

**АВТОНОМНАЯ НЕКОММЕРЧЕСКАЯ ОРГАНИЗАЦИЯ
ПРОФЕССИОНАЛЬНАЯ ОБРАЗОВАТЕЛЬНАЯ ОРГАНИЗАЦИЯ
«Международный Колледж Бизнеса и Дизайна»
(АНО ПОО «Международный Колледж Бизнеса и Дизайна»)**

УТВЕРЖДАЮ
Директор АНО ПОО «МКБид»
Н.Н.Репин

«18» августа 2023 г.



СБОРНИК МЕТОДИЧЕСКИХ УКАЗАНИЙ

Для студентов по выполнению практических работ
по учебной дисциплине МДК 05.03 Тестирование информационных систем

программы подготовки специалистов среднего звена
по специальности: 09.02.07 «Информационные системы и программирование»

Тема: Знакомство с инструментом для написания тест-кейсов Qase TestOps. Создание тестовых коллекций. Создание простого тест кейса, вехи. Изучение основных свойств тест-кейса. Работа с GET запросами.

ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

Qase — это универсальная платформа TestOps, созданная для красивой организации контроля качества как для ручного, так и для автоматизированного тестирования. С помощью Qase команды QA могут управлять тестовыми случаями, управлением дефектами, планами тестирования, тестовыми запусками с информативными отчетами, а также использовать богатый API и веб-перехватчики. Qase имеет интеграцию с самыми популярными трекерами задач, включая Jira, Asana, Slack и десятки других

Общее меню навигации

- Project — содержит проекты, которые мы выбираем в списке из созданных. Можно создать свой проект. Тут же находится список кейсов, раны, планы и прочее.
- Workspace — настройка рабочего пространства (пользователи, пол, роли).
- Reports — отчеты и запросы.
- Apps — интеграции.

Проекты — это рабочие области, в которых все тестирование программного обеспечения, связанное с конкретным приложением, объединяется. В качестве примера мы создадим воображаемое мобильное приложение для заказа еды.

Создание проекта

«Создать новый проект». Далее вам будет предложено заполнить настройки проекта (которые также можно отредактировать позже):

- *Имя проекта:* описательное имя нового проекта.
- *Код проекта:* короткое имя вашего проекта, которое должно состоять из 2-6 символов, используя только латинские символы и цифры. Используйте это имя для идентификации проекта и помните, что оно будет добавлено ко всем тестовым случаям, созданным в этом проекте.
- *Описание:* используйте это поле, чтобы предоставить дополнительный контекст и дополнительные сведения всем, кто участвует в вашем новом проекте.
- *Доступ к проекту:* проекты могут быть общедоступными и частными. Публичный проект будет доступен всем пользователям учетной записи, текущим и всем, кто присоединится позже. Частный проект может следовать одному из трех правил общего доступа:

- *«Добавить всех участников в этот проект»:* каждый приглашенный вами пользователь текущего аккаунта получит доступ к проекту. Каждому новому пользователю, который присоединится к учетной записи позже, необходимо будет предоставить доступ к этому проекту вручную.
- *«Добавить участников из определенной группы»:* выбрав эту опцию, вам нужно будет выбрать группу пользователей, которую вы создали ранее - каждый пользователь, принадлежащий к выбранной группе пользователей, получит доступ к вашему новому проекту.
- *«Не добавлять участников»:* с помощью этой опции вы можете пока сохранить конфиденциальность своего проекта и поделиться им с другими, когда будете готовы.

Последним шагом является подтверждение создания нового проекта, нажав кнопку «Создать проект» ниже - после создания вы готовы создать свой проект с помощью тестовых примеров.

Владельцы рабочей области могут просматривать все проекты, включая частные проекты, созданные другими пользователями рабочей области.

Наборы/коллекции тестов в тестировании программного обеспечения

Набор тестов действует как папка, в которой мы храним наши тестовые примеры, организованные по разделам и подразделам. Можно создать несколько наборов тестов. Например, функциональные и нефункциональные наборы тестов или что-то еще, что следует рассматривать как отдельную группу тестов.

Набор/коллекция +Suite для тест-кейсов содержат следующие поля:

- **Имя набора:** дайте вашему набору тестов описательное имя, которое позволит вам однозначно определить, какую область или область разработки он будет охватывать. Название набора является единственным обязательным полем для создания набора - другие поля являются необязательными и могут быть заполнены на более позднем этапе или оставлены пустыми.

- Родительский пакет: можно иметь многоуровневую иерархию наборов тестов, поэтому, если набор, который вы настраиваете, будет частью более широкого набора сценариев тестирования, вы можете поместить его в родительский набор более высокого уровня.
- Описание: предоставьте дополнительные сведения и контекст относительно того, какие тестовые случаи должны принадлежать набору тестов.
- Предварительные условия: опишите шаги, которые необходимо выполнить, прежде чем вы сможете приступить к тестовым случаям из только что созданного пакета.

Case тест-кейс в тестировании программного обеспечения

Тестовый пример Case содержит все подробности о нашем тесте. Заголовка достаточно, чтобы создать быстрый тестовый пример. Для более подробного тестового примера можно определить предварительные требования, шаги, ожидаемые и фактические результаты и т. д.

Быстрый тестовый пример

1. На странице под вашим набором тестов нажмите на ссылку «+ Создать быстрый тест»
2. Курсор начнет мигать на странице, и вы сможете начать вводить заголовок. Нажмите клавишу ВВОД, чтобы сохранить тестовый случай. В результате новый тестовый пример в наборе тестов отображается с идентификатором и заголовком.

Меню свойства тест-кейса:

1. **Заголовок:** определение имени тестового случая

2. **Описание:** дополнительные сведения для получения дополнительной информации о тестовом случае

STATUS *

Type to search

- Actual
- Draft
- Deprecated

3. **Состояние:** может быть
 - активным,
 - черновиком
 - устаревшим

Suite

Type to search

Test cases without suite

проверка форм

4. **Suite:** выберите, к какому набору тестов относится ваш новый кейс, те наборы, которые мы создадим

Severity *

Type to search

- Not set
- Blocker
- Critical
- Major
- Normal
- Minor
- Trivial

5. **Серьезность:** может быть
 - не установленной
 - блокирующей
 - критической
 - серьезная
 - нормальная
 - незначительная
 - тривиальная

Priority *

Type to search

- Not set
- High
- Medium
- Low

6. **Приоритет:** может быть
 - низким,
 - средним,
 - высоким
 - не установлен

Type *

Type to search

- Functional
- Smoke
- Regression
- Security
- Usability
- Performance
- Acceptance
- Compatibility
- Integration
- Exploratory

7. **Тип:** выберите, какой тип тестирования применим к вашему тестовому случаю:
 - функциональное тестирование
 - дымовое тестирование
 - регрессионное тестирование
 - тестирование безопасности
 - юзабилити тестирование
 - производительности
 - приемочное тестирование
 - тестирование совместимости
 - тестирование интеграции
 - исследовательское тестирование

Layer *

Type to search

- Not set
- E2E
- API
- Unit

8. **Уровень:** выберите слой тестового примера,
 - сквозное тестирование (End-to-end)
 - тестирование интерфейсов (API)
 - модульное тестирование (Unit)

Milestone

- Not set
- Not set
- БАЗОВЫЙ ЭТАП
- НАЧАЛЬНЫЙ ЭТАП

9. **Веха:** выберите, связан ли тестовый случай с одной из ваших вех, которую вы можете создать отдельно
 Это какие-то критические точки для проверки проекта, могут быть версии

Behavior *

Type to search

- Not set
- Positive
- Negative
- Destructive

10. **Поведение:** может быть как
 - деструктивным,

- негативным,
- позитивным,
- неустановленным

- будет автоматизирован
- автоматизированный

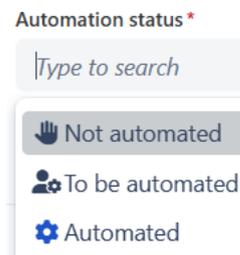
11. Шелушится: если тестовый случай нестабилен, вы можете пометить его как шелушащийся
 -шелушающийся
 -Не шелушащейся

11. Статус автоматизации:

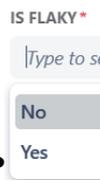
вы можете выбрать
 -не автоматизированный,

Milestone (веха) — в управлении проектами контрольная точка, значимый, ключевой момент, (например, переход на новую стадию, новый этап в ходе выполнения проекта. Используется для обозначения вехи в разработке.

майлстоун при создании рана или кейса в рамках проекта. Если у вас несколько каждой свой проект, создать единую точку входа не получится.
 – у вас выведется окно с горячими клавишами



Можно выбрать команд, и у При нажатии на Р



Горячие клавиши

Страница репозитория

- F** Открыть список фильтров
- C** Перейти на страницу нового тестового случая
- S** Перейти на страницу нового набора тестов

Страница создания/редактирования сущности

- Cmd/Ctrl + S** Сохранить сущность
- Cmd/Ctrl + Shift + S** Сохраните сущность и перейдите на страницу новой сущности

Мастер тестового запуска

- F** Пометка тестового случая как неудачного
- P** Пометить тестовый случай как пройденный
- B** Пометка тестового случая как заблокированного
- S** Пометить тестовый случай как пропущенный
- Я** Пометка тестового случая как недопустимого
- V** Перейти на страницу просмотра тестового примера (откроется в новом окне)
- E** Перейти к тестовому кейсу редактирования

Создание/редактирование страницы тестового случая

- Cmd/Ctrl + .** Добавление нового шага
- Cmd/Ctrl + Shift + .** Добавление нового дочернего шага
- ALT+S** Быстрое сохранение
- Cmd/Ctrl + R** Отправить тестовый случай на проверку

Список сущностей

- F** Открыть список фильтров
- C** Переход на страницу новой сущности

Репозиторий кейсов. Что мы здесь видим:

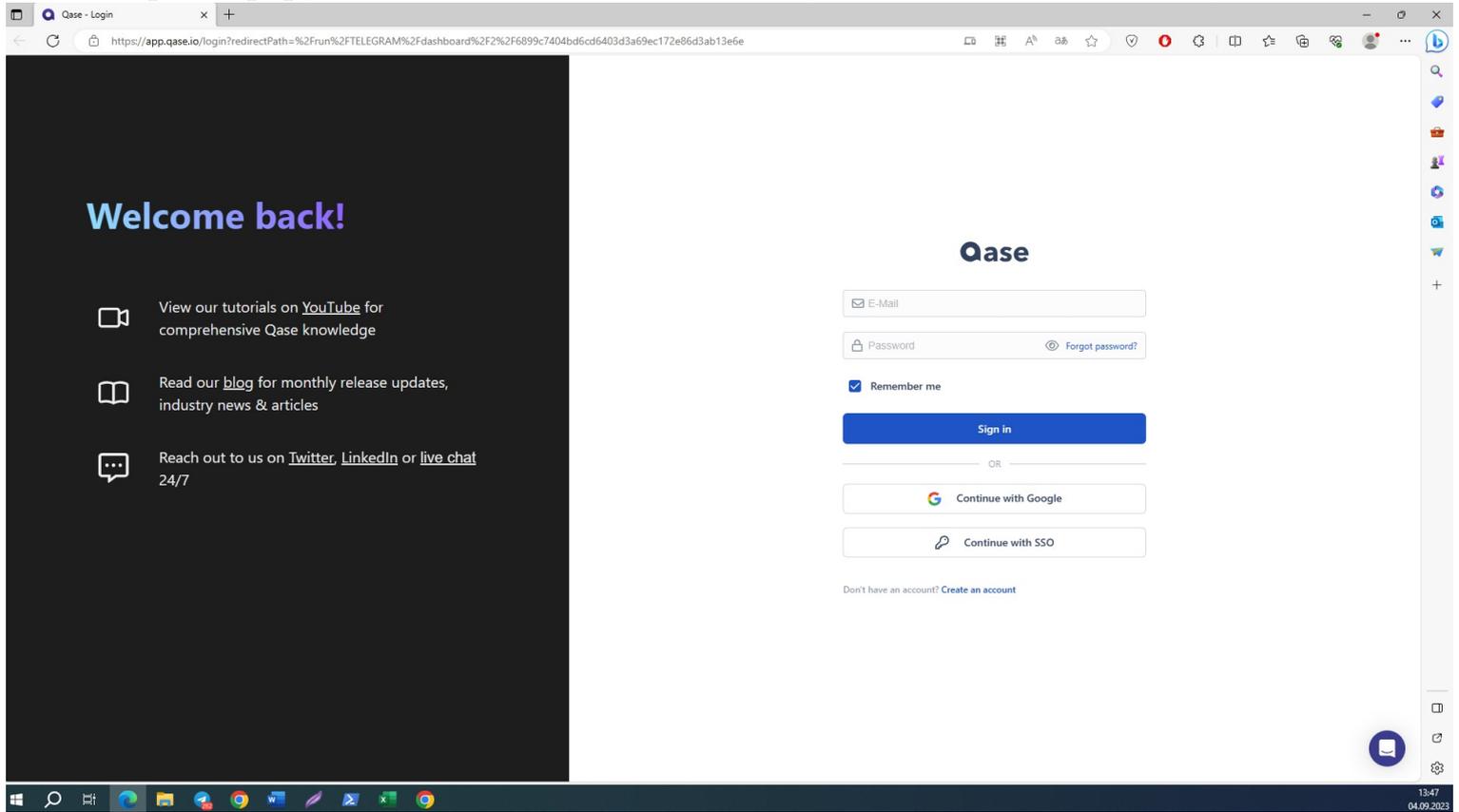
- Меню действий с кейсами — появляется, только если выбрать несколько кейсов. Действия видны на кнопках.
- Список кейсов — содержит иконку автоматизации (рука с шестеренкой, которой настраивается уровень автоматизации), ID кейса и название. Внизу есть кнопка быстрого создания кейса. Около названия сьюта 4 кнопки — создание сьюта или кейса, редактирование описания, клонирование и удаление.
- View — способ отображения кейсов. Все в одну длинную портянку или каждая папка отдельно.
- Меню действий — экспорт, импорт кейсов. А также крайне удобная корзина. Никакой кейс никогда не удалится, а просто будет в корзине.
- Список сьютов в проекте — содержит название и количество кейсов в нем.
- Счетчик кейсов — содержит общее число сьютов и кейсов. Если применить фильтр, будет содержать значения согласно выставленным фильтров.
- Добавление фильтров.

Кратко о плюсе TestRail по этому разделу — там можно добавить отображение информации о дополнительных полях в список. В Qase такого, к сожалению, пока еще нет.

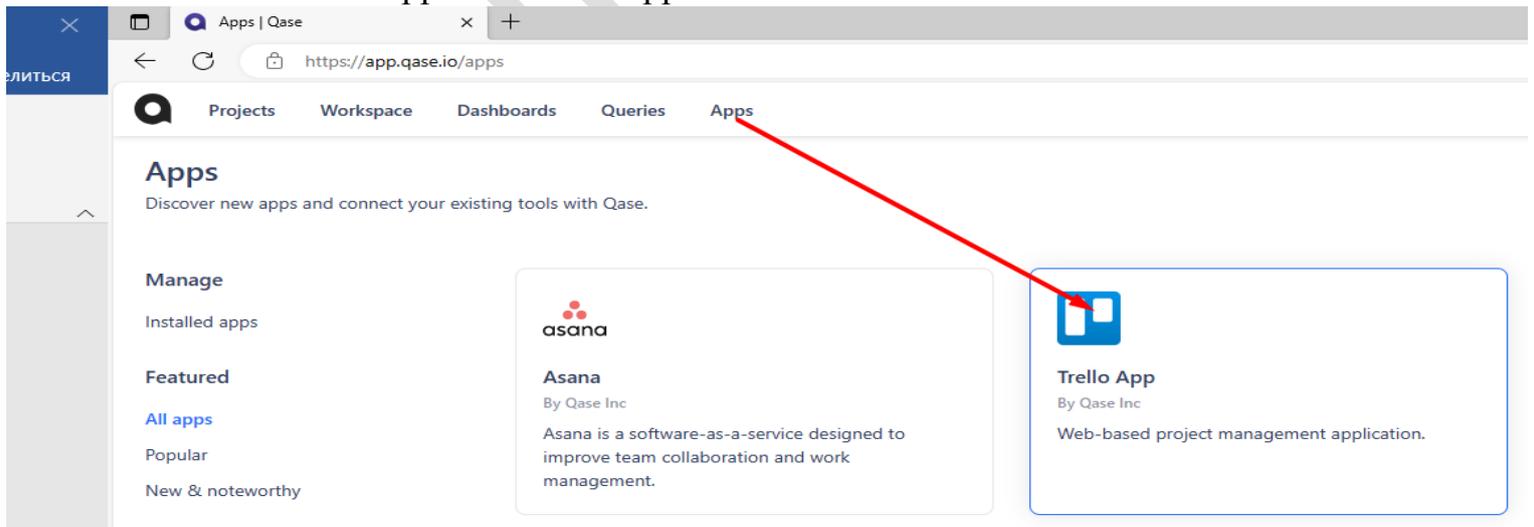
ХОД РАБОТЫ

Задание 1 зарегистрируйтесь и ознакомьтесь с панелью управления. Подключите доску Trello. Создайте две вехи.

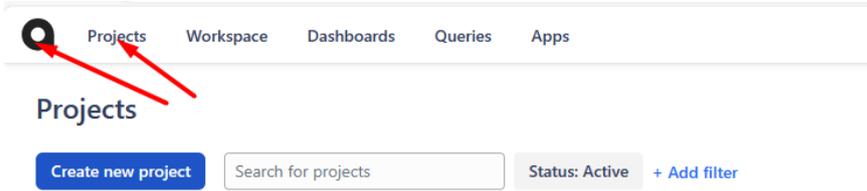
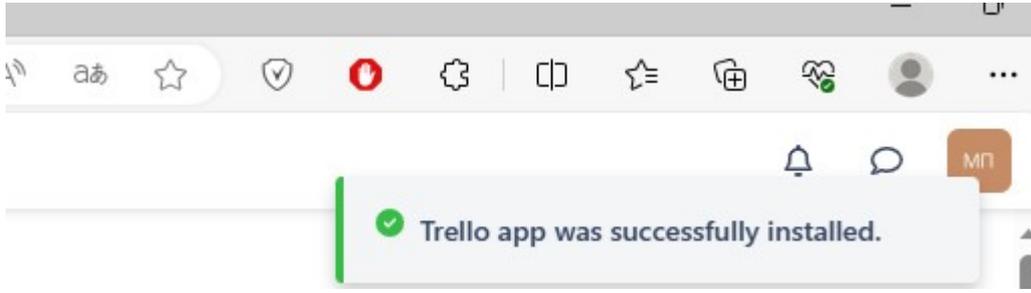
1. Перейдите по ссылке [Qase - Вход](#)
2. Зарегистрируйтесь или войдите с помощью гугл аккаунта



3. Нажмите меню Apps => Trello App

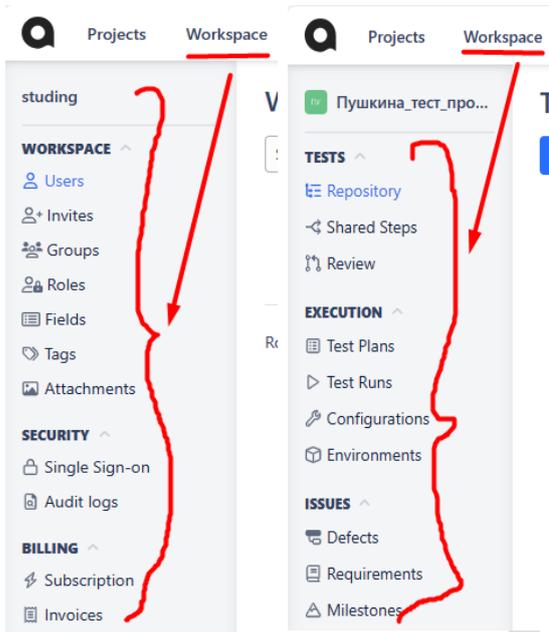


4. Свяжите свои аккаунты



5. Нажмите Projects и создайте новый проект под своей фамилией

5.1 дайте название проекту, название для разработки, описание и нажмите создать



ваша основная рабочая область -это **Workspace** и ее

боковое меню

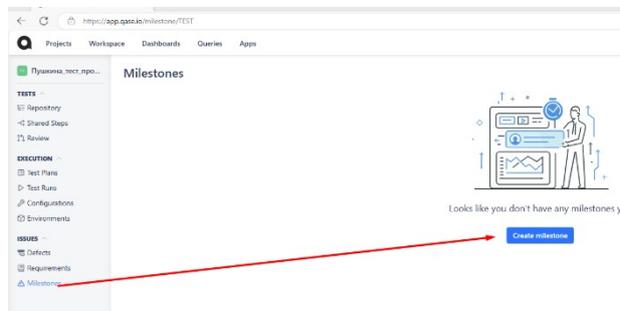
5.3 Изучите каждый пункт бокового меню, за что он отвечает и что он делает

5.4 На рисунках показаны две области рабочие с выбранным проектом и без выбранного проекта

6. Создайте две вехи:

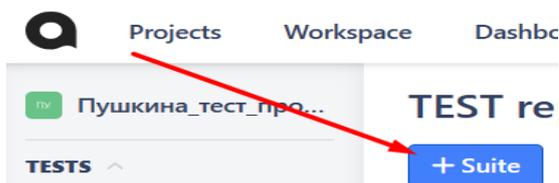
6.1 начальный этап

6.2 базовый этап



первый, ему группу, к

Задание 2 Создайте свой быстрый тест кейс и укажите которой он будет относиться



7. нажмите Suite (для создания вашей первой коллекции) и укажите ему название, к которой данные тест кейсы будут относиться

Теперь заполним поля:

- Имя набора: дайте вашему набору тестов описательное имя
- Родительский пакет: можно иметь многоуровневую иерархию наборов тестов

- Описание: предоставьте дополнительные сведения и контекст относительно того, какие тестовые случаи должны принадлежать набору тестов.
- Предварительные условия: опишите шаги, которые необходимо выполнить, прежде чем вы сможете приступить к тестовым случаям из только что созданного пакета.

7.1 Проверка форм, нажмите создать.

В выпадающем списке вы

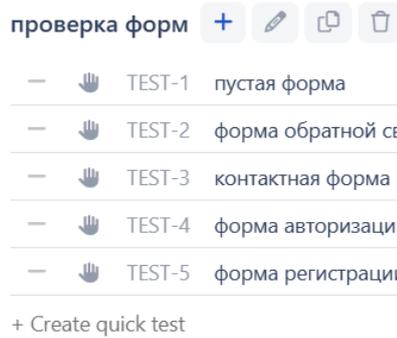
сможете увидеть все коллекции\ группы ваших тестов, по какому-либо признаку. К

примеру: *проверка форм*, может содержать: форму регистрации, форму авторизации, форму обратной связи и т.д.

После этого в репозитории у вас появится ваш первый базовый набор тест-кейсов, который вы сможете в дальнейшем редактировать и задавать ему иерархию, либо клонировать в другие проекты, для этого нажать ... и в списке выбрать **любое из действий**

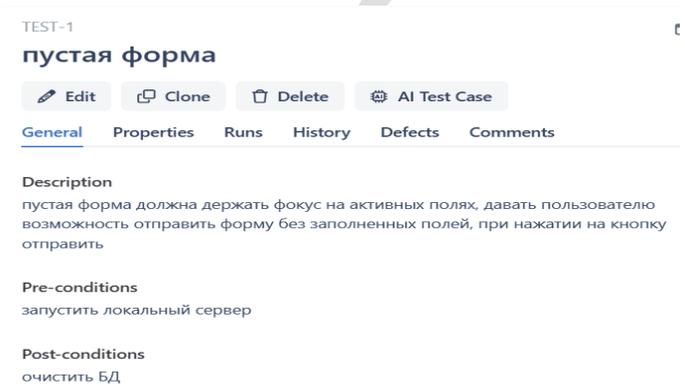
8. Для создания быстрого тест-кейса вам нужно изменить структуру отображения на **FolderView**

8.1 после чего в поле введите название вашего теста



8.2 создайте 5 различных форм

8.3 Нажмите на **Путая форма** и в правом углу вы увидите блок для написания тела тест-кейса, оно состоит из **Описание, Предусловие**



и **Постусловие**. Заполните их.

8.4 После чего изучите вкладки данного окна: свойства, запуск, история, дефекты, комментарии.

