждый запрос по отдельности можно отправить на сервер нажав на него, а затем нажать на кнопку Send.



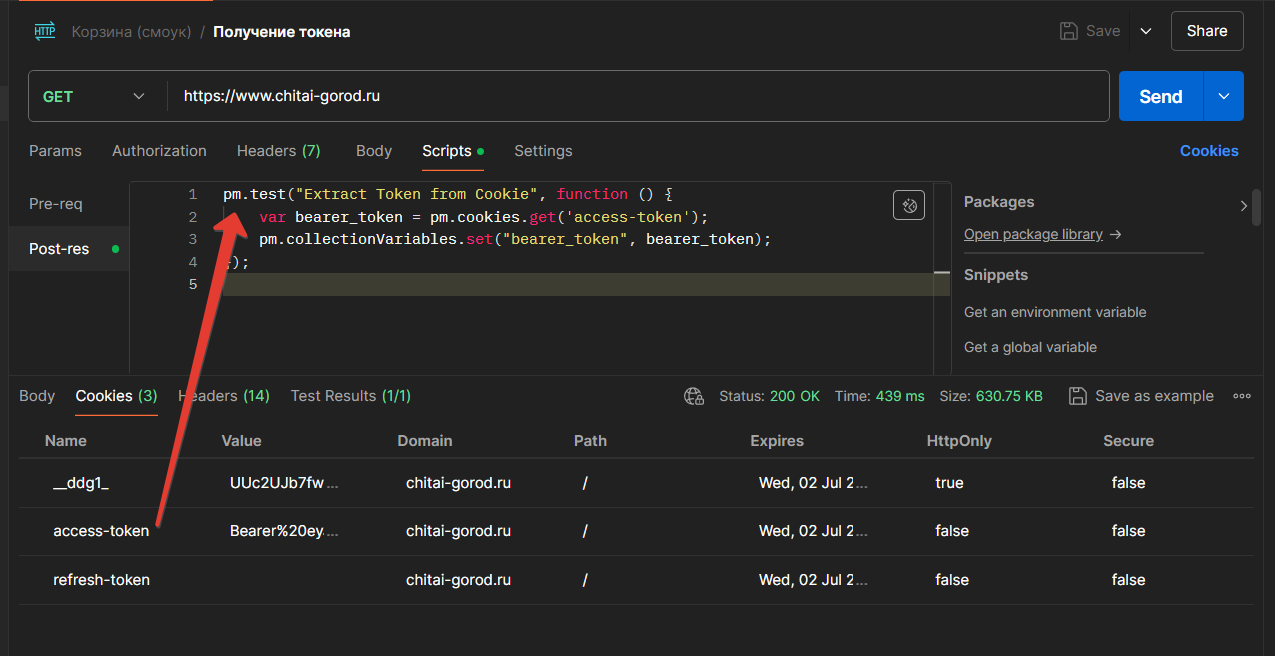
Чтобы последовательно отправить все запросы в коллекции и проверить результаты прогона надо в меню «…» коллекции нажать Run collection, где можно будет запустить все запросы за один прогон.



**Что необходимо для авторизации:**

Предварительно, перед отправкой запросов на сервер, необходимо получить токен авторизации, который будет использоваться в хедерах всех отправляемых запросов. Для этого в самом начале отправляется GET-запрос: <https://www.chitai-gorod.ru> . Из куки, полученной в ответе на него мы сохраняем access-token, который будет использоваться в остальных запросах.

Для этого после отправки запроса, выполнится скрипт, который сохранит токен в переменную коллекции:



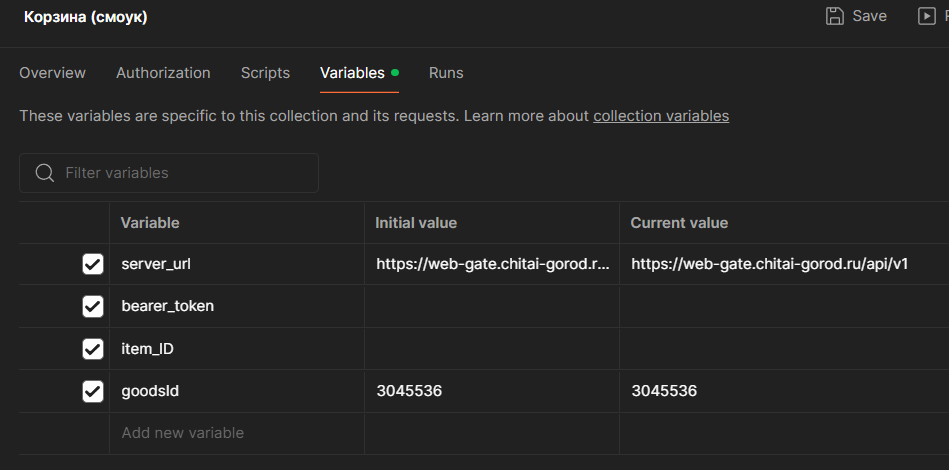
**Какие переменные используются:**

server\_url – урл читай-город

bearer\_token – токен, необходимый в хедерах для каждого запроса (получаем из куки в запросе на получение токена)

item\_ID – id позиции, которую добавили в корзину

goodsId – id выбранного товара, который хотим добавить в корзину. Равен 3045536, это книга «Болотник», можно задать любой другой id, интересующей нас книги.



**Какие скрипты используются:**

1. Для проверки, что от сервера приходит статус код 200:

pm.test("Status code is 200", function () {

    pm.response.to.have.status(200);

});

1. Для получения id позиции в корзине:

var responseJson = pm.response.json();

pm.collectionVariables.set("item\_ID", responseJson.products[0].id);

1. Для проверки, что наш товар в корзине

pm.test("Our good is in a cart", function () {

    var jsonData = pm.response.json();

    pm.expect(jsonData.products[0].goodsId).to.eql(3045536);

});

1. Для проверки, что товаров стало 2

pm.test("quantity of goods = 2", function () {

    var jsonData = pm.response.json();

    pm.expect(jsonData.products[0].quantity).to.eql(2);

});

1. Для проверки, что от сервера приходит статус код 204:

pm.test("Status code is 204", function () {

    pm.response.to.have.status(204);

});

1. Для проверки, что корзина пуста

pm.test("Quantity of goods is 0", function () {

    var jsonData = pm.response.json();

    pm.expect(jsonData.products).to.be.an('array').that.is.empty;

});