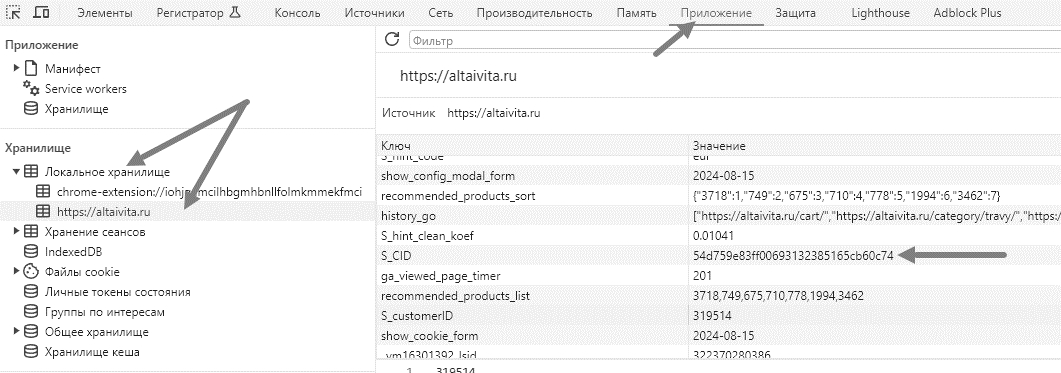
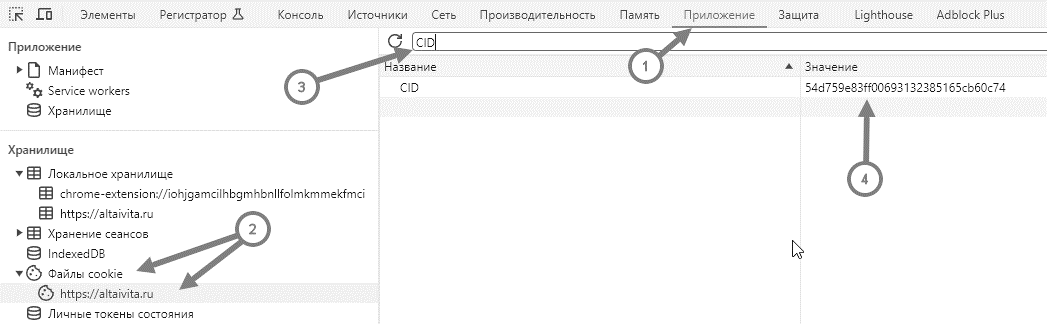
**Инструкция**

1. Войдите на сайт [https://altaivita.ru](https://altaivita.ru" \t "_new), используя свои учетные данные или тестовые (логин – xxxx@xxx.ru, пароль – xxxx).
2. Откройте консоль разработчика, перейдите в раздел "Приложения", выберите "Локальное хранилище", найдите сайт altaivita.ru и скопируйте значение S\_CID. Также это значение можно найти в разделе "Файлы cookie" для сайта altaivita.ru, выполнив поиск по ключу "CID".
3. Вставьте скопированное значение в переменную id\_session.
4. Теперь вы можете использовать коллекцию.





Коллекция включает следующие переменные:

1. base\_url – URL адрес.
2. id\_session – идентификатор сессии.
3. customer\_id – идентификатор покупателя (пользователя).
4. current\_sum\_quantity – текущее количество товаров в корзине.
5. quantity – количество товара, на которое следует увеличить или уменьшить общее количество в корзине.
6. product\_id – идентификатор товара, с которым проводится работа.
7. itemIDbyCart – динамическое значение, идентификатор товара, добавленного в корзину пользователем.

Коллекция содержит следующие папки с запросами:

1. Позитивные.
2. Негативные.
   * Добавление товара в корзину через превью товара.
   * Добавление товара в корзину через персональную карточку товара.
   * Изменение количества/удаление товаров.

**Папка “Позитивные”** содержит следующие запросы:

1. Добавление товара в корзину через превью товара.
2. Добавление товара в корзину через превью товара, язык английский, валюта доллар.
3. Удаление товара (уменьшение количества товара) через превью товара.
4. Добавление товара в корзину через персональную карточку товара.
5. Удаление товара (уменьшение количества товара) через карточку товара.
6. Изменение количества товара на странице корзины.
7. Удаление товара (всей категории) через кнопку в корзине.