Обратите внимание на вкладку Свойства [Properties](#)

Изучите и запомните все свойства тест-кейса

8.5 нажмите удалить Контактную форму

контрольные вопросы:

1. что такое тест-кейс?
2. Что такое коллекция тестов?
3. Как создать проект?
4. Как создать быстрый тест-кейс?
5. Какие бывают статусы?
6. Чем отличается пред условие от пост условия?
7. Перечислите основные свойства тест-кейса
8. Назовите уровни тестирования
9. Что значит шелушение? Какие возможны состояния шелушения?
10. Какие поведение тестирования можно установить?
11. Какие приоритеты знаете?
12. Зачем ставят метки?
13. Какие состояния серьезности знаете?
14. Назовите типы тестирования

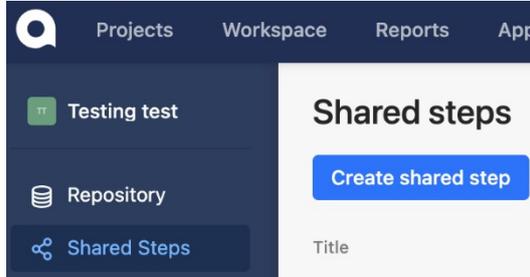
Практическая 2-3

Тема: создание полных тестовых случаев на основе сайта Ozon. Клонирование коллекций. Клонирование тест-кейсов с выборкой. Редактирование тест кейсов. Использование и редактирование свойство тест-кейсов. Работа с шагом тест-кейса. Общий шаг и его редактирование для всех тест-кейсов.

ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

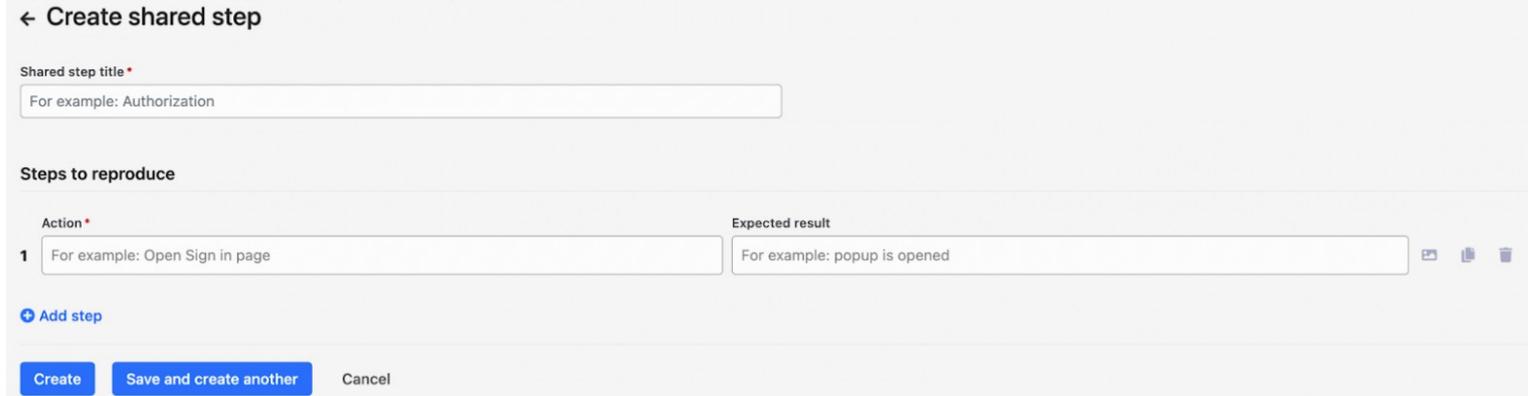
Shared Steps

Здесь содержится список Shared Steps для проекта. Доступно создание шареда и его редактирование, а также поиск по имеющимся. Удобно тем, что они хранятся не в общей куче, для каждого проекта свой набор. Их без проблем можно редактировать и тут же создавать. Как можно заметить — отображается количество кейсов, в которых он используется.



Shared Steps *Минус* — можно создать шареды с одинаковым названием. И только тут можно сделать его из двух и более шагов.

При создании шареда в режиме редактирования кейса создавать можно лишь из одного шага:

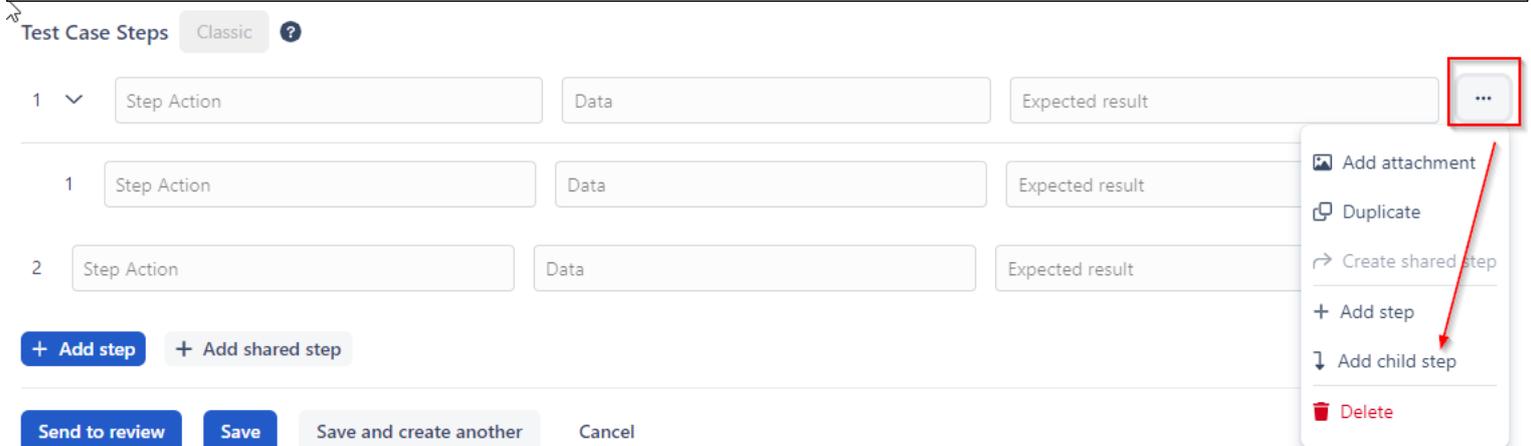


Что такое вложенные шаги в тестировании программного обеспечения?

Это список подшагов, которые необходимо выполнить в течение одного шага. Этот шаг имеет отношения «родитель-потомок» и является подшагом другого шага.

Использование вложенных шагов

Чтобы использовать вложенный шаг, нажмите на трехточечное меню шага и выберите «Добавить дочерний шаг».



Для создания общего шага

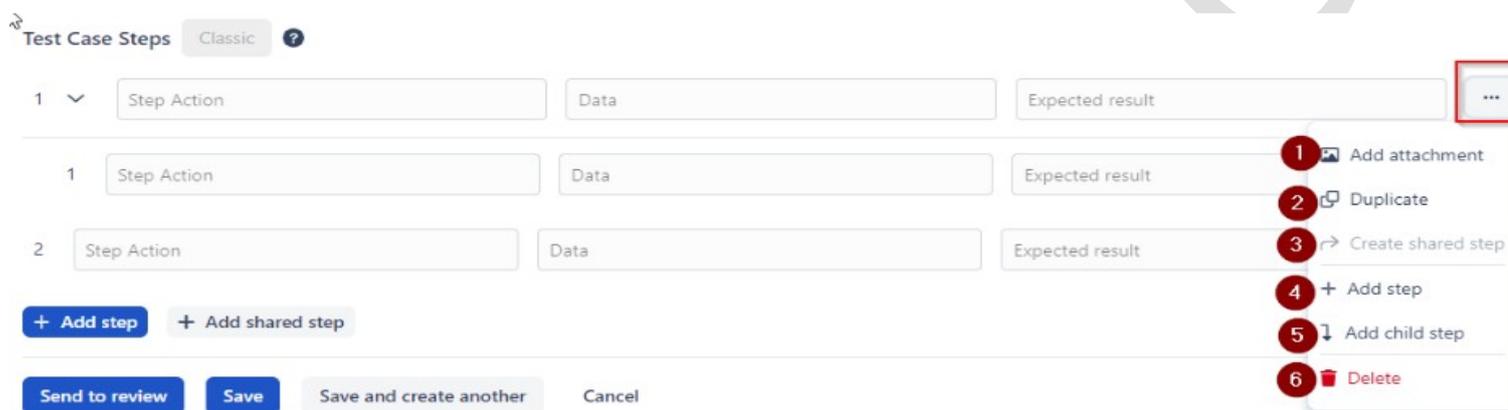
Общие шаги — это уникальная функция Qase, которая позволяет создать один шаг, который можно использовать совместно для нескольких тестовых случаев. Это устраняет необходимость повторного создания одного и того же шага вручную для каждого тестового случая и обеспечивает согласованность во всех тестовых случаях, в которых используется общий шаг.

Если у вас есть шаги, которые применяются к нескольким тестовым случаям в проекте и имеют совпадающие свойства, вы можете сэкономить время, создав общие шаги. После того, как у вас есть хотя бы один общий шаг, созданный в представлении «Общие шаги» вашего проекта, вы можете выбрать «+ Добавить общий шаг» при создании или редактировании тестового случая.

После чего вы можете настроить шаг кейса

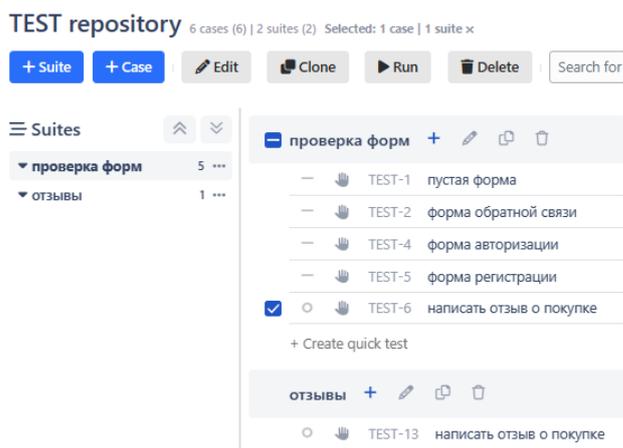
При настройке шага кейса вы можете настроить его с помощью кнопок рядом с полем "Ожидаемый результат":

1. Вы можете прикрепить файлы из существующих файлов в Qase или с вашего компьютера.
2. Вы можете дублировать/клонировать шаг.
3. Вы можете создать общий шаг.
4. Вы можете добавить шаг.
5. Вы можете добавить вложенный шаг.
6. Вы можете удалить шаг.



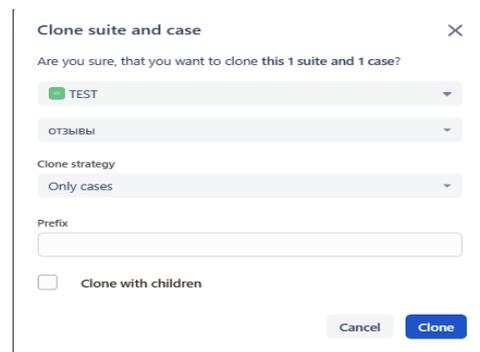
ХОД РАБОТЫ

1. Откройте ресурс Перейдите по ссылке [Qase - Вход](#)
2. Авторизуйтесь



Задание 1 Создайте одну коллекцию назовите Отзывы, клонируйте туда любой тест кейс

Для этого создайте коллекцию, после чего быстрый тест, поставьте галочку возле данного тест-кейса и нажмите



клонировать. В диалоговом окне укажите : родителя Отзывы и укажите только тест-кейсы, нажмите клонировать

Задание 2 Создать коллекцию «написать отзыв о покупке». Тест-кейс-1 «Заполнить все поля»

3. Для создания тест-кейса вам нужно нажать либо на **+Case** либо на созданную вашу коллекцию ... и в списке выбрать **+Create Case**

После нажатия создать кейс у вас появится базовое окно для написания тест кейсов

4. Перейдите по ссылке, авторизируйте [OZON — интернет-магазин. Миллионы товаров по выгодным ценам](#)

Ваша задача установить все те же параметры, как показаны на скрине ниже. в пред условия вы опишите все шаги, которые вы сделали, чтобы достичь данного этапа «написать отзыв о покупке»

5. Введите все параметры как на скрине ниже, мы вводим все данные которые нам предоставят для тестирования

5.1 первая колонка что тестируем(название)

5.2 какие данные вводим

5.3 третья ,какой результат должен быть

← Create test case

Basic

Title* Status*

Description

Suite Severity* Priority*

Type* Layer* Is flaky*

Milestone Behavior* Automation status*

Conditions

Pre-conditions

Post-conditions

Test Case Steps Classic

1	оценка звезд	выставить 5 звезд	5 звезд выделены желтым, текст "Отлично"
2	поле достоинства	хорошо защищает от бликов и подтеkania, плотно прилегают	
3	поле недостатки	не было в комплекте переносиц	
4	поле отзыв	советую данные очки для слепых, отлично видно все под водой=)	советую данные очки для слепых, отлично видно все под водой=)
5	добавить картинку очков	загрузить картинку в PNG формате	Загружены фотографии 1/10 x Отлично
6	добавление видео	загрузка в формате .MOV	Загружены видео 1/3
7	опубликовать отзыв	нажатие на кнопку "оставить отзыв"	опубликованный комментарий

+ Add step + Add shared step

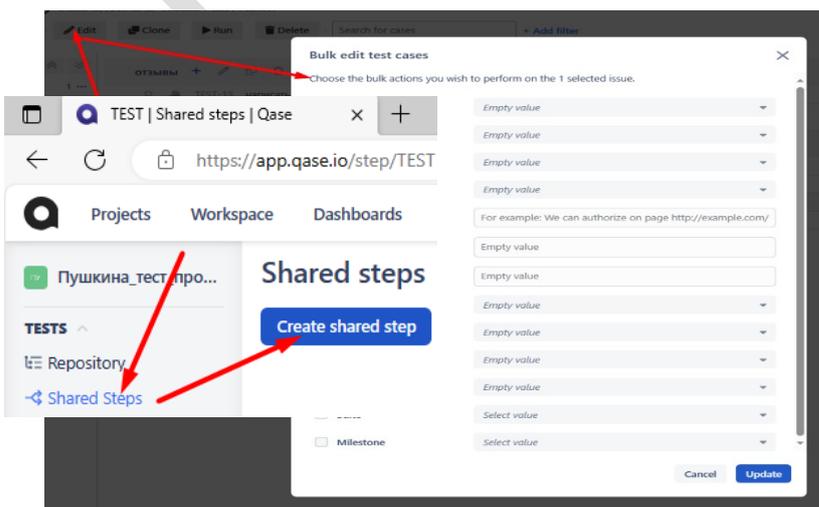
Save Save and create another Cancel

Для того, чтобы быстрому тест-кейсу или дублированному тест-кейсу задать все свойства:

1. вариант: поставьте галочку возле данного тест-кейса после чего, нажмите Редактировать и в диалоговом окне установите все нужные свойства
2. Вариант: нажмите на тест-кейс и в боковом меню слева измените свойства
3. Вариант нажмите на сам Id-тест кейса и в диалоговом окне редактируйте его.

Задание 3 Создайте вложенность шага в тест кейсе и настройте общий шаг несколькими тест-кейсам

1.откройте любой тест кейс на редактирование создайте шаг, после чего в шаге нажмите ...



Сохраните тест кейс

2. Создайте общий шаг несколькими тест-кейсам

2.1 нажмите общий шаг, после чего добавьте новый шаг

2.2 введите общее название шагу «Сохранить»

2.3 введите название, действие и результат

Shared step title *

Сохранить

Steps to reproduce

1 сохранить форму нажать кнопку сохранить форма сохранилась

+ Add step

Create Save and create another Cancel

2.4 откройте нужный вам тест-кейс, перейдите к списку ваших тест-кейсов, после чего выберите нужный вам тест-кейс, где будет действие сохранить, нажмите

+ Add shared step

Add shared step

Choose shared step *

Type to search

Сохранить

2.5 в диалоговом окне выберите нужный вам шаг Сохранить, нажмите добавить

2.6 результат

Test Case Steps Classic

1 поле имя марина марина

2 сохранить форму нажать кнопку сохранить форма сохранилась

+ Add step + Add shared step

Save Cancel

Для просмотра нажмите тест-кейс и справа в меню будут шаги которые вы добавили

отзывы + формы +

- TEST-13 написать отзыв о покупке
- TEST-18 загрузка видео
- TEST-19 заполнение только спец символами
- TEST-20 заполнение минимального количества полей
- TEST-21 оценка товара
- TEST-22 проверка слайдера товара
- TEST-23 загрузка картинок

контактная форма

форма обратной связи

контактная форма

Edit Clone Delete AI Test Case

General Properties Runs History Defects Comments

Description
Not set

Pre-conditions
Not set

Post-conditions
Not set

Steps Shown as table

Action	Data	Expected result	Attachments
1 поле имя	марина	марина	
2 сохранить форму	нажать кнопку сохранить	форма сохранилась	

Задание 4 создать из шага, общий шаг для нескольких тест кейсов с возможностью дальнейшего редактирования данного поля

1. При редактировании любого из тест-кейсов, нажмите ... выберите **Create shared step** задайте название общему шагу Ввод текста и нажмите создать.

Create shared step

Step title *

ВВОД ТЕКСТА

Cancel Create

1.1 В Shared Steps

отобразится ваш общий шаг

ТЕСТЫ ^

Хранилище

Общие шаги

Обзор

Создание общ

Титул

ВВОД ТЕКСТА

1.2 чтобы использовать общий шаг, но при этом иметь возможность его редактировать для этого нужно открыть любой тест-кейс, где применялся общий шаг в данном шаге нажмите ...

и выберите  Convert to regular

[+ Add attachment](#)

Parameters

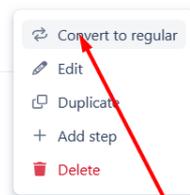
[+ Add parameter](#)

Test Case Steps Classic ▾

1 поле имя

марина

марина



Задание 3.1 Создать тест-кейс:

- В слайдер "Ожидания оценки": убедится, что выводится правильное кол-во купленных товаров, их картинки. Слайдер должен пролистываться туда назад, при нажатии выбирается товар, по которому будет совершена оценка, ниже выводится текст с названием товара.

Контрольные вопросы:

1. Есть ли ограничения по шагам в создании тест-кейса?
2. Как клонировать коллекцию со всеми тест-кейсами?
3. Какие три колонки есть при создании тест-кейса?
4. Как редактировать тест-кейс?
5. Как быстрому тесту задать все свойства?
6. Назовите все способы как задать тест-кейсу определенные свойства
7. Как создать общий шаг? Что это такое?
8. Как создать вложенность шага?
9. Как использовать общий шаг ,но при этом иметь возможность его изменить?

Практическая 4-5

Тема: создание тестовых случаев на основе сайта Ozon. Раздел отзыва о товаре. Создание тестовых параметров. Написание тест кейсов: для оценки товара, загрузки картинок, загрузки видео.

ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

Что такое параметры при тестировании ПО?

Параметры теста — это заполнители для фактических значений. Они позволяют вам писать тестовые примеры и шаги таким образом, чтобы тестировщики могли выполнять их с определенными вариациями различных значений. Эти параметры поддерживают создание нескольких вариантов *этапов тестирования* с несколькими разными значениями. Использование параметров теста может повысить гибкость ваших тестов, позволяя повторно выполнять один и тот же шаг с разными значениями данных.

Тестировщики могут записывать переменные в процессе тестирования в качестве параметров. Это позволяет им использовать один и тот же тестовый случай для нескольких выполнений вместо того, чтобы создавать отдельные тестовые случаи или несколько наборов шагов.

Как работают параметры?

После того, как у вас есть параметры в случае, когда такой случай добавляется в тестовый запуск, он будет разбит на несколько идентичных копий одного и того же тестового примера - по одной на каждое указанное значение параметра

Пример

Рассмотрим, например, оценку процесса входа в веб-приложение. Вы можете предпринять следующие действия:

1. Запустите браузер вашего компьютера.
2. Введите URL-адрес веб-сайта
3. Введите соответствующие поля для имени пользователя и пароля.
4. Нажмите кнопку «Войти».

Учтите, что вы хотите провести этот тест с различными браузерами. Параметры теста, помеченные браузером, могут быть созданы и включены в тестовый пример при настройке тестового примера, в отличие от разработки отдельных тестовых случаев с аналогичными данными.

Так вот браузеры и будут вашими параметрами

ХОД РАБОТЫ

1. Откройте ресурс
2. Выберите проект
3. Перейдите к тест кейсу

Задание 1 добавим в тест кейс Контактная форма, параметры: Браузер и OS

1. Откройте тест кейс на редактирование

Parameters

2. Нажмите, добавить параметр

+ Add parameter

- 2.1 введите в первое поле Браузер, во второе поле различные браузеры 3 шт
- 2.2 введите в первое поле OS во второе поля две OS различные

Parameters

Parameter title: OS

Parameter values: windows, ubuntu

+ Add value

Parameter title: Браузер

Parameter values: хром, яндекс, опера

+ Add value

+ Add parameter

Строить тест кейсы с использованием вложенности шагов и параметров!

Задание 2 Создать тест-кейс:

Оценка от Ужасно до Отлично, при переключении звездочек. Есть возможность выбрать не оценивать эту покупку, тогда она будет убрана из слайдера.

Задание 3 Создать тест-кейс:

Загрузка картинок - работает только с определенными форматами: jpg, png с размером ДО 5-10-100мб(любой размер), если больше выдавать ошибку с « несоответствием размера или формата картинки». Загрузить битую картинку. Загрузить картинку с одинаковым именем. Удалить картинку

Задание 4 Создать тест-кейс:

Загрузка Видео, выдавать ошибку, если видео менее 3 сек, и длительность больше 5 минут. Загрузить битое видео, посмотреть как видео воспроизводится, удалить видео.

Контрольные вопросы:

1. Как просмотреть отображение созданных шагов?
2. Как изменить нескольким тест-кейсам одновременно несколько свойств?
3. Как восстановить из корзины удаленный тест кейс?
4. Как глянуть сколько у нас тестов содержит определённый статус?
5. Как вывести все тест-кейсы содержащие определенные теги?

Практическая 6-7

Тема: создание тестовых случаев на основе сайта Ozon. Раздел отзыва о товаре. Написание сквозных, функциональных тест кейсов: для форм со спец символами и минимальным количеством заполненных полей. Тестовые запуски. Статистика тестовых запусков.

ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

1. Проекты - проект представляет собой коллекцию всех артефактов тестирования
2. Наборы тестов - набор тестов представляет собой коллекцию всех тестовых случаев
3. Тест-кейсы - для хранения деталей нашего теста
4. Тестовые запуски — для определения того, какие тесты должны выполняться при сборке продукта, и отслеживания состояния выполнения.
5. Тестирование дефектов - на основе неудачного результата тест-кейса
6. Настройки учетной записи - для настройки рабочего пространства

Тестовые раны

Это набор кейсов для их прогона в рамках релиза или фичи. Особенность — есть визард, своеобразный интерфейс для прохождения рана.

Несколько входных точек для создания рана:

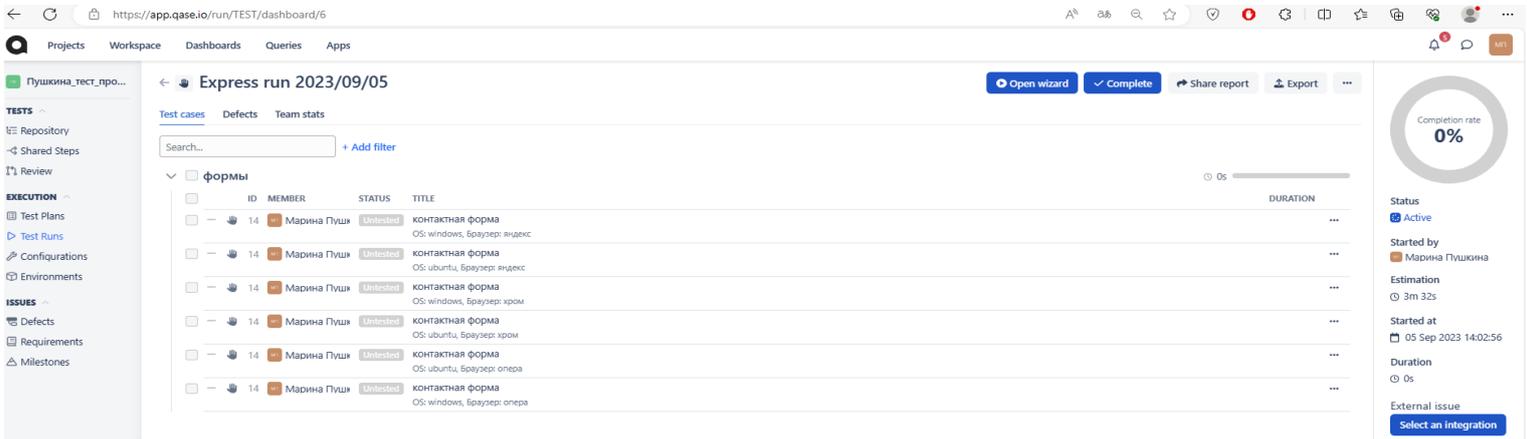
1. Выбрать план — на его основе создать ран.
2. Зайти в раздел Test Runs — создать там.
3. Выделить один или более кейсов в репозитории — стартовать при помощи панели действий (кнопка Run).

The screenshot shows a testing tool interface. On the left, there's a sidebar with navigation options like 'TESTS', 'EXECUTION', and 'ISSUES'. The main area displays a table of test cases under the heading 'Express run 2023/09/05'. The table has columns for ID, MEMBER, STATUS, and TITLE. Several test cases are listed, all with a status of 'Unlisted' and a title of 'контактная форма'. On the right, there's a detailed view of a specific test case titled 'контактная форма TEST-14'. This view includes a 'View' button, 'Execution' status (Passed, Failed, Blocked, Invalid, Skipped), 'Parameters' (OS: windows, Браузер: яндекс), and a 'Run History' section with a 'View' button. Below the parameters, there are two steps: '1 поле имя' and '2 сохранить форму'.

Тестовые раны

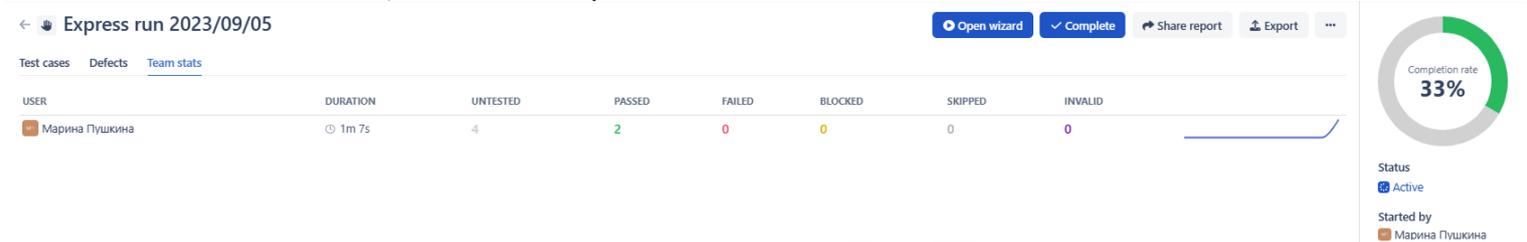
- Слева — список кейсов. Можно фильтровать их так, как нужно.
- Справа — сам кейс. Вверху результат для кейса, под каждым шагом — свой результат.
- Кнопки View и Edit открывают кейс в новом окне для просмотра и редактирования соответственно. Изменения в кейсе ведут к изменению кейса в ране после обновления страницы.
- Кастомные поля отображаются в ране над шагами (видим на примере поля AndroidResult).
- Case Run History — история прохождения кейса. Отображает результат прогона в прошлых ранах.

*Из минусов — не видно, что вписали в коммент при прохождении кейса. Нужно открыть общий список кейсов в ране (*закрыв визард) и нажать на лейбл результата. А если не проставить результат каждому шагу кейса — не увидишь шагов в пройденном кейсе.*



Видим список кейсов, при выделении доступно еще несколько действий. В боковом меню — диаграмма статусов кейсов (при нажатии выполняется фильтрация по выбранному статусу) и немного инфы (кто стартовал и во сколько).

Вкладка Team stats покажет, кто сколько прошел:



ХОД РАБОТЫ

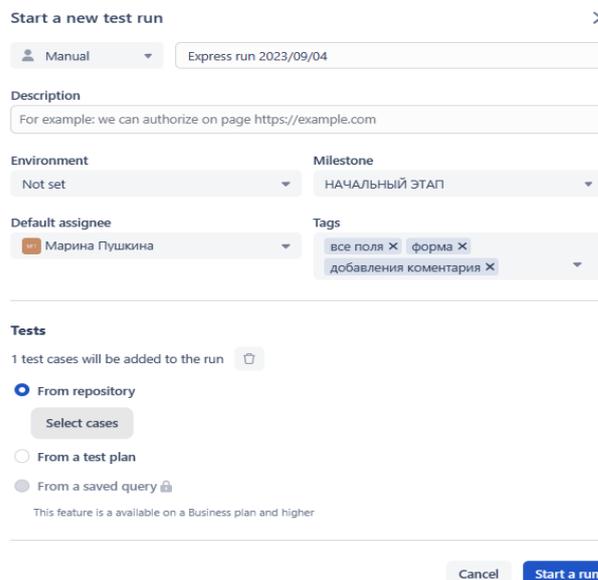
1. Откройте ресурс
2. Выберите проект
3. Перейдите к тест кейсу написать отзыв о покупке

Задание 1 запустить и проверить по тест-кейсу функционал

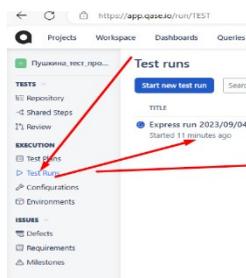
- 1.1 выберите тест кейс Написать отзыв о товаре
- 1.2 поставьте галочку
- 1.3 нажмите запустить
- 1.4 в диалоговом окне укажите себя ,веху Начальный этап и несколько тегов добавьте и нажмите запустить

Изменять, удалять вы можете в разделе тестов

1.5 для проверки тест-кейса вам нужно нажать на сам тест кейс, после чего по результату выполнения ставить состояния шага



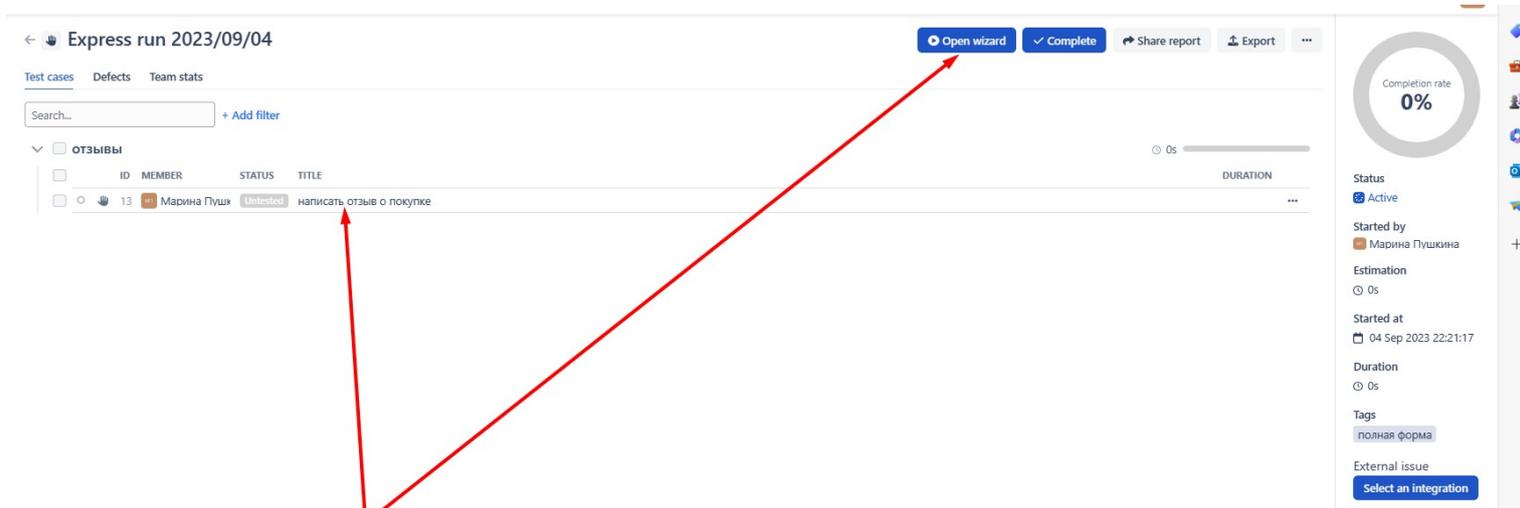
редактировать и запущенных



метки

0





написать отзыв о покупке TEST-13

View Edit case

Execution Run History

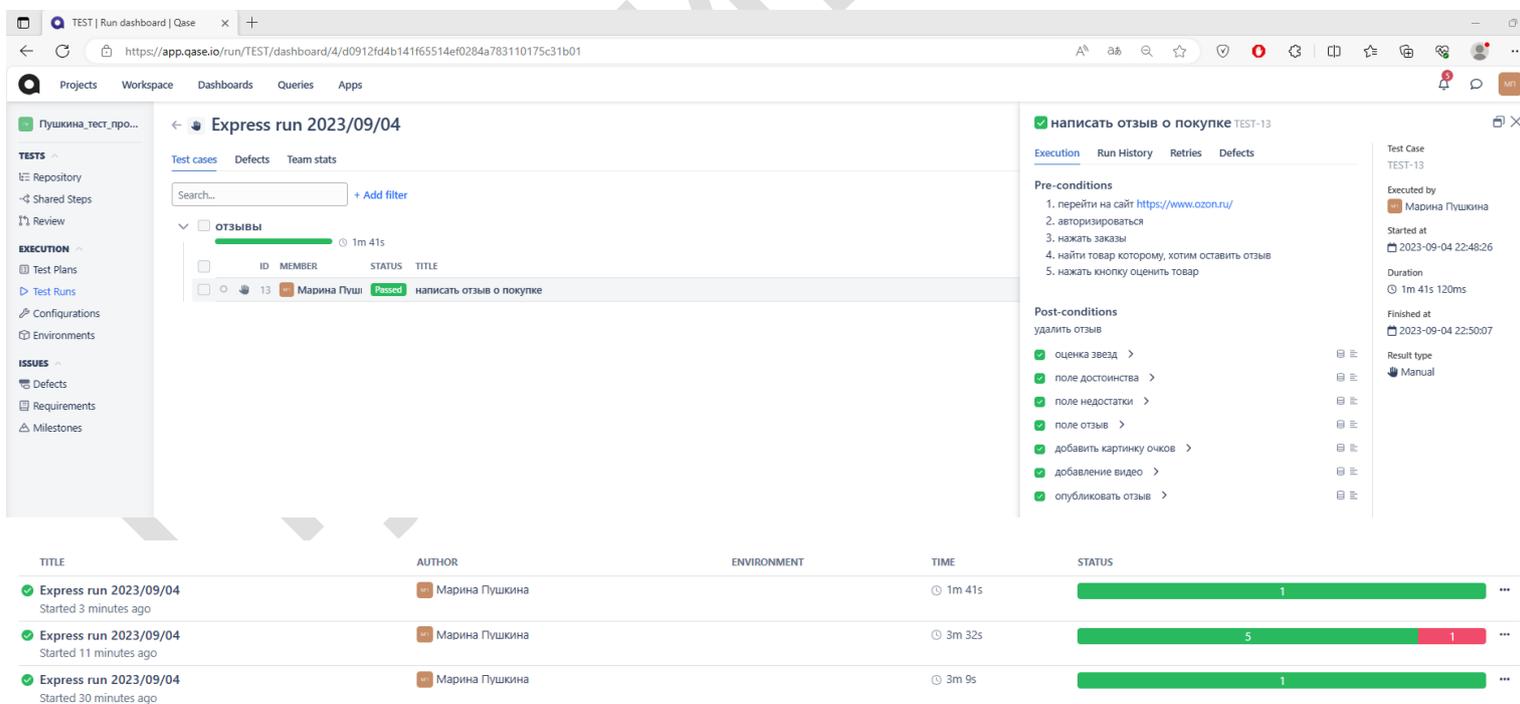
Passed Failed Blocked Invalid Skipped

1.6 По завершению всех шагов установите пройденному тест-кейсу одно из статусов (хорошо, провалено, заблокировано, неверный тест кейс, или пропустить, в том случае, если данный тест кейс в этом комплекте не используется) и нажмите **Complete**

1.7 результат можете посмотреть по выполненным шагам в разделе TestRun выбрать текущий запущенный тест и перейти к просмотру его выполнения или Общая информация видна

справа после завершения тест-кейса или на вкладке общая статистика по всем пользователям.

Test cases Defects Team stats



Задание 4 Создать, запустить и пройти тест-кейс:

- заполнить всю форму только с спец. Символами

Задание 5 Создать, запустить и пройти тест-кейс:

-заполнить форму с минимальным количеством полей

Задание 6 Показать статистику по каждому тестовому запуску и прохождению.

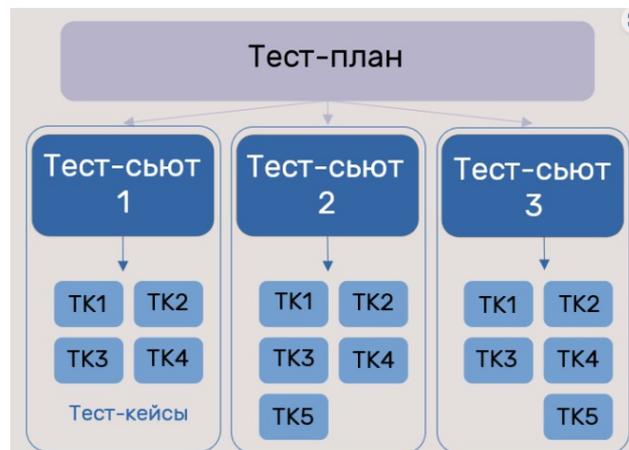
Контрольные вопросы:

1. Как запустить тест кейс?
2. Как отмечать пройденные шаги?
3. Как можно посмотреть запущенные тест-кейсы?
4. Как посмотреть итоги по тест кейсам?
5. Как закрыть запуск тест кейса?
6. Как удалить запуск тест кейса?
7. Можно ли несколько раз подряд запускать один и тот же тест кейс?
8. Где посмотреть общую статистику по выполненным тест кейсам по всем пользователям?

Практическая 8-9

Тема: создание тестовых случаев на основе сайта Ozon. Создание, редактирование и работа с тестовым планом. Настройка конфигурации для тестирования. Тестовые запуски с двумя параметрами.

ТЕОРЕТИЧЕСКАЯ ЧАСТЬ



Тест план — это артефакт тестирования, описывающий действия, которые будут происходить в процессе тестирования — от разработки стратегии до критериев поиска ошибок. Он также описывает логику завершения задач и оценку рисков со сценариями их разрешения.

Виды тест планов

Несмотря на стандартную структуру, существует несколько типов тестовых планов. Они отличаются особенностями описанных задач и объемом работ. QA-команды, как правило, используют следующую классификацию:

- **Тест планы по уровням** — планы модульного, интеграционного, системного, приемочного тестирования
- **Тест планы по типам** — планы функционального

тестирования, тестирования производительности или юзабилити, план автоматизированного тестирования и т.д.

- **Мастер тест план** — это комплексный план тестирования. Включает высокоуровневую информацию, которая не часто меняется в ходе тестирования и требования к которой не часто пересматриваются.

По сравнению с простым тест планом мастер тест план более статичен. Это ключевое различие. Как правило, команда проекта использует один мастер тест план и несколько более подробных тест планов для разных уровней или типов тестирования, в которых описываются отдельные модули одного приложения.

Вы можете создать тест план любого типа без использования каких-то особых инструментов. Вам может повстречаться выражение «Инструменты управления тест планами», но это неточная формулировка. Тест план — это документ, и единственный инструмент, который вам нужен для управления им, это текстовый редактор. Обычно речь идет об инструментах управления тестированием

Здесь все более интересно:

Title	Run time	Cases
пример Created 1 second ago	00:00:00	4 test cases

План Qase — план представляет собой шаблон для будущего рана.

План содержит название и предполагаемое время прогона — формируется на основе срока прохождения кейсов (если они прогонялись). Удобно, если хотите дать прогноз по времени прогона.

При создании плана можно дать название, описание и набрать кейсы по условию. Также при выборе кейсов доступен ассайн на выбранного человека.

Из минусов — нет динамических фильтров. Если набрали кейсы по условию, а потом они перестали туда входить — план не изменится.

Взаимодействие с Тестовыми ранами

Как и в TestRail, это набор кейсов для их прогона в рамках релиза или фичи. Особенность — есть визард, своеобразный интерфейс для прохождения рана.

Несколько входных точек для создания рана:

1. Выбрать план — на его основе создать ран.
2. Зайти в раздел Test Runs — создать там.

3. Выделить один или более кейсов в репозитории — стартовать при помощи панели действий (кнопка Run).

Управление планом тестирования

Где найти сведения о плане тестирования?

После создания нового плана тестирования его можно найти в разделе «Планы тестирования». Там вы увидите некоторые детали плана: время выполнения, количество обращений, время создания и время обновления.

Примечание: После того, как тестовый запуск был создан из этого плана, при открытии этого плана вы увидите вкладку «Тестовые запуски», которая показывает все запуски, начатые на основе этого плана.

Дополнительные опции

В трехточечном меню можно редактировать, удалять или экспортировать план тестирования в формате XLS:

Внесение изменений в план тестирования

Если вы вносите изменения в план тестирования (добавляете или удаляете в него или из него регистры), это может повлиять на текущие активные тестовые запуски, использующие этот план. В таблице ниже показано, какие изменения произойдут:

	Активный тестовый запуск, нет результатов, Запуск не завершен	Активный тестовый запуск, Результаты добавлены, но прогон не завершен	Завершенный тестовый запуск
Тестовый случай добавляется в план	Кейс добавлен в прогон	Кейс добавлен в прогон	Регистр НЕ добавляется к запуску
Тестовый случай удален из плана	Кейс снят с прогона	Чехол НЕ снимается с прогона	Чехол НЕ снимается с прогона

Для чего используются конфигурации?

Конфигурации можно использовать для указания для любого тестового запуска, на каком оборудовании или программном обеспечении должен выполняться этот запуск.

Вы можете создать несколько конфигурационных групп и иметь комбинации, например, разных браузеров и разных ОС:

Configurations

The screenshot shows the 'Configurations' section of a testing tool. It features two main dropdown menus: 'Browsers' with 'Chrome' selected, and 'OS' with 'Windows' selected. Below the 'OS' dropdown, a list of options is visible: 'Windows', 'Linux', and 'MacOS X'. Underneath these menus, it indicates '0 test cases selected' and provides a '+ Add cases' button.

Вы также можете редактировать или удалять как группы конфигурации, так и отдельные конфигурации после их создания

ХОД РАБОТЫ

1. Запустите систему тестирования
2. Откройте проект

3. Перейдите на пункт конфигурации [Configurations](#) и нажмите создать конфигурацию

Задание 1 создать конфигурацию под разные ОС и браузеры, ввести 3 разных ОС и 5 различных браузеров и конфигурацию размеры экранов: в них создать (1920*1080, 1366*768, 1440*900, 768*1024)

1. Нажмите создать конфигурацию

.1 в диалоговом окне задайте название данной конфигурации ОС

Configurations

Create configuration group

ОС

Windows

Linux

Mac

+ Create configuration

Браузер

хром

яндекс

опера

сафари

бинг

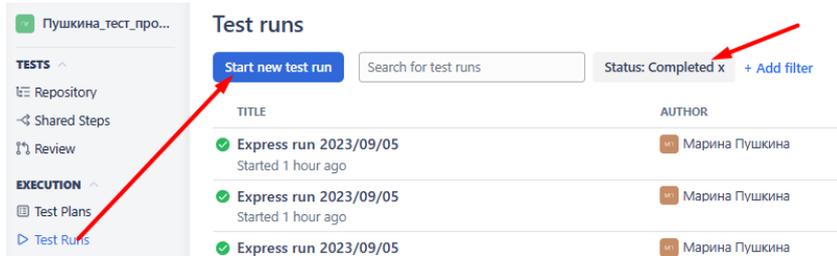
+ Create configuration

-после чего нажмите+ и введите конкретные ОС

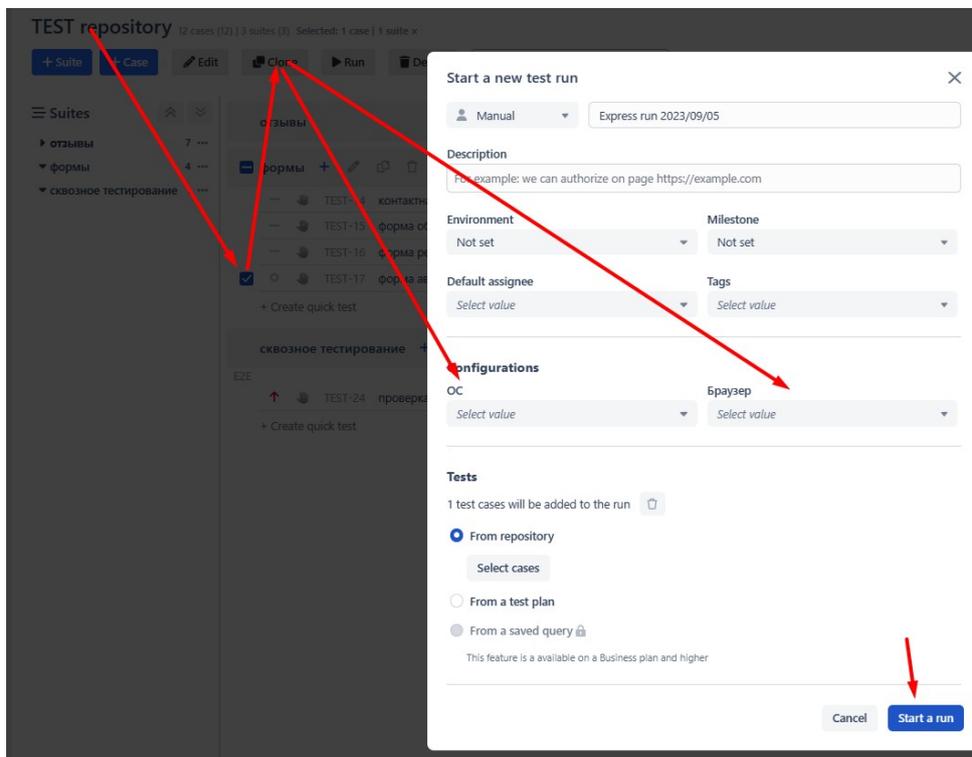
2 в диалоговом окне задайте название данной конфигурации Браузер

-после чего нажмите+ и введите конкретные Браузеры

1.3 после чего перейдите в тестовый репозиторий выберите , конкретный тест-кейс и при запуске в диалоговом окне выберите конкретную ОС и браузер или через тест ран и установленный фильтр

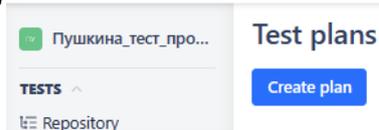


Данные конфигурации позволят при создании тестовых запусков выбирать среды, ОС, но при этом не создавать параметры или использовать определенные ОС не все сразу, тест-кейс становится более гибким, но при этом можно дополнительно создавать отдельные настройки для каждого тест-кейса.



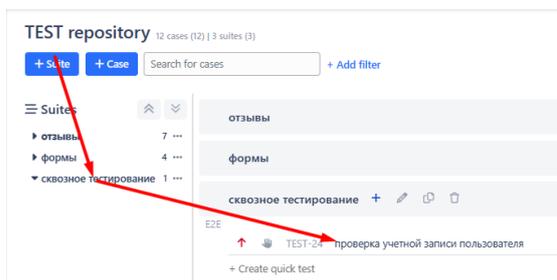
Задание 2 создать и запустить 2 вида различных тест-плана, в каждом тест плане по 2 сквозных тест-кейса

1.1 выберите из пунктов меню тест план
1.2 - нажмите создать план



2.1 дайте название виду плана Сквозное тестирование

3. Откройте репозиторий тест-кейсов и создайте **сквозные тесты** и добавьте 4 сквозных тест-кейса с различными критическими сценариями



1й тест кейс в репозитории создайте по примеру:

➤ Установите базовые свойства: свойства для сквозного тестирования учетной записи

← Edit test case

Basic

Title *	проверка учетной записи пользователя	Status *	Actual
Description	For example: We can authorize on page http://example.com/login		
Suite	сквозное тестирование	Severity *	Critical
Type *	Functional	Layer *	E2E
Milestone	БАЗОВЫЙ ЭТАП	Behavior *	Positive
		Priority *	High
		Is flaky *	No
		Automation status *	Not automated

Conditions

Pre-conditions	1. перейти на сайт https://www.ozon.ru/	Post-conditions	Empty value
----------------	--	-----------------	-------------

➤ Установите Дополнительные параметры

Parameters

Parameter title	VPN	Parameter values	Россия
			Франция
			+ Add value

➤ Задайте шаги сквозного тестирования проверки учетной записи пользователя, шаги

← Edit test case

1	в меню выбрать Войти	нажать ВОЙТИ	
1	войти с телефона	+79851116955	форма для ввода кода
1	поле кода	1235	в системе под своей учеткой  Марина
2	войти с APPLE	логин: "арлрва" пароль: "гипп"	в системе под своим именем  Марина
3	вход через гос услуги	логин: "арлрва" пароль: "гипп"	в системе под своим именем  Марина
4	войти по почте	логин: "арлрва" пароль: "гипп"	в системе под своим именем  Марина

+ Add step + Add shared step

➤ Сохраните тест-кейс и запустите его в конфигурации виндовс -сафари, пройдите тест кейс

ID	MEMBER	STATUS	TITLE	DURATION
24	Марина Пуш	Passed	проверка учетной записи пользователя VPN: Россия	53s 918ms
24	Марина Пуш	Untested	проверка учетной записи пользователя	

4. Теперь вернитесь на вкладку тест план и нажмите там добавить кейс

4.1 в диалоговом окне выберите те группы, которые будут относиться к сквозному тестированию в данном случае это Сквозные тесты

4.1 нажмите DONE

4.2 после чего **Create plan**

4.1 в диалоговом окне выберите те группы, которые будут относиться к сквозному тестированию в данном случае это Сквозные тесты

4.1 нажмите DONE

4.2 после чего **Create plan**

Select test cases

Search for cases

- ОТЗЫВЫ 0/7
- формы 0/4
- сквозное тестирование 1/1

Контрольные вопросы

1. Что такое тест план? Как его создать и запустить?
2. Как создать конфигурацию и где она применяется?
3. Можно ли одновременно использовать и конфигурацию и параметры?
4. Как отображаются параметры при запуске тестового кейса?