Запросы содержат следующие тестовые скрипты (скрипты однотипные и адаптированы в зависимости от запроса):

1. **Проверка статус-кода ответа:**

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

pm.response.to.have.status(200);

});

**2.Проверка итогового количества товаров после операций добавления/уменьшения количества:**

var current\_sum\_quantity = pm.collectionVariables.get("current\_sum\_quantity");

var quantity = pm.collectionVariables.get("quantity");

quantity = parseInt(quantity);

pm.test("Товар был добавлен корректно", function () {

var jsonData = pm.response.json();

var expected\_sum\_quantity = (parseInt(current\_sum\_quantity) + quantity).toString();

pm.expect(jsonData.sum\_quantity).to.eql(expected\_sum\_quantity);

});

pm.collectionVariables.set("current\_sum\_quantity", pm.response.json().sum\_quantity);

1. **Установка значения товаров в корзине:**

var quantity = pm.collectionVariables.get("quantity");

pm.test("Новое значение товара было установлено корректно", function () {

var jsonData = pm.response.json();

pm.expect(jsonData.sum\_quantity).to.eql(quantity);

});

1. Проверка количества товаров после полного удаления категории:

pm.test("Проверяем, что sum\_quantity равно 0", function () {

var jsonData = pm.response.json();

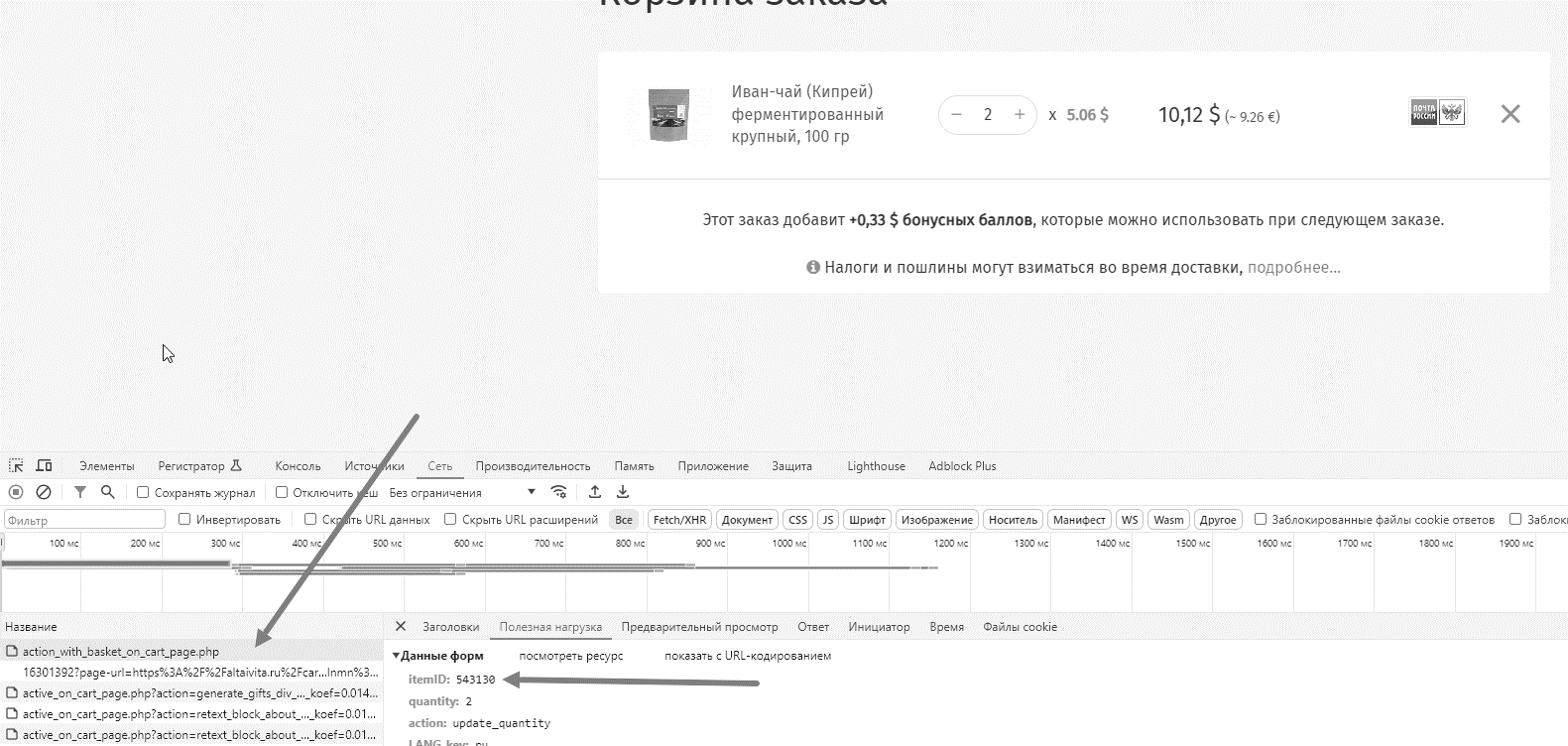
pm.expect(jsonData.sum\_quantity).to.eql("0");

});

Инструкция по работе с папкой:

Перед началом работы установите значение 0 для переменной current\_sum\_quantity.