Практическая 10-11

Тема: Тестовые запуски. Дефекты. Описание дефектов в баг-репорте и отправка на доску задач Trello. Генерация автоматизированный, системный тест-кейс с помощью искусственного интеллекта qase.

ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

Если кратко, то хороший баг-репорт позволяет:

1. воспроизвести проблему (это не всегда возможно, но надо стремиться).
2. понять, в чем проблема и какова ее важность.

Как написать хороший баг-репорт?

Для начала надо подготовиться. Если вы обнаружили баг, не стоит моментально бежать в баг-трекер и писать «ничего не работает!». Воспроизведите ошибку. Воспроизвелась? Отлично. Не воспроизвелась? Значит, что-то вы не учли. Вспоминайте, что делали.

Снова воспроизвелась? Ура! А теперь минимизируйте количество шагов для воспроизведения, удостоверьтесь, что нет ничего лишнего.

Если используются какие-то входные данные, удостоверьтесь, что и они не содержат лишнего (действительно ли весь этот здоровенный кусок текста роняет редактор, а не один символ из него?).

Когда вы поняли, какие именно данные и какие ваши действия приводят к проблеме, кратко сформулируйте ее суть — придумайте заголовок баг-репорта. Опишите проблему настолько подробно и конкретно, насколько позволяет заголовок, при этом не увлекаясь его длиной в описание желательно вдавить код ошибки и из вкладки сеть, пункт полезная, код.

Теперь откройте баг-трекер, начните заполнять форму баг-репорта. Запишите **заголовок**.

В каких-то баг-трекерах поля «Подробное описание» и «Шаги для воспроизведения» различаются, в каких-то — нет. Если поле **«Подробное описание»** есть, опишите в нем проблему более подробно — уточните те детали, которые пришлось опустить в заголовке. Если вы понимаете, в чем причина проблемы (используется устаревшая формула для расчетов, не учитывается какое-то значение и т.д.) — тоже пишите все здесь. Если не знаете — лучше не гадайте.

Если в форме записи об ошибке нет отдельного поля Affect version (версия продукта, в котором проявляется проблема), то укажите версию здесь.

«Шаги для воспроизведения» — основное поле для заполнения в баг-репорте.

Запишите шаги, которые вы определили. Как уже было сказано, шагов должно быть необходимо и достаточно для воспроизведения проблемы. Лишние не пишите. Необходимых тоже не пропускайте :)

После описания шагов обязательно напишите результат — что получилось.

Далее здесь же опишите ожидаемый результат, если это необходимо. Конечно, не стоит писать «Редактор не падает», но если, например, результаты расчетов не соответствуют ожидаемым, то это надо указывать.

Таким образом, описание шагов для воспроизведения должно выглядеть как-то так:

Шаги для воспроизведения:

1. Открыть...
2. Кликнуть...
3. Ввести в поле... значение $N1$
4. Ввести в поле... значение $N2$
4. Кликнуть кнопку Calculate

Результат:

В поле Result отображается $V1$.

Ожидаемый результат:

В поле Result отображается $V2$.

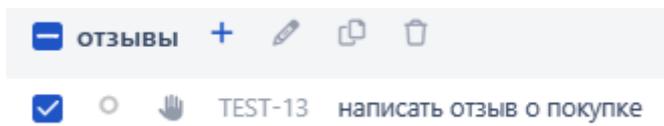
Если требуются исходные файлы, данные, дампы и пр. — сразу приаттачьте. Само собой, файлы должны содержать только информацию, необходимую для воспроизведения ошибки. Подчистите все лишнее. Если проблема с визуальным отображением, то скриншот обязателен — можно будет понять ошибку и без воспроизведения шагов.

ХОД РАБОТЫ

1. Откройте систему
2. Зайдите в проект

Задание 1 запустить тест кейс, во время проверки допустить несколько ошибок и отправить отчет

3. Выберите из репозитория тест-кейсов Написать отзыв о покупке

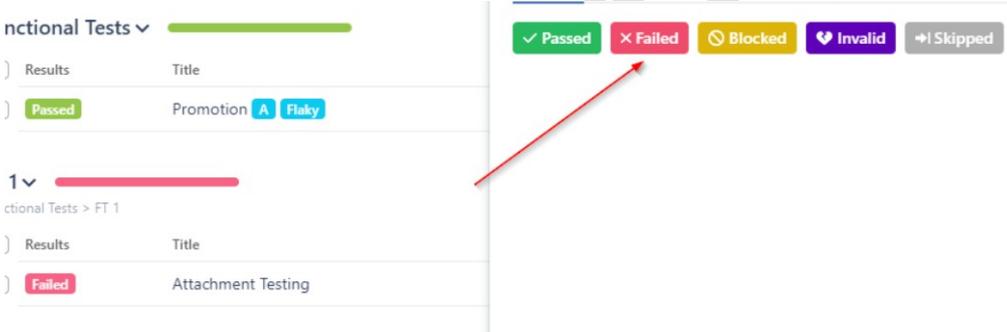


4. Запустите данный тест кейс с конфигурацией сафари

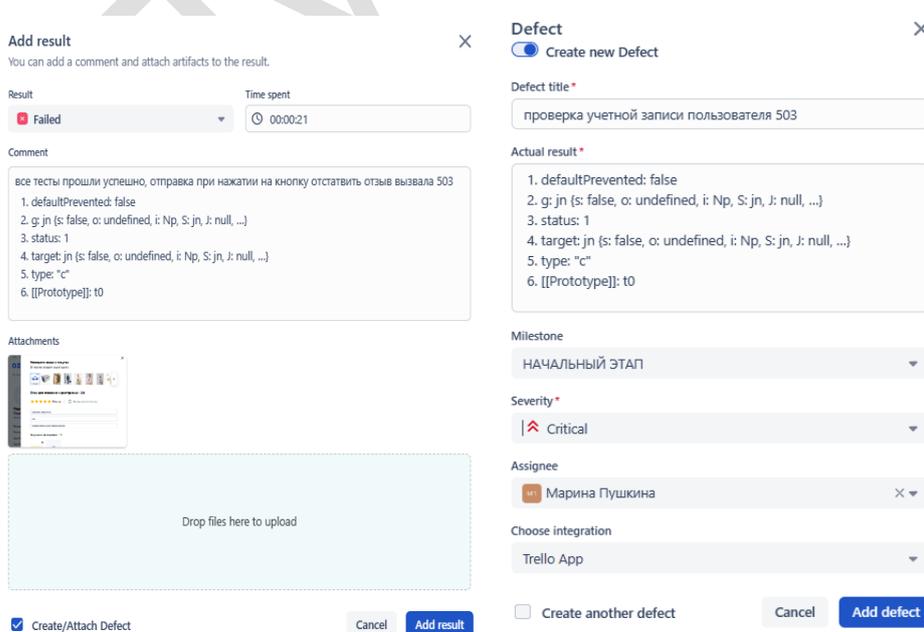
Configurations



5. При прохождении данных этапов допустите ошибку



6. и в поле комментария опишите проблему после чего возникла ошибка, вставьте код ошибки со вкладки полезные данные и заголовком, добавьте скрин с результатом к



ошибке вставьте данные баги, которые нужно скопировать из того, что вы увидите, укажите этап, на кого назначить, критичность ошибки, и добавьте в задачу в трело

Trello App issue

Title*

проверка учетной записи пользователя 503

Description

```
### Preconditions
1. перейти на сайт [https://www.ozon.ru/](https://www.ozon.ru/)
### Steps to reproduce
** *Step 1*
Action: в меню выбрать ВойтиExpected Result: [image.png](https://qase-app-prod.s3.us-east-2.amazonaws.com/public/team/92f1e3bcf6a3fc529e3fb91c16896de9668376b7/attachment/c6369c2ec41c57f9cc0de3a358ad71fbfe299f/image.png)** *Step 1*
Action: войти с телефонаExpected Result: форма для ввода кода
** *Step 2*
Action: поле кодаExpected Result: в системе под своей учеткой[image.png](https://qase-app-prod.s3.us-east-2.amazonaws.com/public/team/92f1e3bcf6a3fc529e3fb91c16896de9668376b7/attachment/8c7cb8be3642eda2da21f69f7928c1b13f2d0de/image.png)
** *Step 3*
Action: войти с APPLEExpected Result: в системе под своим именем[image.png](https://qase-app-prod.s3.us-east-2.amazonaws.com/public/team/92f1e3bcf6a3fc529e3fb91c16896de9668376b7/attachment/8c7cb8be3642eda2da21f69f7928c1b13f2d0de/image.png)
```

Board*

тест1

List*

текущие

5. при отправке отзыва на озон в сафари, была ошибка, чтобы ее просмотреть нажмите F12

5.1.1 перейдите на вкладку сеть

5.1.2 скопируйте и вставьте url, код ошибки, метод и код из вкладки заголовка

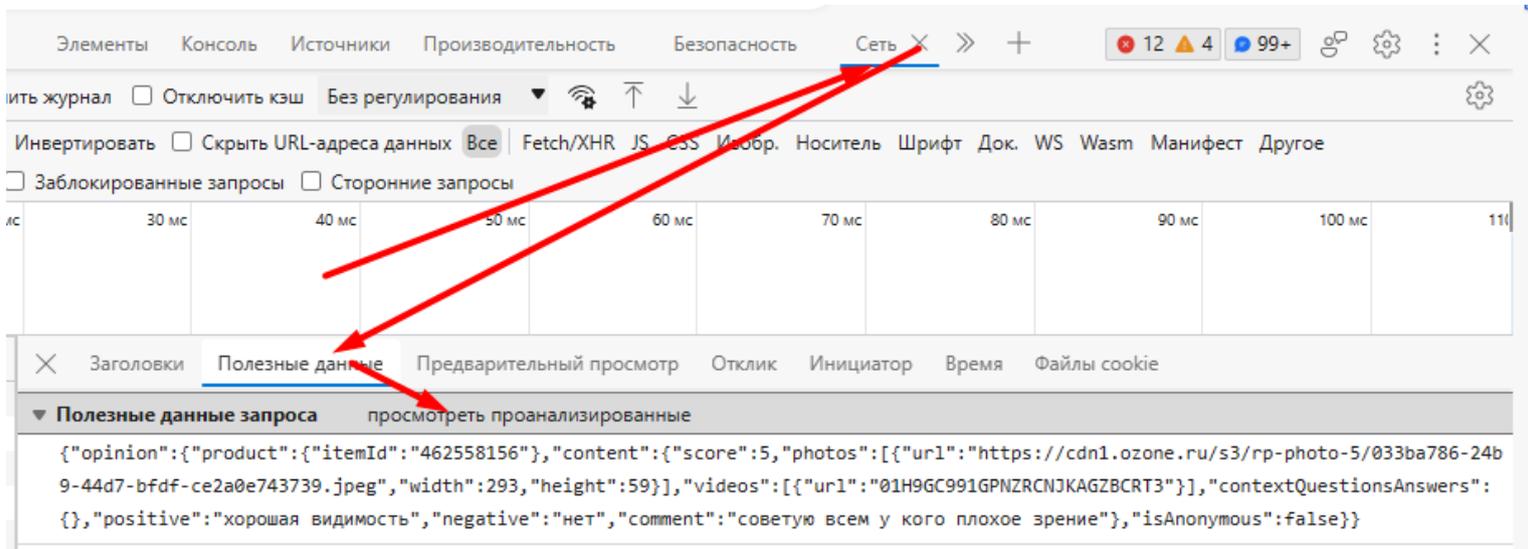
5.1.3 с вкладки полезные данные скопируйте код

5.1.4 с вкладки предварительный просмотр скопируйте код и вставьте

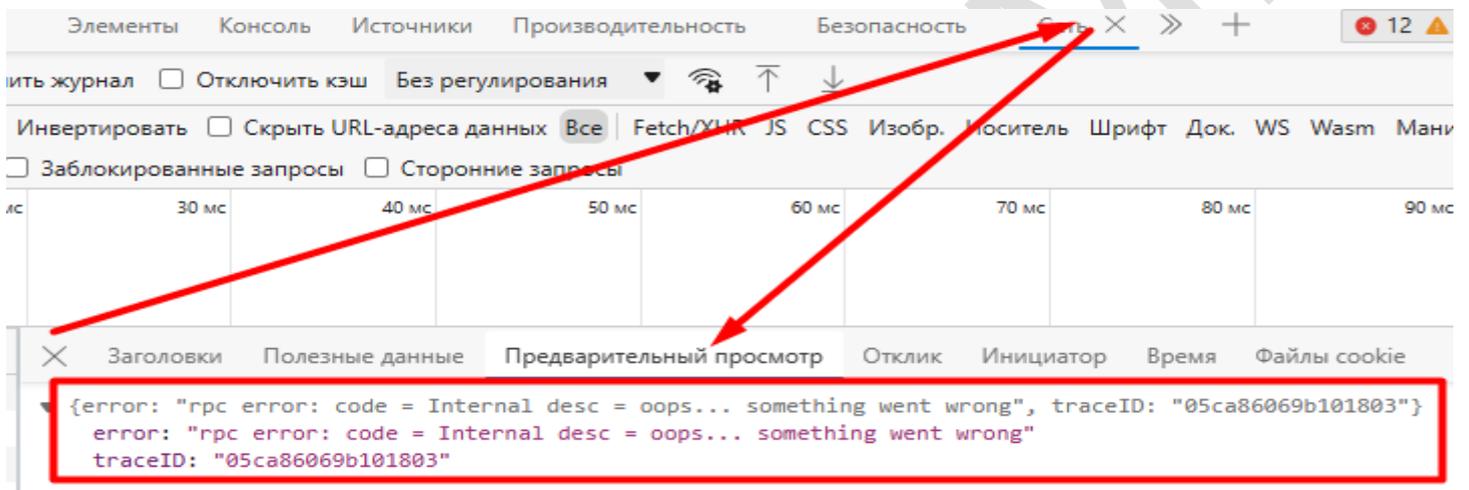
The screenshot shows the Safari Network Inspector. The 'Network' tab is active, displaying a list of requests. A red arrow points to a request with a status of 503. The 'Headers' tab is selected, showing the following details:

- URL-Адрес Запроса: https://www.ozon.ru/api/composer-api.bx/_action/
- Метод Запроса: POST
- Код Состояния: 503
- Удаленный Адрес: 162.159.140.11:443
- Политика Источника: no-referrer-when-downgrade
- Ссылки:

Копируем весь ответ



Копируем ответ



7. В поле прикрепить файл, прикрепите ошибку с визуальной частью, что вам показал сайт

RUB 6

ОZON

Марина Пуш

Личная

Главная

Озон К

Озон Р

Мои ко

Кодовы

сертиф

Баллы

Способ

Озон Р

Баланс

Мой га

Заказы

Моя ко

Мои за

Мои ва

Куплен

ОZON T

Для ме

Сравне

Напишите отзыв о покупке

20 покупок ожидают вашей оценки

Очки для плавания с диоптриями - 2.5

★★★★★ Отлично

Не оценивать эту покупку

хорошая видимость

нет

советую всем у кого плохое зрение

Загружены фотографии 1/10

Загружены видео 1/3

Вы оставляете отзыв как:

Марина П.

Скрыть мои данные в отзыве

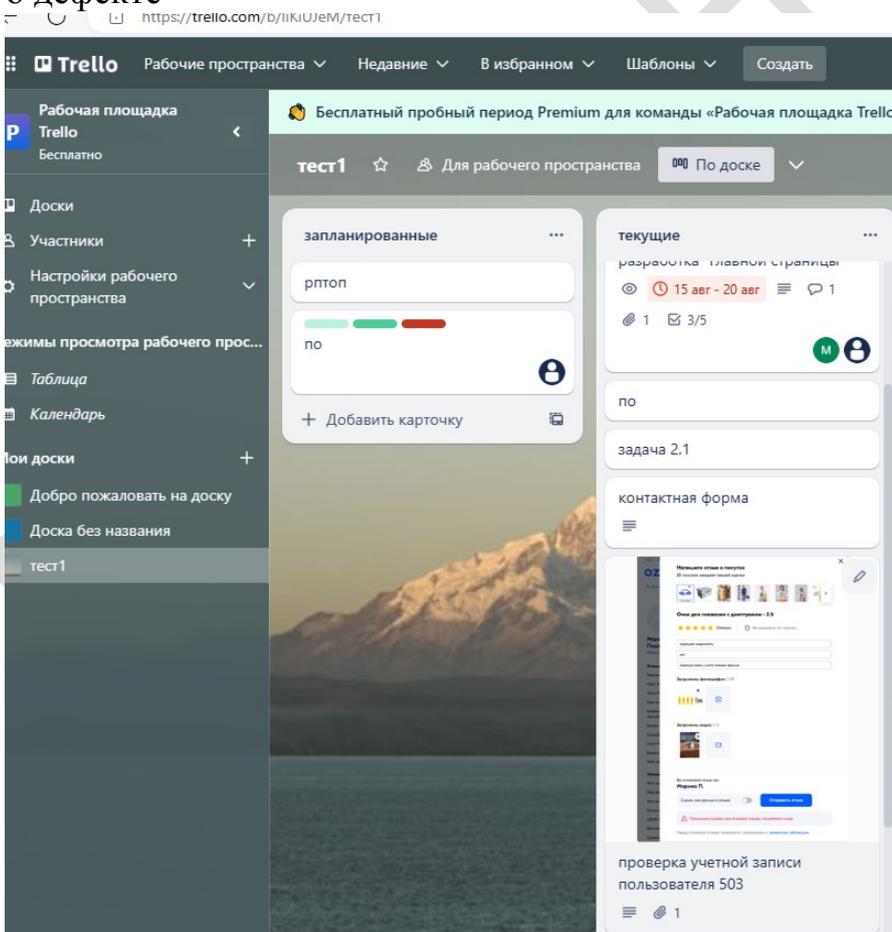
Отправить отзыв

Произошла ошибка при отправке отзыва, попробуйте снова

Перед отправкой отзыва, пожалуйста, ознакомьтесь с правилами публикации.

8. После чего нажмите добавить результат

9. Зайдите в трело к себе на доску и сделайте скрин добавившейся автоматически задачи о дефекте



Задание 2 перейти в раздел дефектов, выбрать дефект, отредактировать описание в формат кода, оставить комментарий и перевести его в состояние прогресса. Найдите в

тест ране данный дефект, перезапустите его и покажите положительную проверку с историей исправлений. Создайте 2 новых дефекта.

← контактная форма 503

Reopen Resolve Invalid Edit

Description

```
{ "opinion": { "product": { "itemId": "742129719" }, "content": { "score": 4, "photos": [], "videos": [], "contextQuestionsAnswers": {}, "pickerValues": { "orderId": "" }, "positive": "anp", "negative": "anp", "comment": "p" }, "isAnonymous": false } }
```

Attachments



Test run results

Test case	Author	Found in run
TEST-14 контактная форма	Марина Пушкина	R-6 - Express run 2023/09/05

Comments

Марина Пушкина Teacher
нужно бы исправить
1 second ago

Write a comment

Status

In Progress

Severity

Minor

Milestone

БАЗОВЫЙ ЭТАП

Reporter

Марина Пушкина

Assignee

Марина Пушкина

Created

05 September 2023 at 15:07:27

Trello-app

контактная форма

Задание 3* Найдите любой ресурс, где есть ошибка и создайте для него тест кейс с фиксацией дефекта

Задание 4 Сгенерировать автоматизированный системный тест-кейс для формы авторизации Азон с помощью qase

1. Выберите тест-кейс для авторизации. Если его нет создайте по форме авторизации для Азон
2. Поставьте рядом с ним галочку и нажмите в свойствах справа
3. В меню выберите
 - 3.1 уровень сквозной
 - 3.2 язык PHP
 - 3.3 фреймверк Phpunit
 - 3.4 включите автоматизацию:
 - 3.4.1 базовый
 - 3.4.2 [ссылку](#) для авторизации азон
 - 3.4.3 номер телефона
 - 3.5 нажмите сгенерировать

Данный код вставляется разработчиками в проект! Или тестировщиками, которые пишут юнит-тесты.

Контрольные вопросы:

1. Что такое дефект? Как его зафиксировать?
2. Как сгенерировать код юнит теста для определенного тест-кейса?
3. Что мы помечаем в дефекте тест-кейса, обязательно?
4. Как сделать, чтобы дефект добавился на доску трело?

Пушкина М.С

Практическая 12

Тема: Selenium IDE для автоматизированного тестирования. Установка, обзор и изучение инструмента. Проверка ссылок и переходов.

ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

Selenium-Это инструмент автоматизированного тестирования, который позволяет пользователям настраивать, редактировать и запускать тесты без какого-либо опыта программирования. С помощью Selenium IDE можно записывать и воспроизводить тестовые сценарии с помощью различных команд, поддерживаемых ядром браузера.

Устойчивые тесты

Selenium IDE записывает несколько локаторов для каждого элемента, с которым он взаимодействует. Если один локатор даст сбой во время воспроизведения, остальные будут проверены до тех пор, пока один из них не окажется успешным.

Повторное использование тестового примера

С помощью команды `run` вы можете повторно использовать один тестовый пример внутри другого (например, позволяя повторно использовать логику входа в систему в нескольких местах пакета).

Поток управления

Selenium IDE поставляется с обширной структурой потока управления и доступными командами, такими как `if`, `while` и `times`.

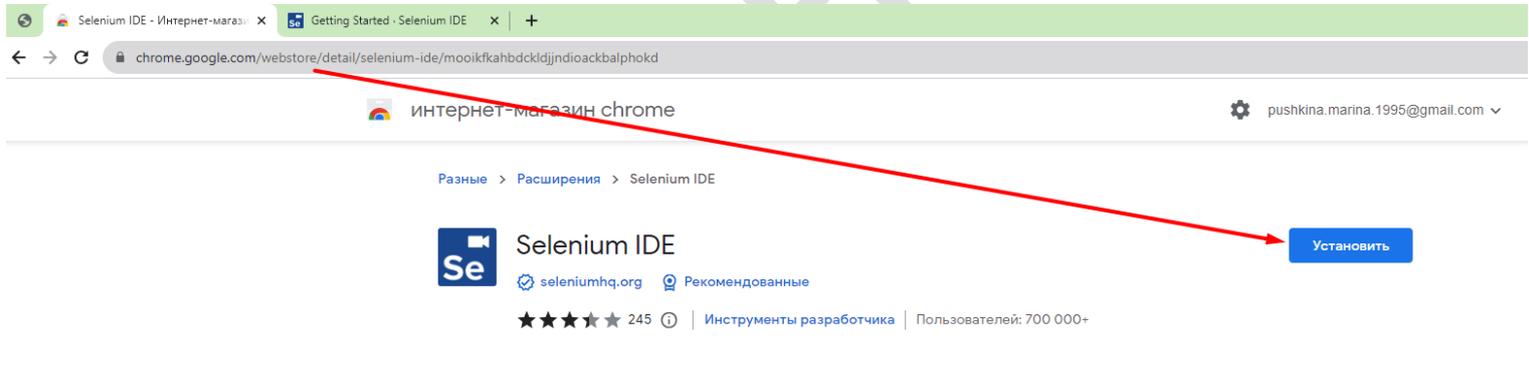
ХОД РАБОТЫ

1. Откройте браузер гугл

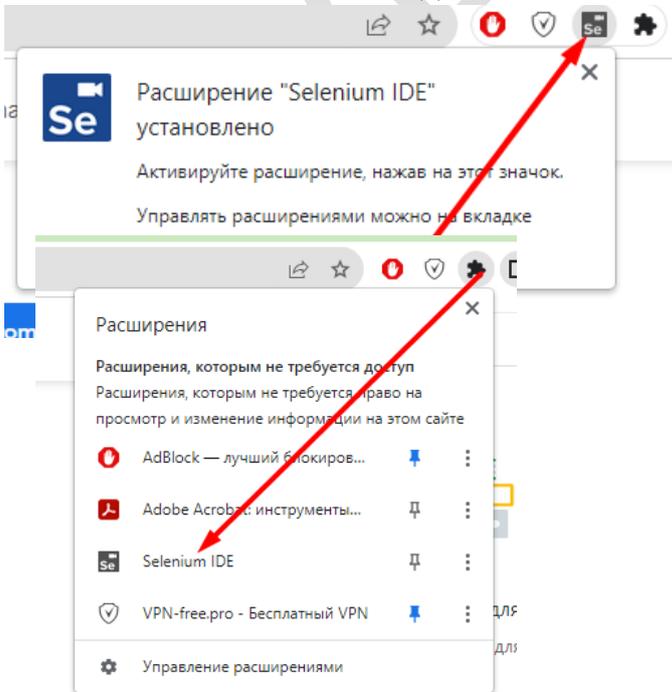
2. Перейдите по ссылке

<https://chrome.google.com/webstore/detail/selenium-ide/mooikfahbdckldjndioackbalphokd>

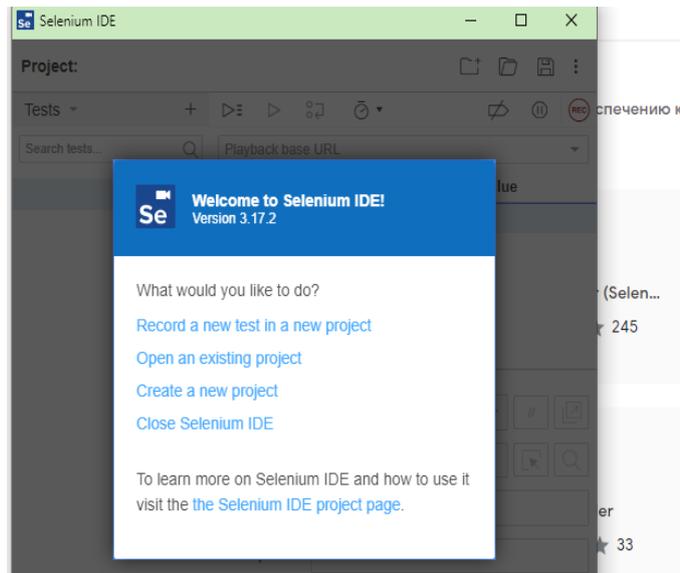
3. нажмите Установить



4. после чего нажмите на данный значек



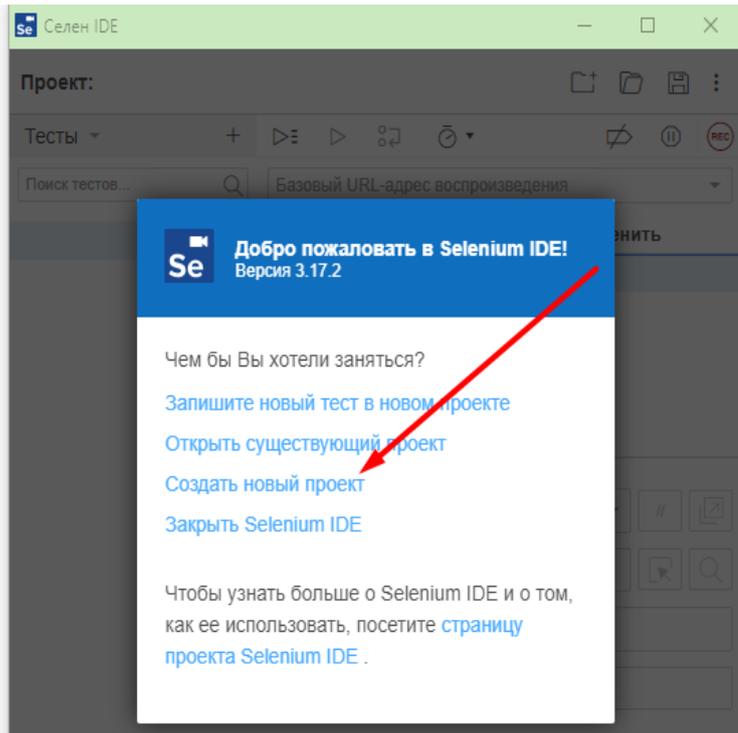
4.1 запустится селениум



Запись ваших действий используется тестировщиков в качестве авто-тестов, для прогона основных последовательностей и поиска ошибок. Используется в регрессионном тестировании с сквозном. В качестве дополнительного инструмента, чтобы тестировщик вручную не тратил время для повтора одинаковых действий на сайте.

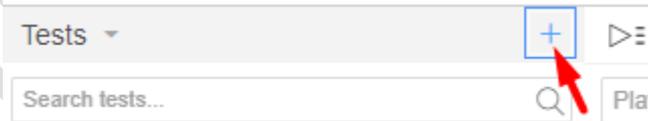
Задание 1 используя селениум создать авто тест. Авто тест сквозного тест-кейса основной последовательности действий на сайте.

1. Для этого нажмите создать новый проект



2. Дайте название проекту азон

3. Нажимаете +

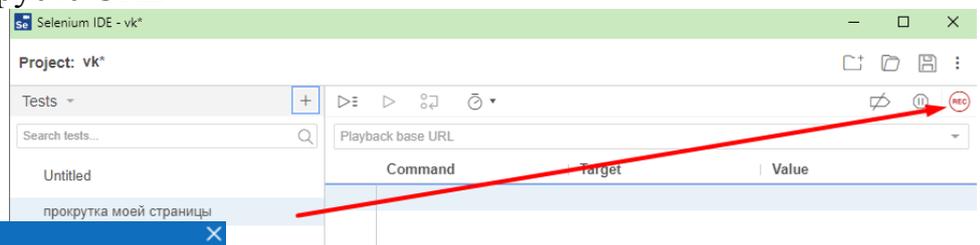


3.1 дайте название Прокрутка главной страницы

3.2 переходите на сайт ВК и копируете URL

3.3 нажимаете REC вставляете URL в поле

3.4 нажимаете старт



3.5 Для начала выполняете самостоятельно действие нажатие Главной страницы и прокрутите до конца своей стены

3.6 по завершению действий нажмите стоп
Вся ваша история запишется, без учета javascript, если вы хотите, чтобы у вас отработал в последовательности джава скрипт, вы должны

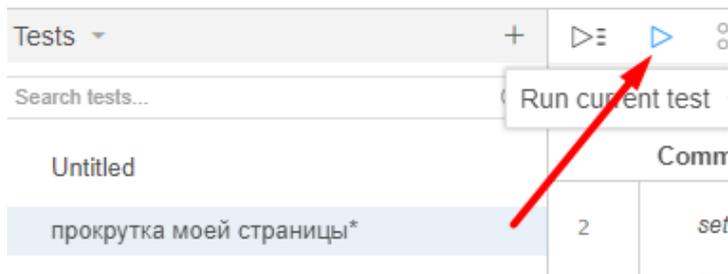
Before you can start recording, you must specify a valid base URL for your project. Your tests will start by navigating to this URL.



при выполнении определенных этапов с javascript наводит мышкой на данные блоки. В этом случае тест выполнится

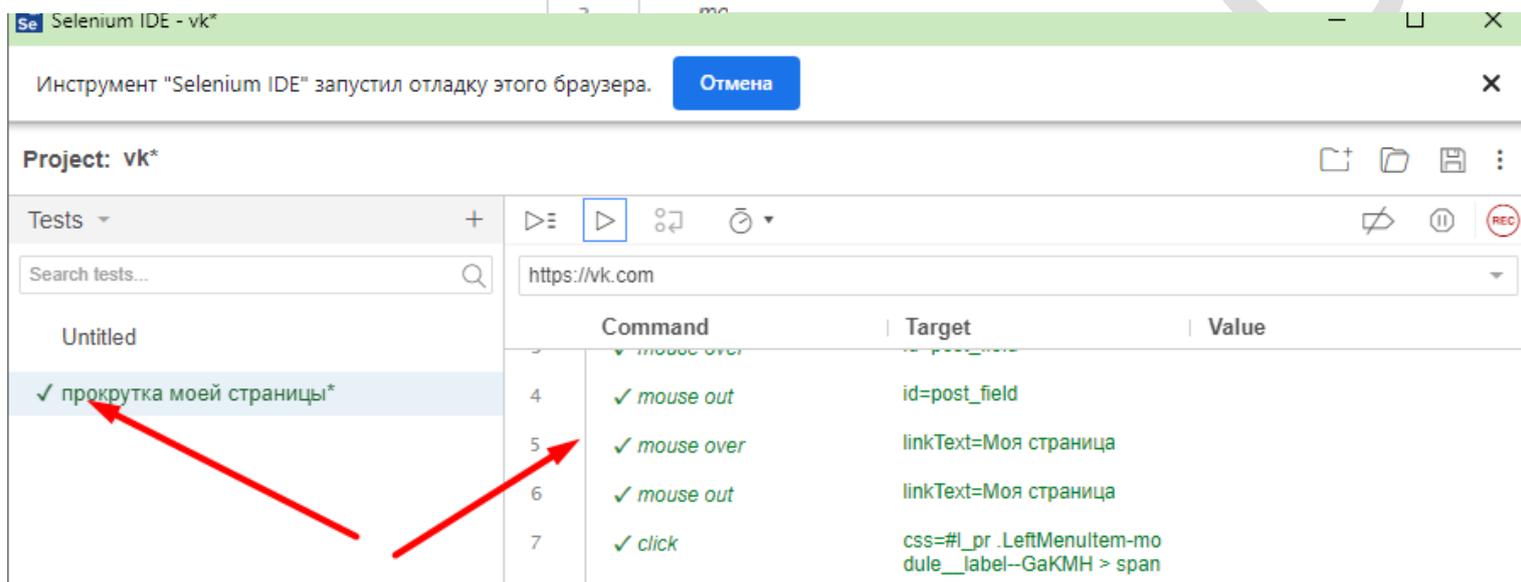
3.7 после чего нажмите Run current test

Project: vk*



Данное действие запустит вашу последовательность вместо вас в реальном времени.

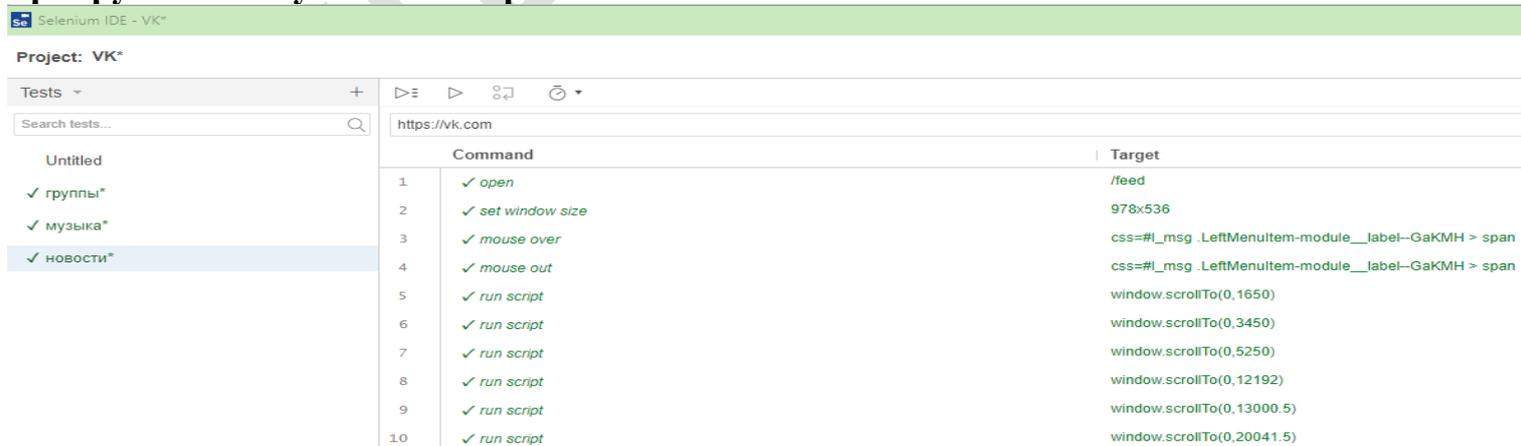
Результат проверки последовательности:



Задание 2 создайте авто тест при нажатии на новости и прокрутите ленту продолжительность 20 сек., завершите тест кейс.

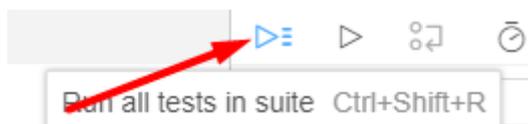
Задание 3 создайте авто тест при нажатии на музыку, нажмите на плей своего река и после чего остановите музыку. Завершите тест кейс.

Задание 4 перейдите в раздел сообщества, выберите сообщество, кликните на него, прокрутите ленту 20 сек завершить тест кейс.



Задание 5 После чего сделайте полный запуск всех тест кейсов последовательно

5.1 аргументируйте по чему тест с музыкой зафиксировал баг? И баг ли это, как это исправить?



5.2 сделайте полное выполнение всех тест кейсов

The screenshot shows the Selenium IDE interface with two test cases. The first test case, 'Музыка', has 6 steps: 1. open, 2. set window size (978x536), 3. mouse over, 4. mouse out, 5. run script (window.scrollTo(0, 1650)), and 6. run script (window.scrollTo(0, 3450)). The second test case, 'Новости', has 7 steps: 2. set window size (978x536), 3. mouse over, 4. click, 5. mouse out, 6. click, and 7. click. A red arrow points from the 'Музыка' test case to the 'Новости' test case.

Step	Command	Target	Value
1	open	/feed	
2	set window size	978x536	
3	mouse over	css=#_l_msg_LeftMenuItem-module__label-GakMH > span	
4	mouse out	css=#_l_msg_LeftMenuItem-module__label-GakMH > span	
5	run script	window.scrollTo(0, 1650)	
6	run script	window.scrollTo(0, 3450)	

Step	Command	Target	Value
2	set window size	978x536	
3	mouse over	css=#_l_aud_LeftMenuItem-module__label-GakMH > span	
4	click	css=#_l_aud_LeftMenuItem-module__label-GakMH > span	
5	mouse out	css=#_l_aud_LeftMenuItem-module__label-GakMH > span	
6	click	css=_l_audio_row_42524730_456239425_audio_row__cover_icon	
7	click	css=_l_audio_row_42524730_456239425_audio_row__cover_icon	

5.3 сохраните свой проект

Задание 6 выберите любой ресурс и проверьте 4 страницы на битые ссылки.

Контрольные вопросы:

1. Что такое селениум?
2. Как его применяют и в каких случаях?
3. Как проверить последовательность с джава скриптом?
4. Как запустить несколько созданных автоматически?

Практическая 13-14

Тема: тестирование ресурса wordclass. Адаптивность под популярные мобильные устройства и операционные системы. Панель разработчика от гугл хром, и сафари. Использование инструмента iloveadaptive. Тестирование и анализ скорости загрузки страницы с помощью pagespeed.

ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

ILoveAdaptive-Сервис, с помощью которого можно проверить адаптивность под популярные мобильные устройства и операционные системы.

Хороший инструмент проверки адаптивности на различных мобильных устройствах.

Преимущества:

- Есть возможность посмотреть, как выглядит сайт на разных операционных системах (IOS и Android);
- Сам понимает, есть ли мобильная версия;
- Расширение для Google Chrome.
- Проводит детальную валидацию ошибок
- Анализирует скорость загрузки страницы на мобильном и десктопном варианте
- Есть возможность сравнивать сверстаный ресурс с макетом

PageSpeed Insights API (PSI) позволяет получать отчеты о скорости загрузки страниц на мобильных устройствах и компьютерах, а также советы, как эту скорость увеличить.

PSI предоставляет как данные о том, насколько быстро страница загружалась у настоящих пользователей, так и данные, полученные в результате имитации процесса загрузки. Поскольку имитация выполняется в управляемых условиях, с ее помощью удобно выявлять и устранять проблемы со скоростью, но есть риск упустить некоторые из тех, которые возникают в действительности. Данные же наблюдений от пользователей отражают реальное положение дел, но набор доступных показателей ограничен. Более подробная информация об этих двух типах сведений представлена на странице [Что нужно понимать, работая с инструментами для оптимизации скорости загрузки.](#)

Оценка скорости загрузки

Вверху отчета PSI показывается общая оценка скорости загрузки страницы в баллах. Эта оценка рассчитывается по итогам имитации загрузки с помощью инструмента Lighthouse. Результат от 90 баллов и выше считается хорошим, от 50 до 90 – средним. Если набрано меньше 50 баллов, значит страница загружается медленно.

Данные наблюдений

После того как вы даете PSI задание проанализировать страницу по определенному URL, выполняется поиск сведений о ней в отчете об удобстве пользования браузером Chrome. В отчет PSI включаются доступные данные по показателям первой отрисовки контента (FCP) и первой задержки ввода (FID) для всего источника или конкретной страницы с указанным URL.

Высокая, средняя и низкая скорость загрузки

При анализе собранных у пользователей данных из отчета об удобстве пользования браузером Chrome сервис PSI распределяет страницы по трем категориям: с быстрой, обычной и медленной загрузкой. Критерии классификации указаны в таблице ниже.

	Высокая скорость	Средняя скорость	Низкая скорость
FCP	0–1000 мс	1000–2500 мс	Более 2500 мс
FID	0–50 мс	50–250 мс	Более 250 мс

В целом примерно у 10 % страниц наблюдается высокая скорость загрузки, у следующих 40 % – средняя, а у последних 50 % – низкая. Числа округлены для простоты понимания. Приведенные в таблице пороговые значения одинаковы для мобильных устройств и компьютеров и введены с учетом особенностей человеческого восприятия.

Распределение значений FCP и FID, а также выбранное для каждого показателя значение

В PSI показывается распределение значений FCP и FID, полученных в результате анализа данных об определенной странице или источнике. Категорий при этом тоже три: "Быстро", "Средне" и "Медленно". На диаграмме распределения они обозначены зеленым, оранжевым и красным цветом соответственно. К примеру, если в оранжевой части диаграммы напротив показателя FCP указана доля 14 %, значит 14 % всех

имеющихся значений FCP находится в диапазоне от 1000 до 2500 мс. Это агрегированные данные о загрузках страницы за прошедшие 30 дней.

First Contentful Paint

2.3 s ✓



Над диаграммой распределения в PSI приводится 90-й перцентиль по FCP в секундах и 95-й перцентиль по FID в миллисекундах. Эти перцентили выбираются, чтобы разработчики могли понять, насколько медленно их страницы загружаются в наихудших условиях. Показатели с определенными значениями перцентилей классифицируются как высокие, средние или низкие по тем же критериям, которые указаны в таблице выше.

Общий результат

Скорость загрузки страницы оценивается на основе значений обоих показателей:

- Высокая – если и FCP, и FID высокие.
- Низкая – если FCP или FID низкий.
- Средняя – во всех остальных случаях.

Различия между данными PSI и отчета об удобстве пользования браузером Chrome

Данные о фактической скорости загрузки в PSI обновляются ежедневно и охватывают последние 30 дней, в то время как данные отчета об удобстве пользования браузером Chrome, просматриваемые в BigQuery, обновляются только раз в месяц.

Имитация загрузки страницы

Инструмент PSI с помощью технологии Lighthouse получает различные показатели скорости загрузки страницы с указанным URL, такие как Первая отрисовка контента, Время загрузки достаточной части контента, Индекс скорости загрузки, Время окончания работы ЦП, Время загрузки для взаимодействия и Приблизительное время задержки при вводе.

По каждому показателю дается отдельная оценка с пометкой в виде определенного значка:

- Если показатель высокий, напротив него показывается зеленый кружок с галочкой.
- Если показатель средний, напротив него показывается оранжевый кружок с буквой i.
- Если показатель низкий, напротив него показывается красный треугольник с восклицательным знаком.

Аудиты

Аудит Lighthouse делится на три части:

• **Оптимизация.** В этом разделе даются рекомендации по поводу того, как улучшить показатели скорости загрузки страницы. Кроме того, здесь указывается, насколько быстрее будет загружаться страница, если реализовать ту или иную рекомендацию.

• **Диагностика.** В этом разделе представлены сведения о соответствии страницы рекомендациям в отношении веб-разработки.

• **Успешные аудиты.** В этом разделе перечислены аудиты страницы, которые были пройдены с хорошим результатом.

В каждый браузер встроены инструменты разработчика — они позволяют быстро отловить и исправить ошибки в разметке или в коде. С их помощью можно узнать, как построилось DOM-дерево, какие теги и атрибуты есть на странице, почему не подгрузились шрифты и многое другое.

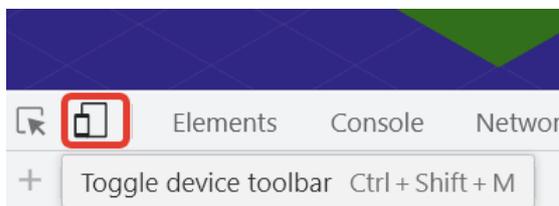
В этом цикле статей мы разберём базовые возможности Chrome DevTools, но их будет вполне достаточно для начала обучения.

Как открыть инструменты разработчика

□□ macOS — Ctrl+Cmd+I

Windows — F12

Linux — Ctrl+Shift+I

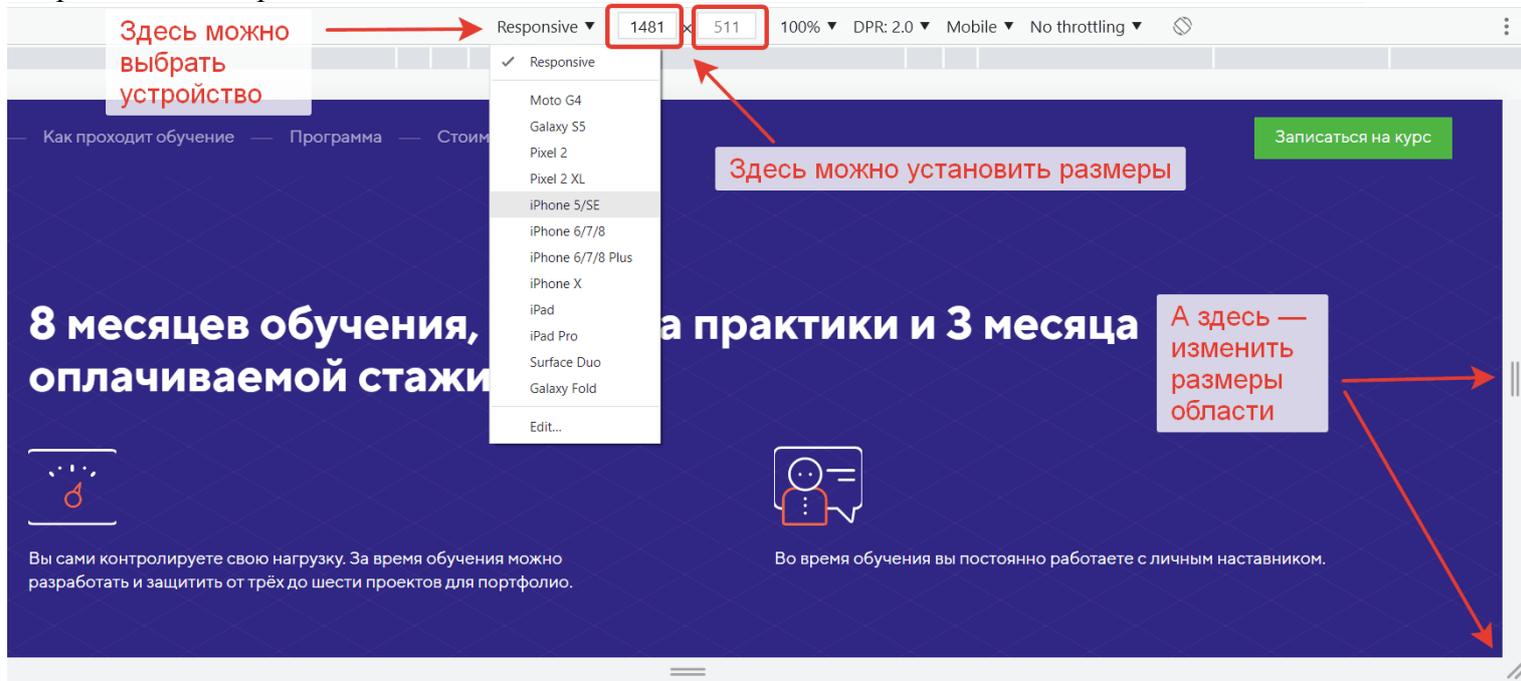


Посмотреть, как выглядит страница с телефона и планшета

При создании адаптивных сайтов или веб-приложений полезно знать, как выглядит страница на планшетах и мобильных устройствах. С помощью инструментов разработчика вы можете

сделать это за несколько секунд. Нужно лишь **открыть Chrome Devtools**, а затем кликнуть на иконку Toggle device toolbar в левом углу или нажать комбинацию Ctrl+Shift+M.

Над страницей появится панель с режимами эмуляции из мобильных устройств и планшетов. По умолчанию панель инструментов открывается в режиме адаптивного окна просмотра. Чтобы изменить область до нужных размеров, задайте ширину или потяните за границы рабочей области. А если хотите увидеть, как страница отображается на конкретных устройствах, например, на iPhone 5, кликните на Responsive и выберите подходящий девайс.