При запуске всех тестов может возникнуть ошибка, однако остальные тесты должны пройти успешно. Это связано с тем, что значение itemIDbyCart не передается в ответе при добавлении товара в корзину. Отследите значение этой переменной через консоль: добавьте товар в корзину, затем перейдите в корзину и измените количество товара. Из консоли получите значение itemIDbyCart и установите его в переменную. После этого запросы должны пройти успешно.



**Папка “Негативные”.**

**Подраздел “Добавить товар в корзину через превью товара.”**

Подраздел содержит следующие запросы:

1. Добавить товар в корзину через превью товара по неверному id.
2. Добавить товар в корзину через превью товара, некорректный язык и валюта.
3. Добавить товар в корзину через превью товара, с пустым идентификатором сессии.
4. Добавить товар в корзину через превью товара, некорректный адрес запроса.
5. Добавить товар в корзину через превью товара, другой метод запроса.
6. Добавить товар в корзину через превью товара, количество товара превышает количество на складе.

Запросы содержат тестовые скрипты следующих видов (скрипты однотипные и адаптированы в мелких деталях под каждый запрос):

1. **Проверка статус-кода ответа:**

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

pm.response.to.have.status(200);

});

1. **Проверка содержание в ответе значения error для ключа status:**

pm.test("Проверяем, что status в ответе будет error", function () {

var jsonData = pm.response.json();

pm.expect(jsonData.status).to.eql("error");

});

1. **Проверка, что количество товаров будет отличаться в разных сессиях:**

var current\_sum\_quantity = pm.collectionVariables.get("current\_sum\_quantity");

var quantity = pm.collectionVariables.get("quantity");

quantity = parseInt(quantity);

pm.test("Товар с пустым идентификатором сессии успешно добавляется в корзину (количество товаров разное)", function () {

var jsonData = pm.response.json();

var expected\_sum\_quantity = (parseInt(current\_sum\_quantity) + quantity).toString();

pm.expect(jsonData.sum\_quantity).to.not.eql(expected\_sum\_quantity);

});

1. **Проверка итогового количества товаров после операций добавления/уменьшения количества:**

var current\_sum\_quantity = pm.collectionVariables.get("current\_sum\_quantity");

var quantity = pm.collectionVariables.get("quantity");

quantity = parseInt(quantity);

pm.test("Товар был добавлен корректно (некорректный язык и валюта)", function () {

var jsonData = pm.response.json();

var expected\_sum\_quantity = (parseInt(current\_sum\_quantity) + quantity).toString();

pm.expect(jsonData.sum\_quantity).to.eql(expected\_sum\_quantity);

});

pm.collectionVariables.set("current\_sum\_quantity", pm.response.json().sum\_quantity);

1. **Проверка отсутствия требуемого количества товаров на складе:**

var current\_sum\_quantity = pm.collectionVariables.get("current\_sum\_quantity");

var requestData = pm.request.body.urlencoded.toObject();

var quantity = parseInt(requestData.quantity);

pm.test("Указанного количества товаров нет на складе", function () {

var jsonData = pm.response.json();

var expected\_sum\_quantity = (parseInt(current\_sum\_quantity) + quantity).toString();

pm.expect(jsonData.sum\_quantity).to.eql(expected\_sum\_quantity);

});

pm.collectionVariables.set("current\_sum\_quantity", "0");

Инструкция по работе с данным подразделом:

1. Перед началом работы необходимо убедитесь, что значение 0 установлено для переменной **current\_sum\_quantity**.
2. При авторане будет падение теста в запросе №6 (превышение, количество товаров больше, чем доступно на складе).
3. После выполнения запросов необходимо очистить корзину, это можно сделать физически на сайте или используйте запрос из папки Позитивные – “Удалить товар (всю категорию) через кнопку в корзине”.

**Подраздел “Добавить товар в корзину через персональную карточку товара.”**

Подраздел содержит следующие запросы:

1. Добавить товар в корзину через персональную карточку товара по неверному id товара и parent\_product.
2. Добавить товар в корзину через персональную карточку товара без ключей this\_listId и parent\_product.
3. Добавить товар в корзину через персональную карточку товара, некорректный язык и валюта.
4. Добавить товар в корзину через персональную карточку товара, с пустым идентификатором сессии.
5. Добавить товар в корзину через персональную карточку товара, некорректный адрес запроса.
6. Добавить товар в корзину через персональную карточку товара, другой метод запроса.
7. Добавить товар в корзину через персональную карточку товара, количество товара превышает количество на складе.