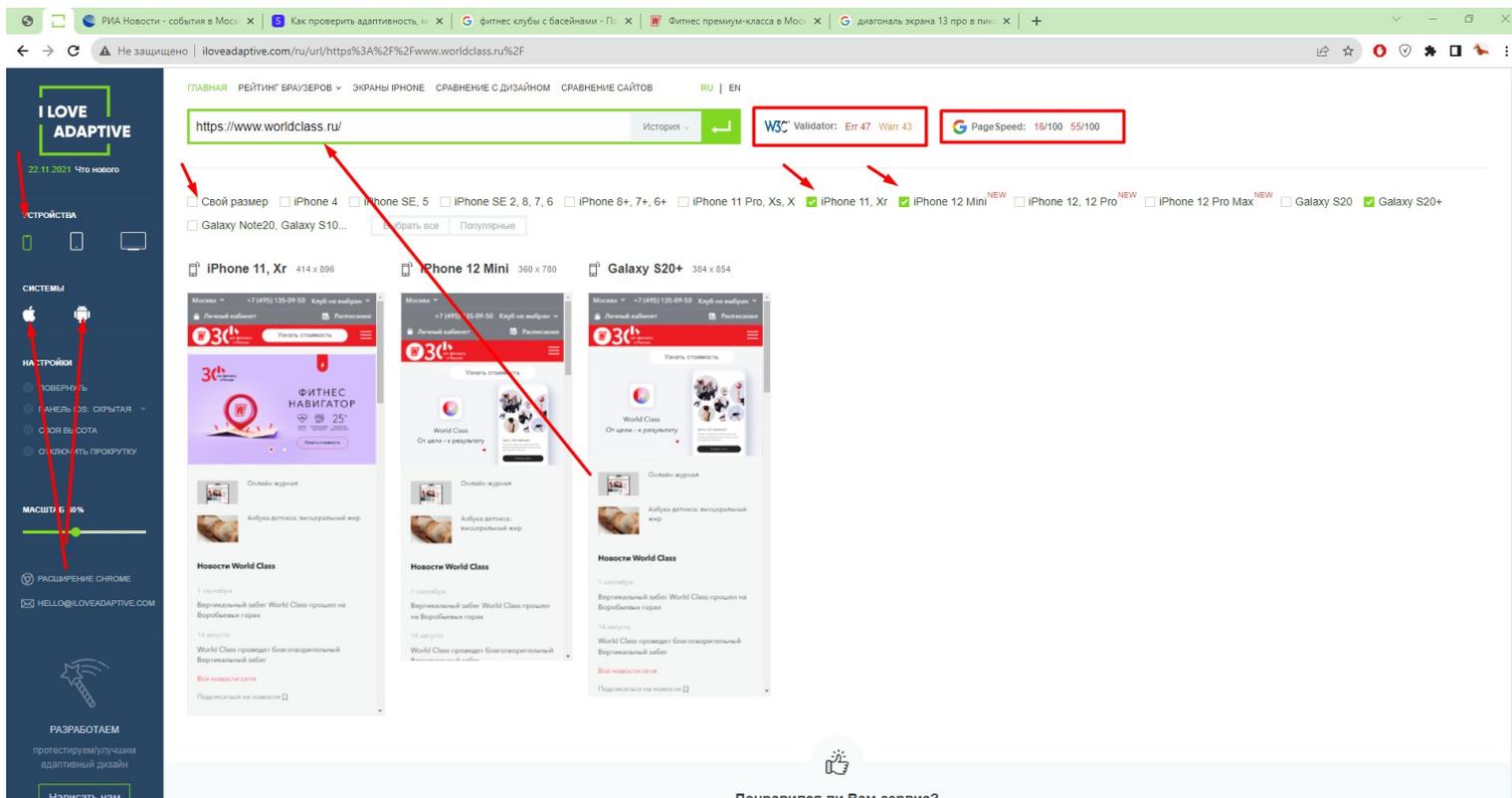
Так выглядит страница в мобильной версии

На этой же панели есть ещё одна полезная кнопка — DPR (Device Pixel Ratio). С её помощью проверяют, как выглядят изображения на ретина-дисплеях — экранах с повышенной плотностью. Чтобы посмотреть, как выглядит графика на разных устройствах, измените значение DPR и обновите страницу.

ХОД РАБОТЫ

Задание 1 выполнить тестирование ресурса под различные экраны и мониторы, выполнить спидтест, валидацию ресурса.

1. Перейдите по ссылке <http://iloveadaptive.com/ru/>
2. Введите URL ресурса wordclass <https://www.worldclass.ru/>
3. Выберите 3 модели телефона и укажите свой размер диагонали телефона



5. Проведите анализ ресурса нажав валидацию:

5.1 Сколько у вас предупреждений-это то что рекомендовано исправить, но не обязательно

1. **Warning** This document appears to be written in Russian. Consider adding `lang="ru"` (or variant) to the `html` start tag.

From line 2, column 16; to line 4, column 6

TYPE `html` → `<html>` → `<html lang="ru">` → `<head>`

For further guidance, consult [Declaring the overall language of a page](#) and [Choosing language tags](#).

If the HTML checker has misidentified the language of this document, please [file an issue report](#) or [send e-mail to report the problem](#).

2. **Warning** Consider avoiding viewport values that prevent users from resizing documents.

From line 9, column 2; to line 9, column 106

`=utf-8">` → `<meta name="viewport" content="width=device-width, initial-scale=1, maximum-scale=2.0, user-scalable=no">` → `<met`

1. Так как русскоязычный ресурс, рекомендовано добавить тег `lang="ru"` в блок `html`

2. Старайтесь избегать присвоения пользовательских параметры `viewport`

5.2 Сколько информационных, они несут информационную нагрузку, можно не учитывать

3. **Info** Trailing slash on void elements [has no effect](#) and [interacts badly with unquoted attribute values](#).

From line 10, column 2; to line 10, column 58

`ble=no">` → `<meta name="apple-mobile-web-app-capable" content="yes"/>` → `<lin`

4. **Info** Trailing slash on void elements [has no effect](#) and [interacts badly with unquoted attribute values](#).

From line 13, column 2; to line 13, column 62

`x-icon">` → `<meta name="yandex-verification" content="9fb444e041f7b620"/>` → `<met`

Лучше не ставить в конце

5.3 Сколько ошибок- данные ошибки нужно отправить на исправления, для повышения качества работы ресурса, его эффективности и для просмотра роботом вашего ресурса при парсинге, в этом случае у вас увеличится скорость загрузки и повысится выдача вашего ресурса в поисковике.

6. **Error** A document must not include more than one `meta` element with a `http-equiv` attribute whose value is `content-type`.

From line 29, column 12; to line 29, column 80

```
<!-- --> <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" /><meta
```

Документ не должен содержать больше, чем одного мета файла с атрибутом `http-equiv` а на данном ресурсе их 2 нужно удалить остальные.

```
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"> == $0
<meta name="viewport" content="width=device-width, initial-scale=1, maximum-scale=2.0, user-scalable=no">
<meta name="apple-mobile-web-app-capable" content="yes">
<link rel="icon" href="/favicon.ico" type="image/x-icon">
<link rel="shortcut icon" href="/favicon.ico" type="image/x-icon">
<meta name="yandex-verification" content="9fb444e041f7b620">
<meta name="facebook-domain-verification" content="eaftn3za5ysktwgas4t7yvn
xvrrkaj">
<link rel="apple-touch-icon" sizes="180x180" href="/apple-touch-icon.png">
<link rel="icon" type="image/png" sizes="32x32" href="/favicon-32x32.png">
<link rel="icon" type="image/png" sizes="192x192" href="/android-chrome-19
2x192.png">
<link rel="icon" type="image/png" sizes="16x16" href="/favicon-16x16.png">
<link rel="manifest" href="/site.webmanifest">
<link rel="mask-icon" href="/safari-pinned-tab.svg" color="#da532c">
<meta name="msapplication-TileColor" content="#da532c">
<meta name="theme-color" content="#ffffff">
<title>Фитнес премиум-класса в Москве, World Class – сеть фитнес клубов
</title>
<!-- jQuery include -->
<!-- -->
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
```

12. **Error** Element `noindex` not allowed as child of element `body` in this context. (Suppressing further errors from this subtree.)

From line 55, column 1; to line 55, column 9

```
label=""><noindex> <!--
```

Content model for element `body`:

[Flow content.](#)

В данном контексте боди не разрешён элемент `noindex`, нужно изменить

Результат все эти ошибки и рекомендации при тестировании у вас пойдут ,как дефекты, которые нужно исправить.

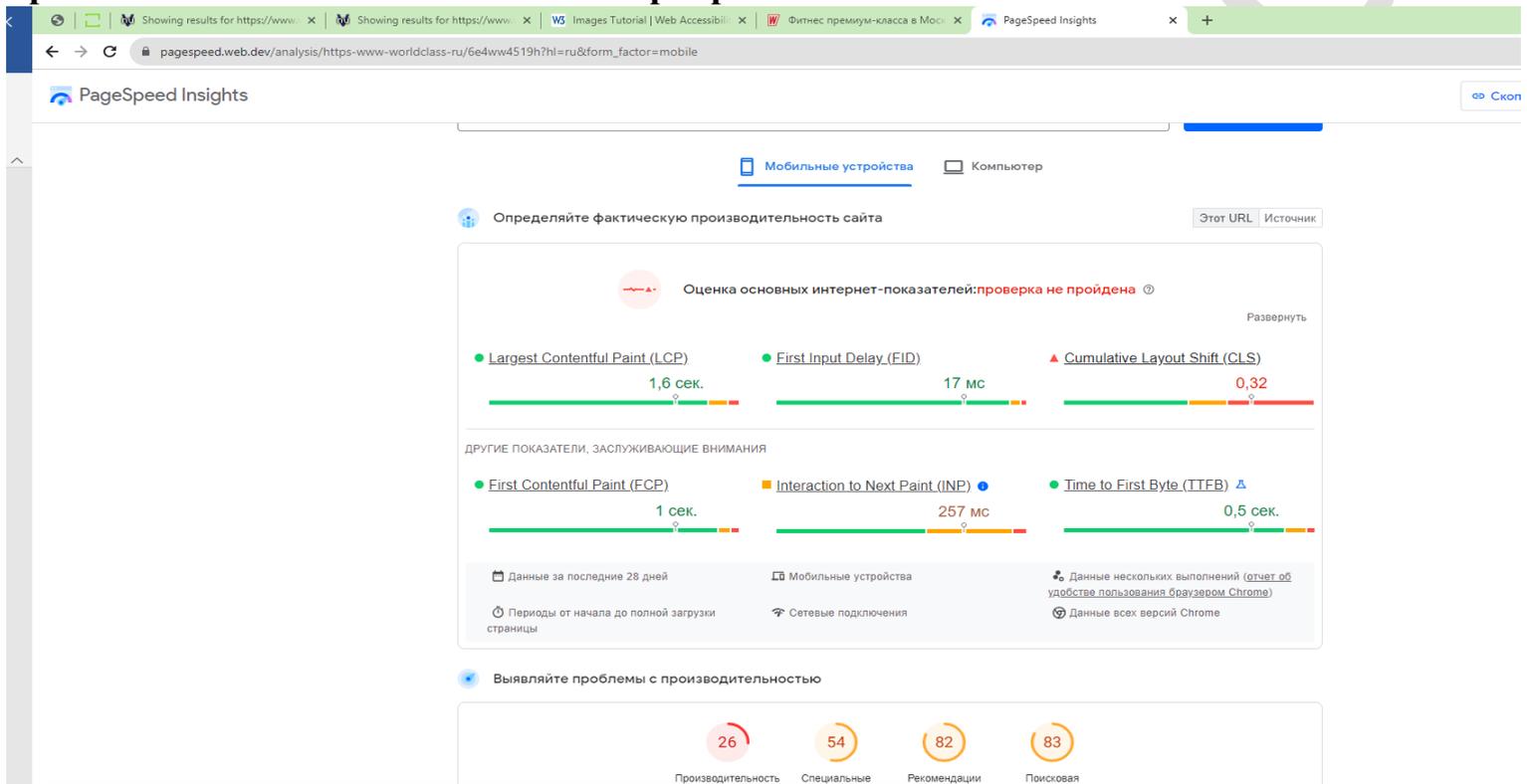
После исправления дефектов, ваш показатель должен изменить не менее 75%, считается плохим ресурсом!

W3C Validator: Err 47 Warr 43

PageSpeed: 16/100 55/100

Задание 2 выбрать ресурс и провести его анализ. Выполнить тестирование ресурса под различные экраны и мониторы, выполнить спидтест, валидацию ресурса. Создать по вашему ресурсу 4 дефекта в ресурсе qise.

Задание 3 перейдите по ссылке <https://pagespeed.web.dev/?hl=ru> и укажите любой ресурс. Проанализируйте показанные вам результаты. Сделайте вывод, и создайте 3 дефекта с критической важностью и высоким приоритетом.

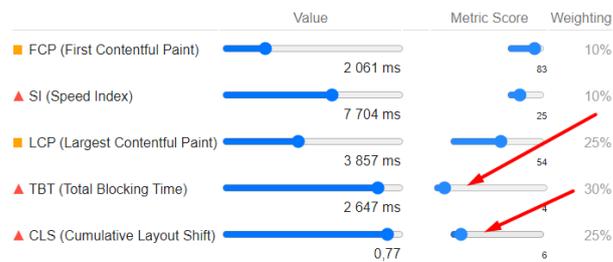
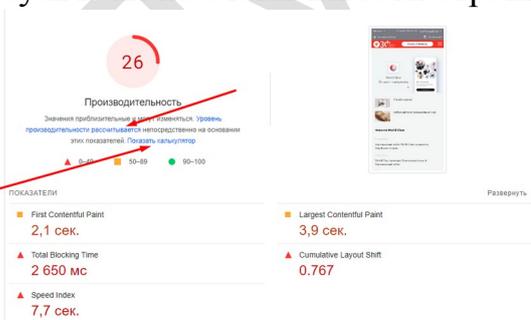


Пример дефекта:

2 дефекты по производительности

▲ TBT (Total Blocking Time)

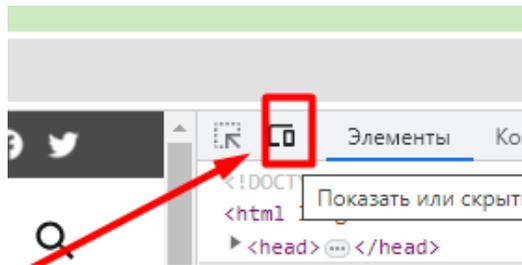
▲ CLS (Cumulative Layout Shift), калькулятор показывает критические результаты по отзывчивости страницы при загрузки и критическое смещение макета



Задание 4 проверить ресурс с помощью Chrome DevTools под различные экраны и мониторы. Выберите 4 различных модели, проверти в горизонтальном и вертикальном отображения экрана.

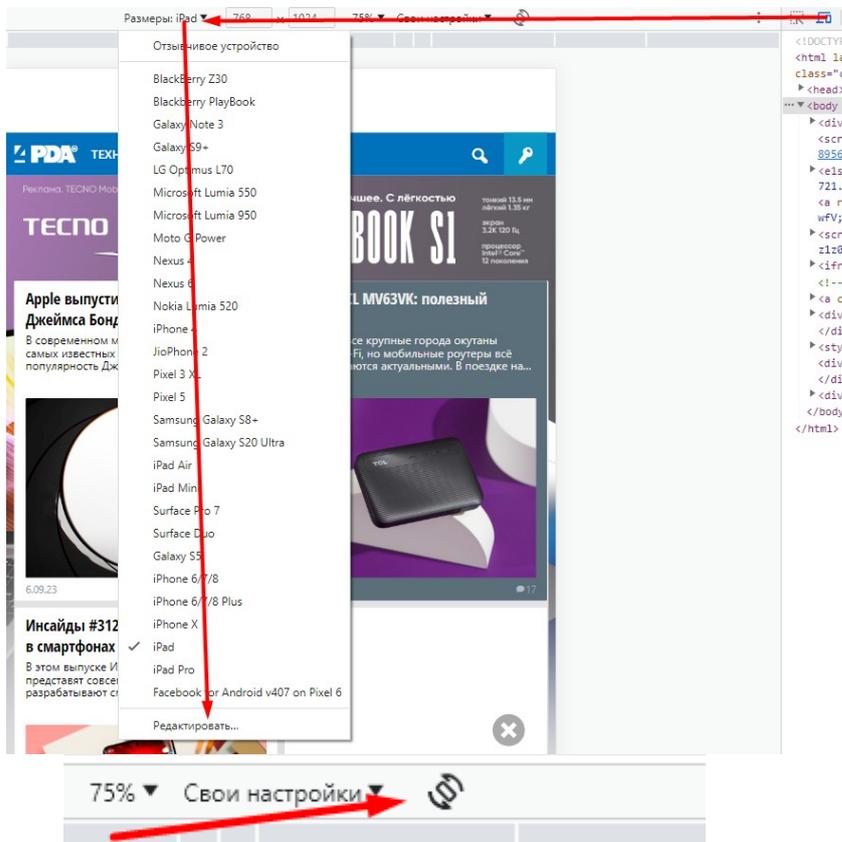
1. Для этого откройте гуглхром
2. Перейдите на любой ресурс

3. нажмите F12



4. выберите данный элемент

5. нажмите редактировать и выберите все модели в появившемся окне поставьте галочки на против нужных вам телефонов и экранов

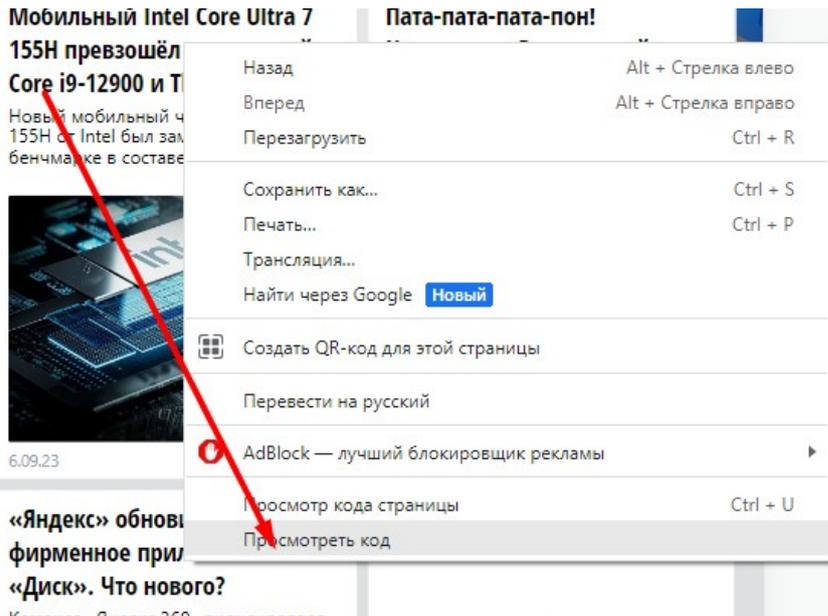


6. для поворота экрана нажмите

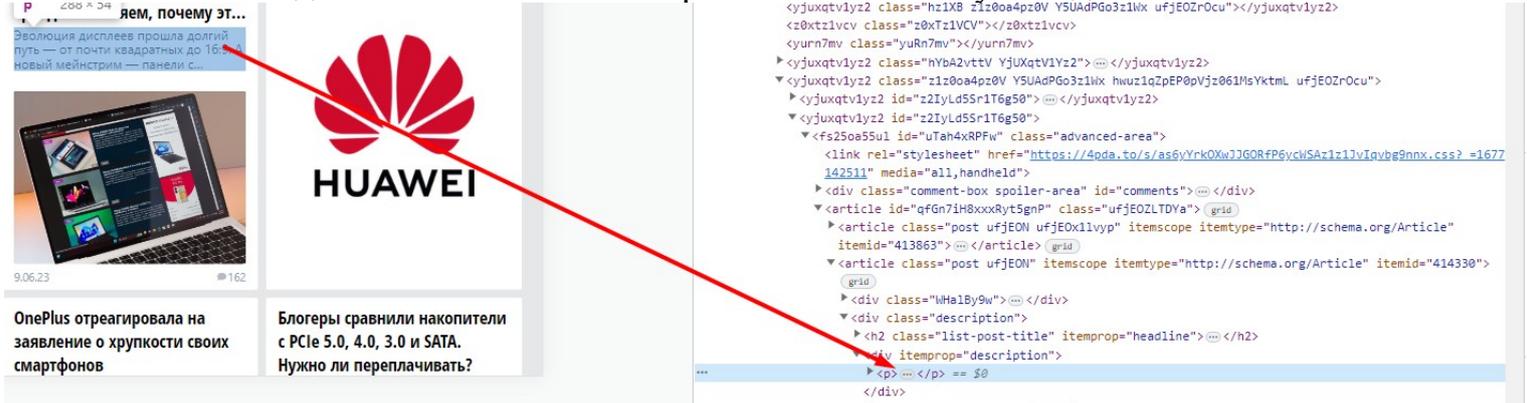
7. Если заметили на каком либо ресурсе сдвиг или неверное отображение, создайте дефект и прикрепите скриншот с описанием и разрешением экрана!

Задание 5 Изменение элементов странице в браузере для выбранного вами ресурса, до перезагрузки страницы. Измените несколько элементов отображения на выбранном ресурсе

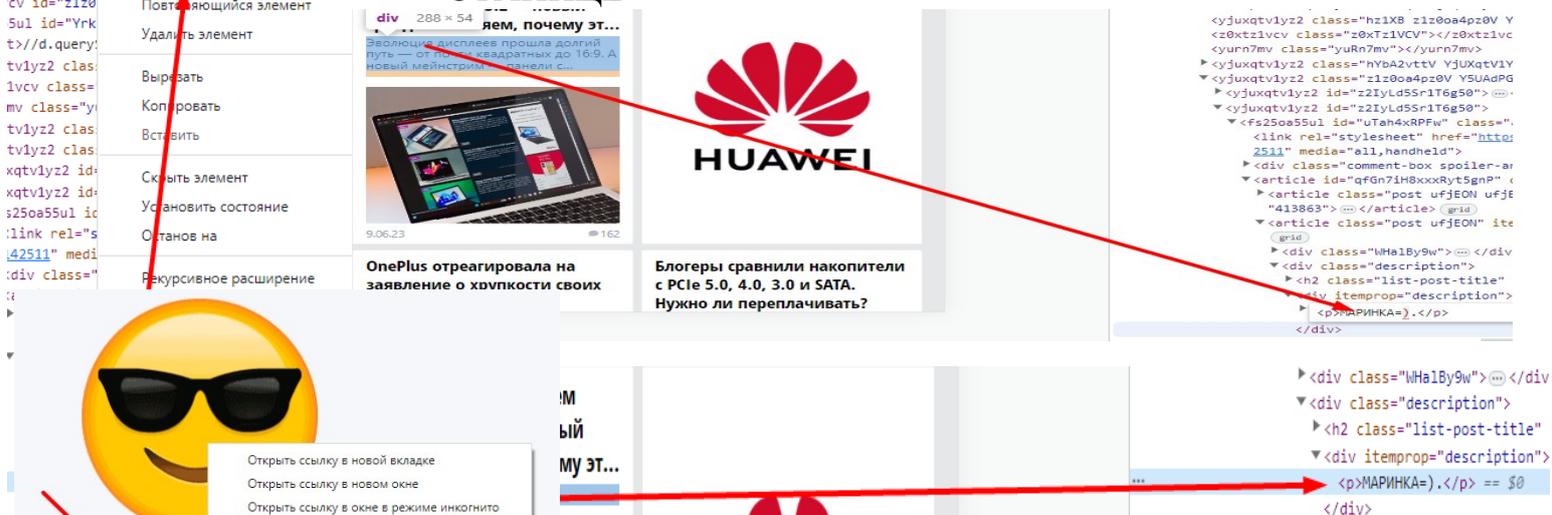
1. Для того чтобы внести изменения в элемент на странице, в рабочей области сайта нажмите правой кнопкой мыши просмотреть код



2. После вам подсвечивается область верстки ответственную за данный блок



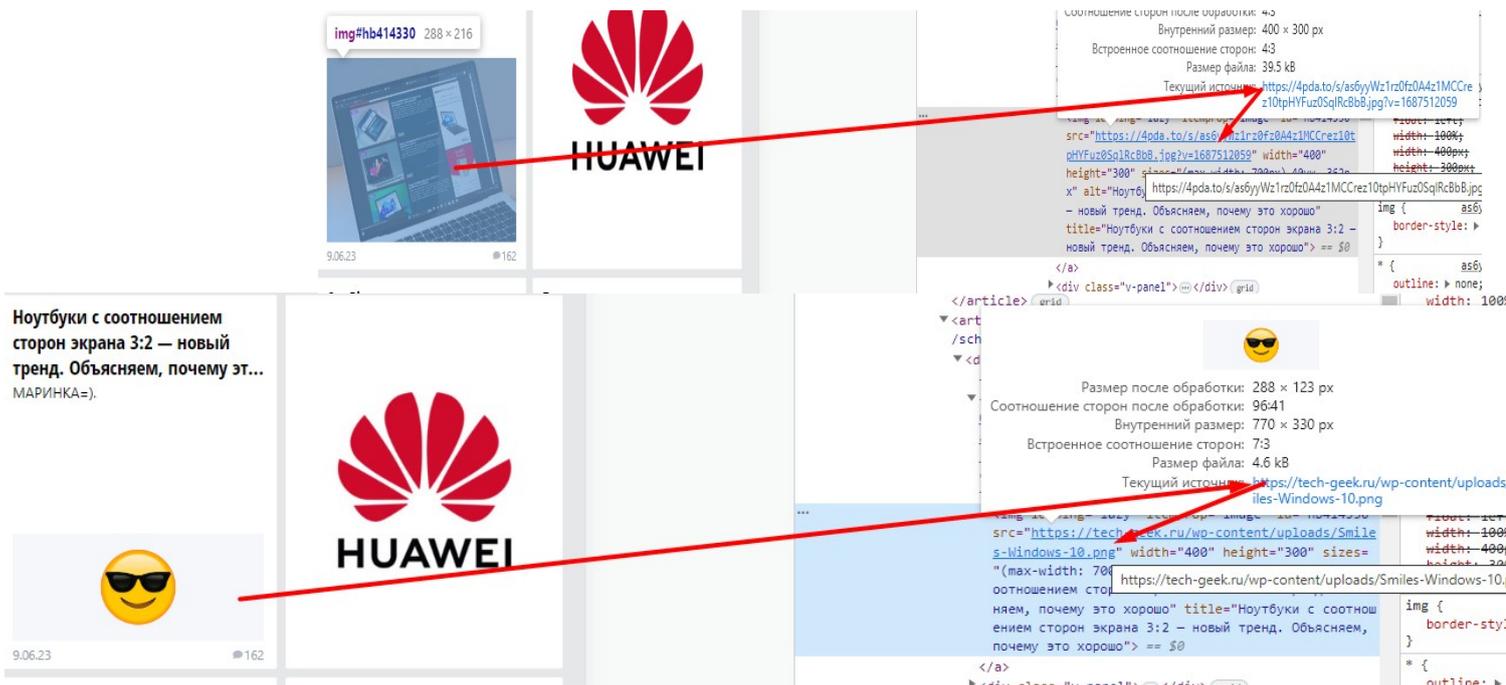
3. Для внесения правок нажмите правой кнопкой мыши редактировать, как HTML
ВПИШИТЕ ЛЮБОЙ ТЕКСТ И ОН ОТОБРАЗИТСЯ НА СТАНИЦЕ



Если хотите подставить другую картинку найдите картинку в нете , нажмите копировать URL картинки

ПУШКИНА М.С

5. После чего вернитесь на свой ресурс в панель редактора, найдите выбранный блок и вставьте скопированный вами URL

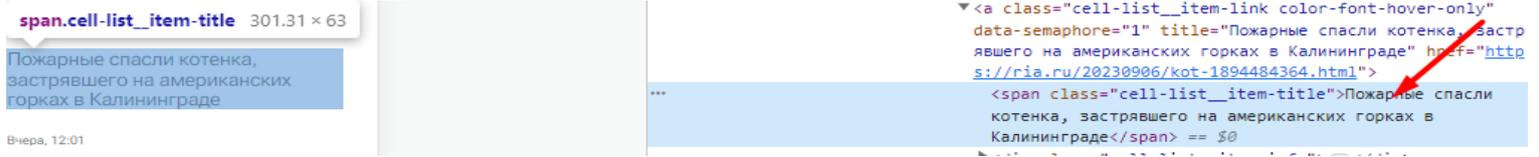


Задание 5 Проверка верстки макета на переполнение. Откройте ресурс и сделайте двум разным блокам проверку на переполнение текстом.