Запросы содержат тестовые скрипты следующих видов (скрипты однотипные и адаптированы в мелких деталях под каждый запрос):

1. **Проверка статус-кода ответа:**

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

pm.response.to.have.status(200);

});

1. **Проверка содержание в ответе значения error для ключа status:**

pm.test("Проверяем, что status в ответе будет error", function () {

var jsonData = pm.response.json();

pm.expect(jsonData.status).to.eql("error");

});

1. **Проверка, что товар добавляется без parent\_product и this\_listId:**

var current\_sum\_quantity = pm.collectionVariables.get("current\_sum\_quantity");

var quantity = pm.collectionVariables.get("quantity");

quantity = parseInt(quantity);

pm.test("Товар добавляется", function () {

var jsonData = pm.response.json();

var expected\_sum\_quantity = (parseInt(current\_sum\_quantity) + quantity).toString();

pm.expect(jsonData.sum\_quantity).to.eql(expected\_sum\_quantity);

});

pm.collectionVariables.set("current\_sum\_quantity", pm.response.json().sum\_quantity);

1. **Проверка, что количество товаров будет отличаться в разных сессиях:**

var current\_sum\_quantity = pm.collectionVariables.get("current\_sum\_quantity");

var quantity = pm.collectionVariables.get("quantity");

quantity = parseInt(quantity);

pm.test("Товар с пустым идентификатором сессии успешно добавляется в корзину (количество товаров разное)", function () {

var jsonData = pm.response.json();

var expected\_sum\_quantity = (parseInt(current\_sum\_quantity) + quantity).toString();

pm.expect(jsonData.sum\_quantity).to.not.eql(expected\_sum\_quantity);

});

1. **Проверка итогового количества товаров после операций добавления/уменьшения количества:**

var current\_sum\_quantity = pm.collectionVariables.get("current\_sum\_quantity");

var quantity = pm.collectionVariables.get("quantity");

quantity = parseInt(quantity);

pm.test("Товар был добавлен корректно (некорректный язык и валюта)", function () {

var jsonData = pm.response.json();

var expected\_sum\_quantity = (parseInt(current\_sum\_quantity) + quantity).toString();

pm.expect(jsonData.sum\_quantity).to.eql(expected\_sum\_quantity);

});

pm.collectionVariables.set("current\_sum\_quantity", pm.response.json().sum\_quantity);

1. **Проверка отсутствия требуемого количества товаров на складе:**

var current\_sum\_quantity = pm.collectionVariables.get("current\_sum\_quantity");

var requestData = pm.request.body.urlencoded.toObject();

var quantity = parseInt(requestData.quantity);

pm.test("Указанного количества товаров нет на складе", function () {

var jsonData = pm.response.json();

var expected\_sum\_quantity = (parseInt(current\_sum\_quantity) + quantity).toString();

pm.expect(jsonData.sum\_quantity).to.eql(expected\_sum\_quantity);

});

pm.collectionVariables.set("current\_sum\_quantity", "0");

Инструкция по работе с данным подразделом:

1. Перед началом работы необходимо убедитесь, что значение 0 установлено для переменной **current\_sum\_quantity**.
2. При авторане будет падение теста в запросе №7 (превышение, количество товаров больше, чем доступно на складе).
3. После выполнения запросов необходимо очистить корзину, это можно сделать физически на сайте или используйте запрос из папки Позитивные – “Удалить товар (всю категорию) через кнопку в корзине”.

**Подраздел “Изменение количества / удаление товаров.”**

Подраздел содержит следующие запросы:

1. Удалить товар (уменьшить количество товара) через превью товара по несуществующему id.
2. Удалить товар (уменьшить количество товара) через карточку товара по несуществующему id.
3. Удалить товар (всю категорию) через кнопку в корзине по несуществующему id.
4. Изменение количество товара на странице корзины по некорректному itemID

Запросы содержат тестовые скрипты следующих видов (скрипты однотипные и адаптированы в мелких деталях под каждый запрос):

1. **Проверка статус-кода ответа:**

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

pm.response.to.have.status(200);

});

1. **Проверка содержание в ответе значения error для ключа status:**

pm.test("Проверяем, что status в ответе будет error", function () {

var jsonData = pm.response.json();

pm.expect(jsonData.status).to.eql("error");

});

1. **Проверка содержание в ответе значения ok для ключа status при удалении категории товара по несуществующему id:**

pm.test("При удалении товаров по несуществующему id status ответа ok", function () {

var jsonData = pm.response.json();

pm.expect(jsonData.status).to.eql("ok");

});

Инструкция по работе с данным подразделом:

1. Достаточно запустить авторан, все тесты будут пройдены успешно.