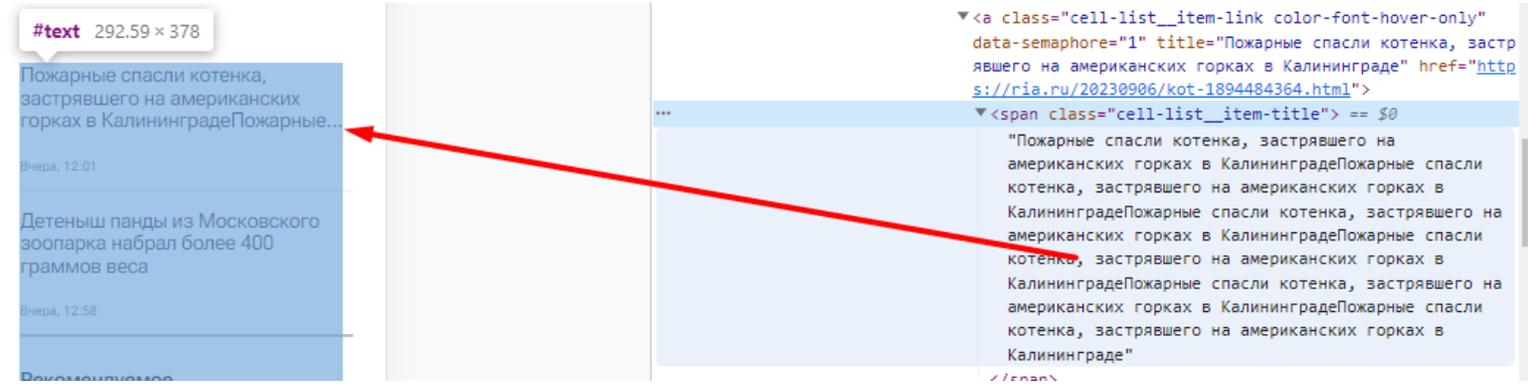
Для этого зайдите в панель редактора F12, на сайте выберите место, которое вам нужно проверить для переполнения, нажмите правой кнопкой мыши=> просмотреть код, справа найдите нужный блок с текстом и добавьте туда пару абзацев.

По результату ваш макет должен или растянуть блок по объёму текста либо, обрезать текст и заменить его...

до



после



Контрольные вопросы:

1. Как оценить скорость загрузки страницы в хrome и просмотреть ее основные критические моменты?
2. Как глянуть в хrome отображение любого ресурса под свои параметры экрана?
3. Как изменить верстку странице в браузере?

4. Как проверить ресурс на переполнение текстом ?
5. Как вызвать панель разработчиков?
6. Как проверить адаптивность сайта для двух различных экранов в iLoveadaptive?
7. Как проверить адаптивность в горизонтальном отображении для определённого экрана?

ПУШКИНА М.С

Практическая 15-16

Тема: нагрузочное тестирование ресурса. Установка, изучение основных элементов JMeter. Работа с группой потоков. слушателями запросов, контроллером Samplers, HTTP-запросом и его свойствами. Анализ результатов тестирования.

ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

Нагрузочное тестирование — это подмножество тестирования производительности, используемое для программного обеспечения, веб-сайтов, приложений и связанных с ними систем. Это нефункциональный тест, который имитирует поведение нескольких пользователей, одновременно обращающихся к системе. Также называемое «объемным тестированием», нагрузочное тестирование воспроизводит производительность, стабильность и функциональность веб-системы в реальных условиях, поэтому оно является одним из последних и наиболее важных видов тестирования, проводимых перед развертыванием.

Нагрузочное тестирование выявляет несколько критических аспектов веб-системы, включая следующие:

- Общая операционная мощность приложения, включая количество одновременно работающих пользователей, которых можно поддерживать
- Способность приложения реагировать на пиковые нагрузки пользователей
- Стабильность инфраструктуры приложения
- Время отклика приложения, пропускная способность и потребность в ресурсах при различных уровнях пользовательской нагрузки

Нагрузочное тестирование — это важный процесс, используемый перед запуском любого клиент-серверного интернет и интранет приложения. Это относится как к внешнему программному обеспечению, например, веб-сайту, так и к внутренним системам, например, серверам, на которых размещен сайт.

Метрики нагрузочного тестирования

Метрики выбираются, исходя из особенностей проекта. Они будут нужны в тех или иных тест-кейсах. Некоторые самые важные метрики:

1. **Среднее время ответа:** усредненный период времени, нужный системе на выполнение запроса, сгенерированного клиентом/пользователем. Метрика показывает некую обобщенную скорость реакции системы.
2. **Количество ошибок:** показатель, обозначающий процент запросов, завершенных с ошибкой, по отношению к общему количеству запросов. Ошибки возникают, если приложение становится не способно обработать запрос за нужное время из-за технических проблем. Большое количество (накопленных) ошибок делает невозможной нормальную работу с приложением.
3. **Пропускная способность:** метрика, оценивающая пропускную способность системы, то есть ее способность «пропустить через себя» и обработать нужное (заданное в требованиях к продукту) количество данных. Выражается в единицах информации за единицу времени — килобайтах/мегабайтах в секунду.
4. **Количество запросов в секунду:** сколько одновременных запросов в секунду приложение способно принять и обработать (прим. под «приложением» здесь имеется в виду *сервер приложений*). Запросы могут быть разноплановыми — как обычные текстовые, так и более «требовательные» — рисунки и видео, или большие сложные документы.
5. **Количество одновременных пользователей:** количество пользователей, одновременно присутствующих в системе в какой-то момент времени. Отличается тем, что учитываются просто пользователи, активные в системе в момент времени, но не делающие каких-либо запросов, не выполняющие действия.
6. **Максимальное время ответа:** измеряет максимальное зафиксированное время, которое уходит на обработку запроса; показывает длительность *цикла обработки запроса*; помогает определить «узкие места» в системе.

Когда необходимо:

- нужна оценка текущего лимита производительности проекта (например, перед какими-то промо-акциями, рассылками, “черной пятницей” и т д) — чтобы оценить объём временного наращивания мощности и изменения в конфигурации проекта на период акции;
- вы ищете пути повышения производительности текущей архитектуры/инфраструктуры (поиск проблемных мест) на постоянной основе;

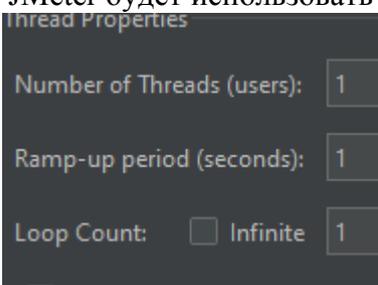
- нужно проверить новый проект перед запуском в продакшн или новую инфраструктуру после переезда.

На базовом уровне **JMeter** работает, имитируя посетителей вашего приложения или сервиса, позволяя пользователям создавать и отправлять запросы http (Hypertext Transfer Protocol) на сервер. Затем собираются данные ответа сервера, а статистические данные отображаются визуально для пользователей в виде диаграмм и отчетов. Из данных, собранных и созданных отчетов, пользователи могут наблюдать за поведением приложений и узкими местами производительности, предлагая представление о том, где необходимо улучшить приложения. Важно отметить, что JMeter не является браузером, то есть он работает только на уровне протокола и не поддерживает все действия, выполняемые клиентом или пользователем, такие как рендеринг JavaScript.

Группа потоков



Элементы группы потоков — это начальные точки любого плана тестирования. Все контроллеры и семплы должны находиться в группе потоков. Другие элементы, например прослушиватели, могут быть размещены непосредственно под планом тестирования, и в этом случае они будут применяться ко всем группам потоков. Как следует из названия, элемент группы потоков управляет количеством потоков, которые JMeter будет использовать для выполнения вашего теста.



Элементы управления группой потоков позволяют:

- **Установите количество потоков**
- **Установите период нарастания**
- **Установите количество раз для выполнения теста**

Каждый поток будет выполнять план тестирования полностью и совершенно независимо от других потоков тестирования. Несколько потоков используются для имитации одновременных подключений к вашему серверному приложению.

Период нарастания сообщает JMeter, сколько времени потребуется, чтобы «набрать» полное количество выбранных потоков. Если используется 10 потоков и период нарастания составляет 100 секунд, то JMeter потребуется 100 секунд, чтобы запустить все 10 потоков. Каждый поток запускается через 10 (100/10) секунд после начала предыдущего потока. Если имеется 30 потоков и период нарастания 120 секунд, то каждый последующий поток будет задерживаться на 4 секунды.

Наращение должно быть достаточно продолжительным, чтобы избежать слишком большой рабочей нагрузки в начале теста, и достаточно коротким, чтобы последние потоки начали выполняться до завершения первых (если только этого не хотят).

Начните с увеличения или уменьшения количества потоков и при необходимости увеличивайте или уменьшайте их.

По умолчанию группа потоков настроена на один цикл обработки своих элементов.

Группа потоков также позволяет указать **время жизни потока**. Установите флажок в нижней части панели «Группа потоков», чтобы включить/отключить дополнительные поля, в которых вы можете ввести продолжительность теста и задержку запуска. Вы можете настроить **Продолжительность (секунды)** и **Задержку запуска (секунды)**, чтобы контролировать продолжительность каждого потока группы и через сколько секунд она начнется. Когда тест запускается, JMeter будет ждать **задержки запуска (в секундах)** перед запуском потоков группы потоков и работать в течение настроенного времени **продолжительности (в секундах)**.



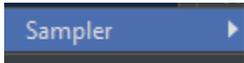
Контроллеры

JMeter имеет два типа контроллеров:

- 1) семплы
- 2) логические контроллеры.

Они управляют обработкой теста.

Сэмплы сообщают JMeter отправлять запросы на сервер. Например, добавьте образец HTTP-запроса, если вы хотите, чтобы JMeter отправлял HTTP-запрос. Вы также можете настроить запрос, добавив один или несколько элементов конфигурации в сэмплер.



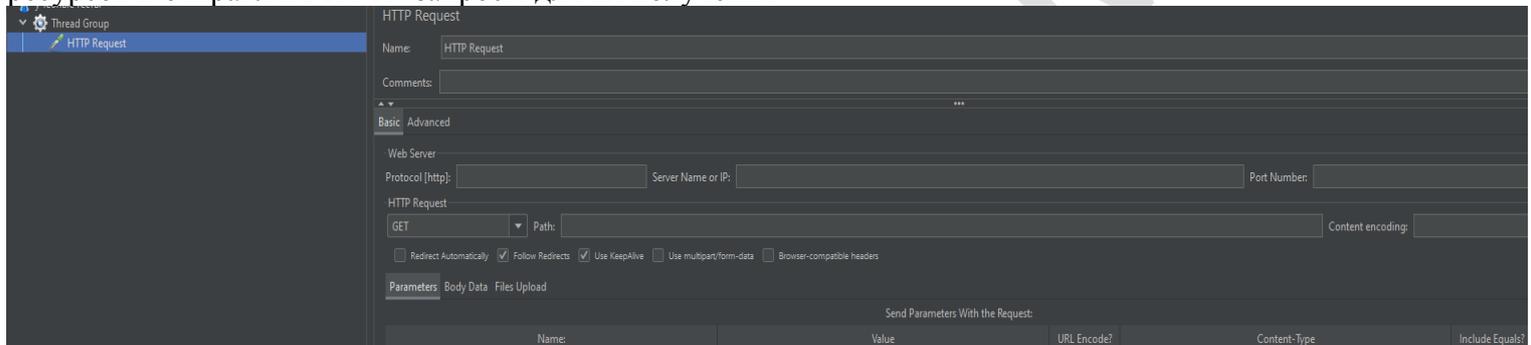
Сэмплеры сообщают JMeter отправлять запросы на сервер и ждать ответа. Они обрабатываются в том порядке, в котором они появляются в дереве. Контроллеры можно использовать для изменения количества повторений сэмплера.

Сэмплеры JMeter включают в себя:

- FTP-запрос
- **HTTP-запрос** (также может использоваться для веб-сервиса SOAP или REST)
- JDBC-запрос
- Запрос объекта Java
- JMS-запрос
- Запрос теста JUnit
- LDAP-запрос
- Почтовый запрос
- Запрос процесса ОС
- TCP-запрос

HTTP-запрос

Этот сэмплер позволяет отправлять HTTP/HTTPS-запрос на веб-сервер. Он также позволяет вам контролировать, будет ли JMeter анализировать HTML-файлы на наличие изображений и других встроенных ресурсов и отправлять HTTP-запросы для их получения



Каждый сэмплер имеет несколько свойств, которые вы можете установить. Вы можете дополнительно настроить пробоотборник, добавив в план тестирования один или несколько элементов конфигурации.

Если вы собираетесь отправлять несколько запросов одного типа (например, HTTP-запрос) на один и тот же сервер, рассмотрите возможность использования элемента конфигурации по умолчанию. Каждый контроллер имеет один или несколько элементов Defaults

listeners

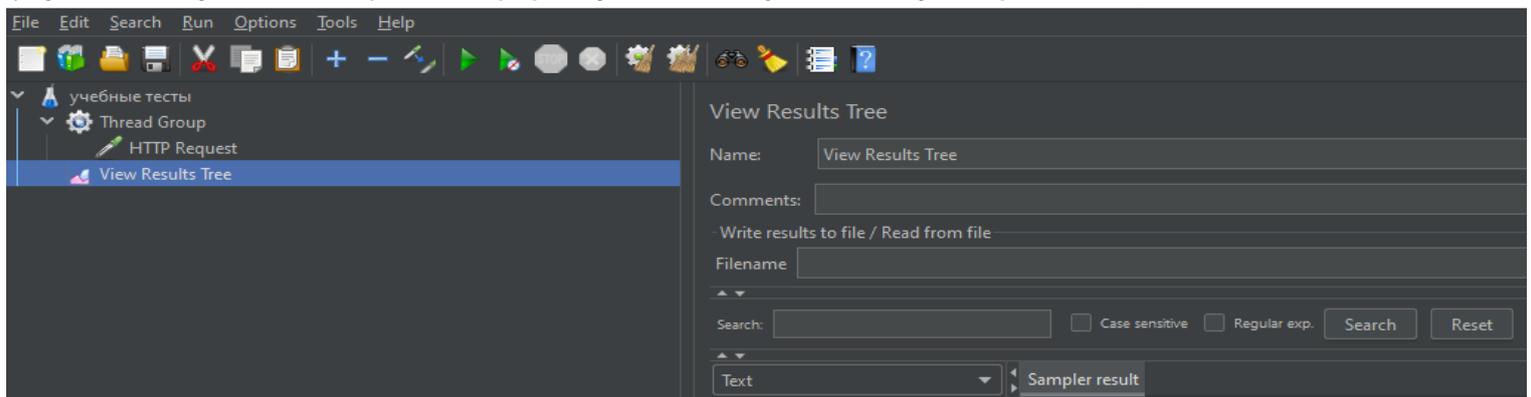


Большинство **слушателей** помимо «прослушивания» результатов теста выполняют несколько ролей. Они также предоставляют средства для просмотра, сохранения и чтения сохраненных результатов испытаний.

Обратите внимание, что прослушиватели обрабатываются в конце области, в которой они находятся.

Сохранение и считывание результатов испытаний являются общими. Различные прослушиватели имеют панель, с помощью которой можно указать файл, в который будут записываться (или считываться) результаты. По умолчанию результаты сохраняются в виде XML-файлов, обычно с расширением «.jtl». Хранение в формате CSV является наиболее эффективным вариантом, но он менее детализирован, чем XML (другой доступный вариант).

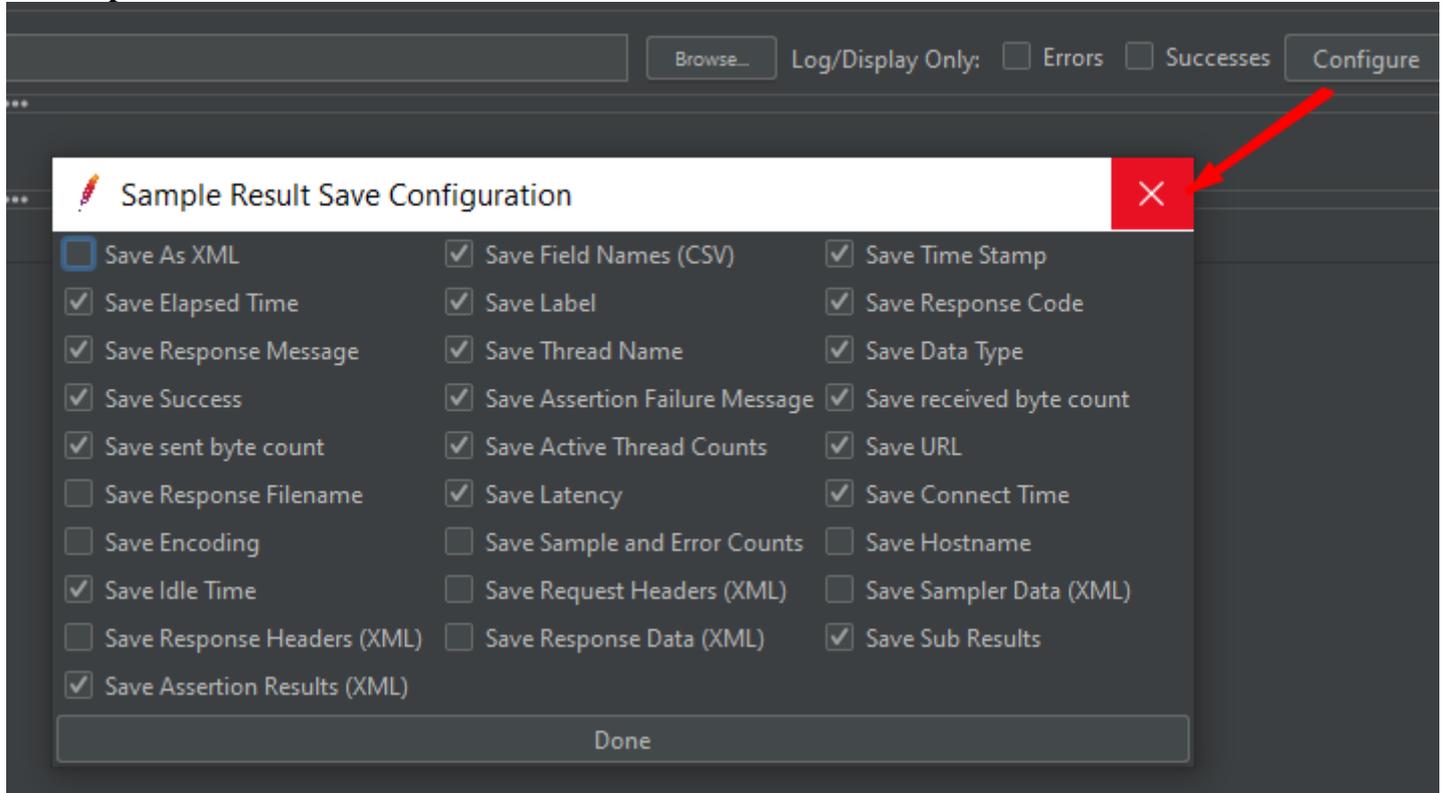
учебные тесты.jmx (C:\Users\Марина\Desktop\apache-jmeter-5.6.2\bin\учебные тесты.jmx) - Apache JMeter (5.6.2)



конфигурации сохранения результатов

Graph Results

Прослушиватели можно настроить на сохранение различных элементов в файлы журнала результатов (JTL) с помощью всплывающего окна «Конфигурация», как показано ниже. Значения по умолчанию определяются, как описано в документации [по конфигурации прослушивателя по умолчанию](#). Элементы с (CSV) после имени применяются только к формату CSV; элементы с (XML) применяются только к формату XML. Формат CSV в настоящее время нельзя использовать для сохранения элементов, содержащих разрывы строк. Обратите внимание, что файлы cookie, метод и строка запроса сохраняются как часть опции « **Данные семплера** ».

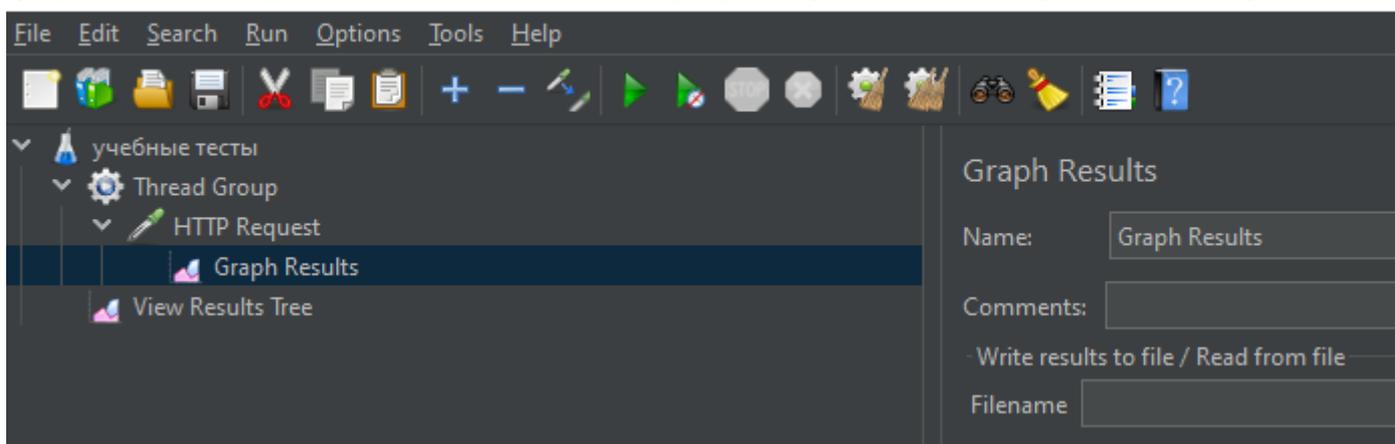


Результаты графика

Прослушиватель результатов графика создает простой график, на котором отображаются все времена расчета. Внизу графика в миллисекундах отображаются текущая выборка (черный), текущее среднее значение всех выборок (синий), текущее стандартное отклонение (красный) и текущая пропускная способность (зеленый).

Число пропускной способности представляет собой фактическое количество запросов в минуту, обработанных сервером. Этот расчет включает в себя любые задержки, добавленные вами к тесту, и собственное время внутренней обработки JMeter. Преимущество такого расчета заключается в том, что это число представляет собой нечто реальное — ваш сервер фактически обрабатывает такое количество запросов в минуту, и вы можете увеличить количество потоков и/или уменьшить задержки, чтобы определить максимальную пропускную способность вашего сервера. А если бы вы произвели расчеты, в которых были бы учтены задержки и обработка JMeter, было бы неясно, какой вывод вы могли бы сделать из этого числа.

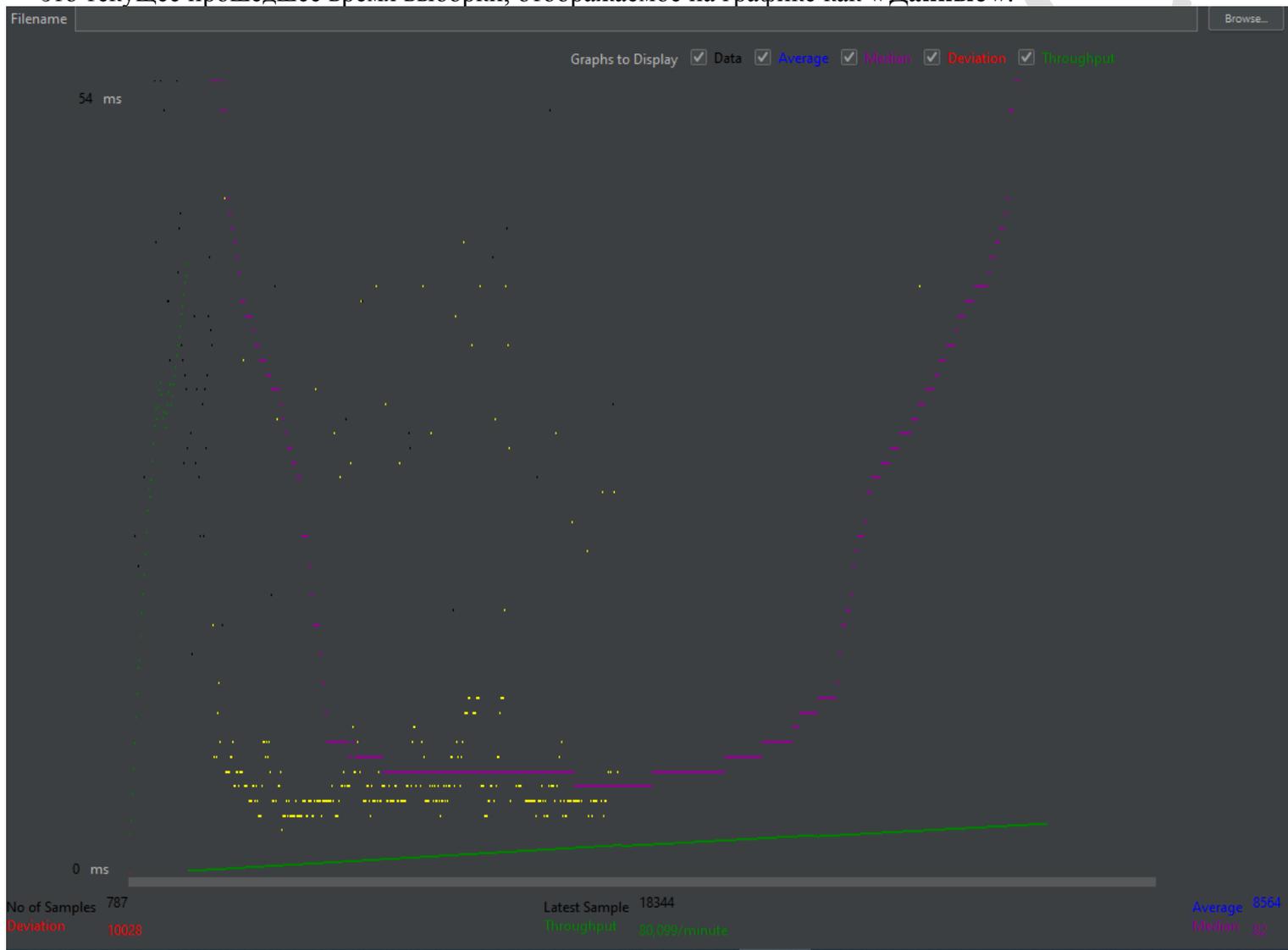
учебные тесты.jmx (C:\Users\Марина\Desktop\apache-jmeter-5.6.2\bin\учебные тесты.jmx) - Apache



В следующей таблице кратко описаны элементы на графике. Более подробную информацию о точном значении статистических терминов можно найти в Интернете, например, в Википедии, или в книге по статистике.

- **Data** **Данные** — постройте фактические значения данных.
- **Average** **Среднее** - постройте среднее значение
- **Median** **Медиана** — постройте медиану (среднее значение)
- **Deviation** **Отклонение** — постройте стандартное отклонение (мера отклонения).
- **Throughput** **Пропускная способность** — постройте количество выборок в единицу времени.

Отдельные цифры в нижней части дисплея представляют собой текущие значения. « **Последняя выборка** » — это текущее прошедшее время выборки, отображаемое на графике как « **Данные** ».



URL состоит из различных частей, некоторые из которых являются обязательными, а некоторые - факультативными. Рассмотрим наиболее важные части на примере:

<http://www.example.com:80/path/to/myfile.html?key1=value1&key2=value2#SomewhereInTheDocument>

http:// *это протокол*. Он отображает, какой протокол браузер должен использовать. Обычно это HTTP-протокол или его безопасная версия - HTTPS. Интернет требует эти 2 протокола, но браузеры часто могут использовать и другие протоколы, например mailto: (чтобы открыть почтовый клиент) или ftp: для запуска передачи файлов, так что не стоит удивляться, если вы вдруг увидите другие протоколы.

www.example.com *это доменное имя*. Оно означает, какой веб-сервер должен быть запрошен. В качестве альтернативы может быть использован и [IP-адрес](#), но это делается редко, поскольку запоминать IP сложнее, и это не популярно в интернете.

:80 это порт. Он отображает технический параметр, используемый для доступа к ресурсам на веб-сервере. Обычно подразумевается, что веб-сервер использует стандартные порты HTTP-протокола (80 для HTTP и 443 для HTTPS) для доступа к своим ресурсам. В любом случае, порт - это факультативная составная часть URL.

/path/to/myfile.html ?key1=value1

→ *Path to resource*

/path/to/myfile.html это адрес ресурса на веб-сервере. В прошлом, адрес отображал местоположение реального файла в реальной директории на веб-сервере. В наши дни это чаще всего абстракция, позволяющая обрабатывать адреса и отображать тот или иной контент из баз данных.

?key1=value1&key2=value2#Some

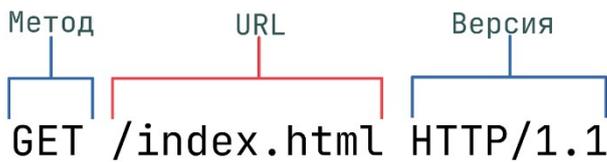
→ *Parameters*

?key1=value1&key2=value2 это дополнительные параметры, которые браузер сообщает веб-серверу. Эти параметры - список пар ключ/значение, которые разделены символом &. Веб-сервер может использовать эти параметры для исполнения дополнительных команд перед тем как отдать ресурс. Каждый веб-сервер имеет свои собственные правила обработки этих параметров и узнать их можно, только спросив владельца сервера.

Стартовая строка HTTP-запроса состоит из трех элементов:

1. *Метод HTTP-запроса* (method, реже используется термин verb). Обычно это короткое слово на английском, которое указывает, что конкретно нужно сделать с запрашиваемым ресурсом.

Например, метод GET сообщает серверу, что пользователь хочет получить некоторые данные, а POST — что



некоторые данные должны быть помещены на сервер.

2. Цель запроса. Представлена указателем ресурса URL, который состоит из протокола, доменного имени (или IP-адреса), пути к конкретному ресурсу на сервере. Дополнительно может содержать указание порта, несколько параметров HTTP-запроса и еще ряд опциональных элементов.

3. Версия используемого протокола (либо HTTP/1.1, либо HTTP/2), которая определяет структуру следующих за стартовой строкой данных.

В примере ниже стартовая строка указывает, что в качестве метода используется GET, обращение будет произведено к ресурсу /index.html, по версии протокола HTTP/1.1:

ХОД РАБОТЫ

Задание 1 загрузить и запустить

1. для установки JMeter перейдите по ссылке и скачайте архив https://jmeter.apache.org/download_jmeter.cgi

Component Reference
• Functions Reference
• Properties Reference
• Change History
• Javadocs
• JMeter Wiki
• FAQ (Wiki)

Tutorials
• Distributed Testing
• Recording Tests
• JUnit Sampler
• Access Log Sampler
• Extending JMeter

For more information concerning Apache JMeter, see the [Apache JMeter](#) site.

KEYS

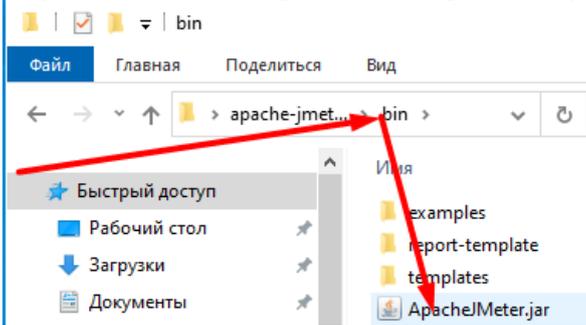
Apache JMeter 5.6.2 (Requires Java 8+)

Binaries

[apache-jmeter-5.6.2.tgz sha512 pgp](#)
[apache-jmeter-5.6.2.zip sha512 pgp](#)

2. разархивируйте

3. перейдите в папку bin и запустите файл



1.1 Требования

JMeter требует, чтобы ваша вычислительная среда соответствовала некоторым минимальным требованиям.

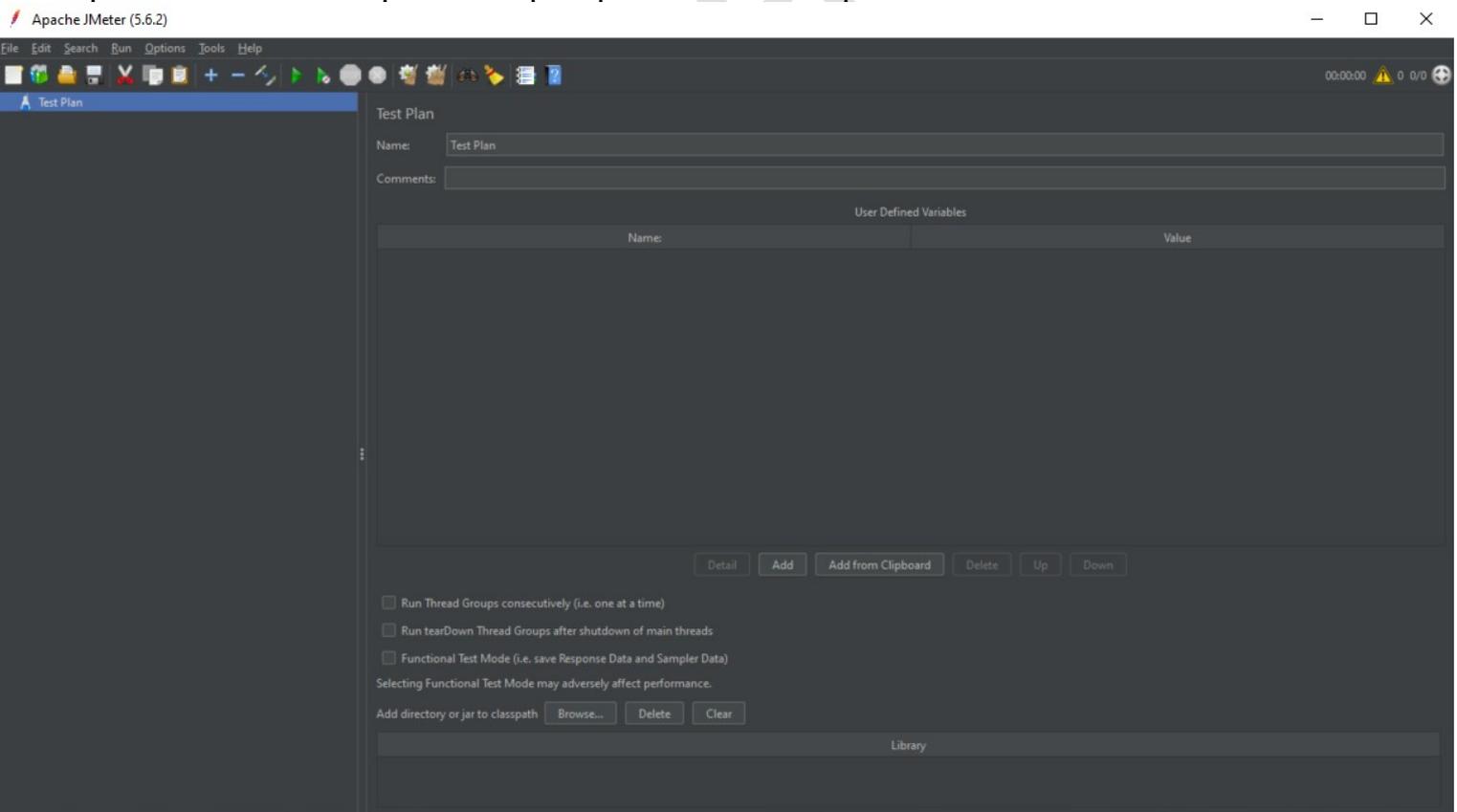
1.1.1 Версия Java

JMeter совместим с Java 8 или выше. Мы настоятельно рекомендуем вам установить последнюю дополнительную версию вашей основной версии из соображений безопасности и производительности.

Поскольку JMeter использует только стандартные API Java, пожалуйста, не отправляйте отчеты об ошибках, если ваша JRE не запускается JMeter из-за проблем с реализацией JRE.

Загрузите, если у вас нет java [Download Java for Windows](#)

4. перед вами появится рабочее пространство джиметр



задание 2 Выполнить нагрузочное тестирование для сайта Азон

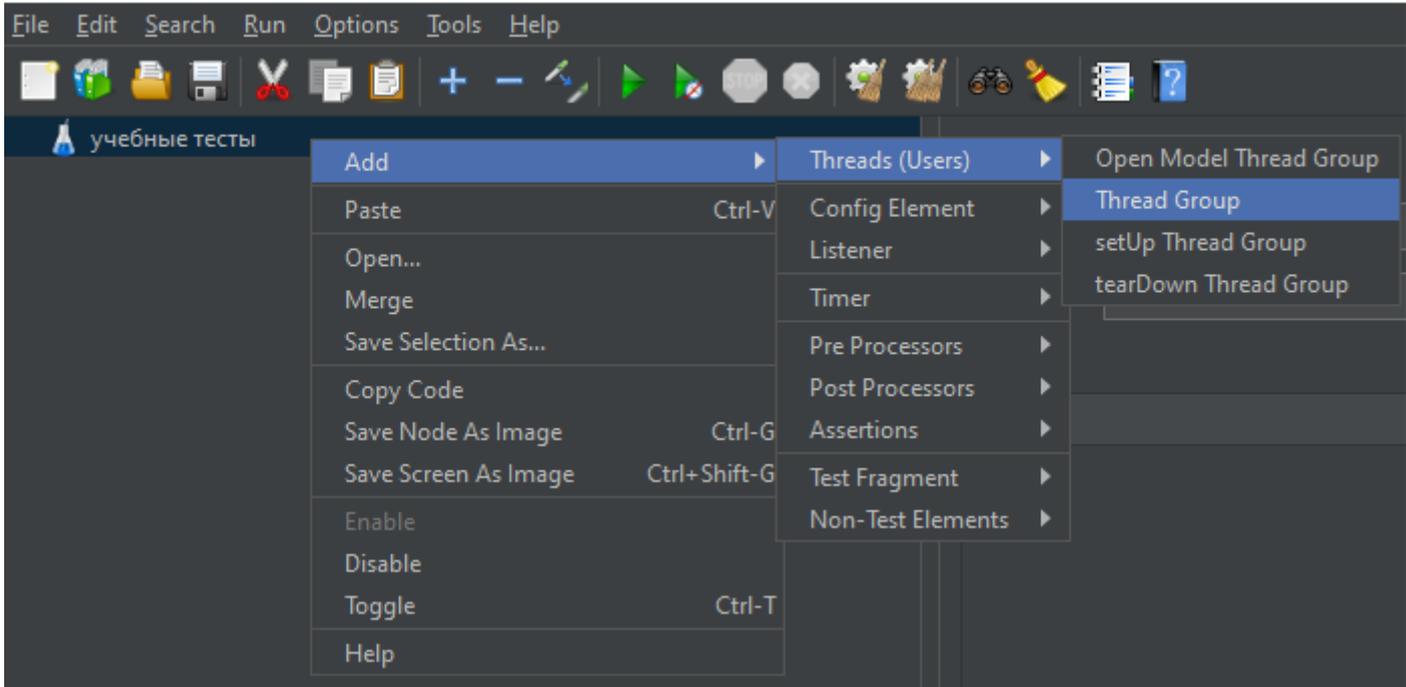
не подавать большое количество запросов, то ваш ip-шник просто заблочат за дедос-атаку! И вы больше не сможете зайти на данный ресурс!

1. Перейдите на сайт Азон в раздел одежда обувь, выберите любой товар и скопируйте его URL

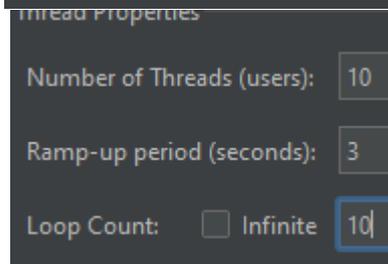
2. Перейдите в JMeter и переименуйте тестовый план в Учебные тесты В левой области будут отображаться все ваши тесты, вы их можете изменять, удалять и редактировать, запускать, останавливать, сохранять



3. После чего нажмите правой кнопкой мыши на ваш Учебный тест и выберите учебные тесты.jmx (C:\Users\Марина\Desktop\apache-jmeter-5.6.2\bin\учебные тесты.jmx) - Apache

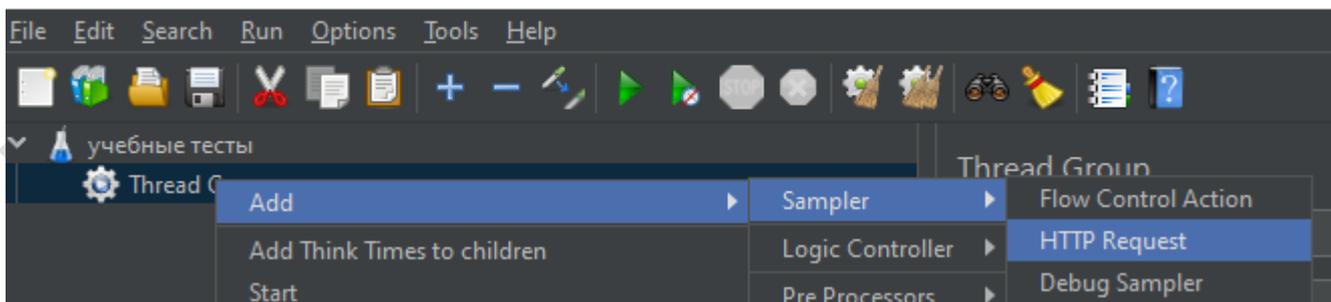


5. Установите количество потоков 10(пользователей), период нарастания 3 (в течении какого времени эти пользователи будут заходить на сайт)и количество раз для выполнения 10 раз. Не задавайте большие значения ибо ресурс вас заблокирует, за дедос атаку



6. Нажмите правой кнопкой мыши на Thread Group и выберите

учебные тесты.jmx (C:\Users\Марина\Desktop\apache-jmeter-5.6.2\bin\учебные тесты.jmx) - Apache



7. Создаем HTTP-запрос

7.1 с азна копируем нашу ссылку на товар

https://www.ozon.ru/product/yubka-shorty-mikmn-1041163834/?advert=nRaDj-vkm45AovH15QexakOLrmtFBmn-iOWbra0oi_dfnhlml70LIakXcv_bp9tts1sug-cyJB2ke07jBO5S1yZXVie38bGuz3mxNO-3ihmH-vMqCFGogG_cQlj7YDUuxJbMz4JqM1tqpGxrdYxisbC6leZiXE356B2AuCeZtvK0uWkCiRjTXLtg1M8Zy93vNbz5TYPBJiX5W3uQAsLo8K6KS2WJGM8bCvVdPLsEgtLZ8ZjQ6MJyELA&avtc=1&avte=2&avts=1694094671&sh=40v1o1sjjw

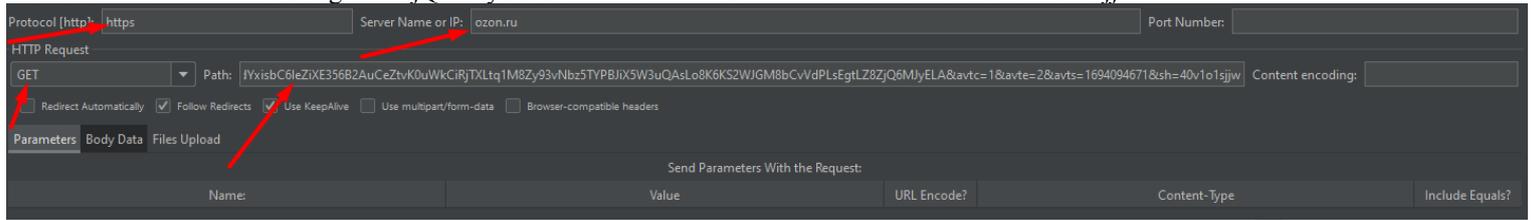
7.2 данную ссылку делим на 3 части:

7.2.1 протокол https

7.2.3 доменное имя ozon.ru

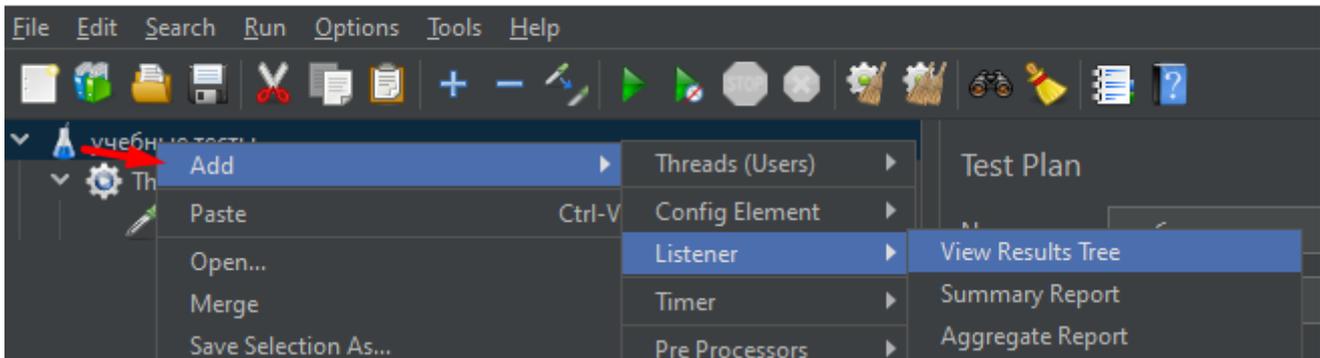
7.2.3 путь

/product/yubka-shorty-mikmn-1041163834/?advert=nRaDj-vkm45AovH15QexakOLrmtFBmn-iOWbra0oi_dfnhlml70LlAkXcv_bp9tts1sug-cyJB2ke07jBO5S1yZXVie38bGuz3mxNO-3ihmH-vMqCFGogG_cQlj7YDUuxJbMz4JqM1tqpGxrdYxisbC6leZiXE356B2AuCeZtvK0uWkCiRjTXLtg1M8Zy93vNbz5TYPBJiX5W3uQAsLo8K6KS2WJGM8bCvVdPLsEgtLZ8ZjQ6MJyELA&avtc=1&avte=2&avts=1694094671&sh=40v1o1sjiw



8. Добавим слушателя наших запросов. Для этого нажмите правой кнопкой мыши на учебные тесты и выберите

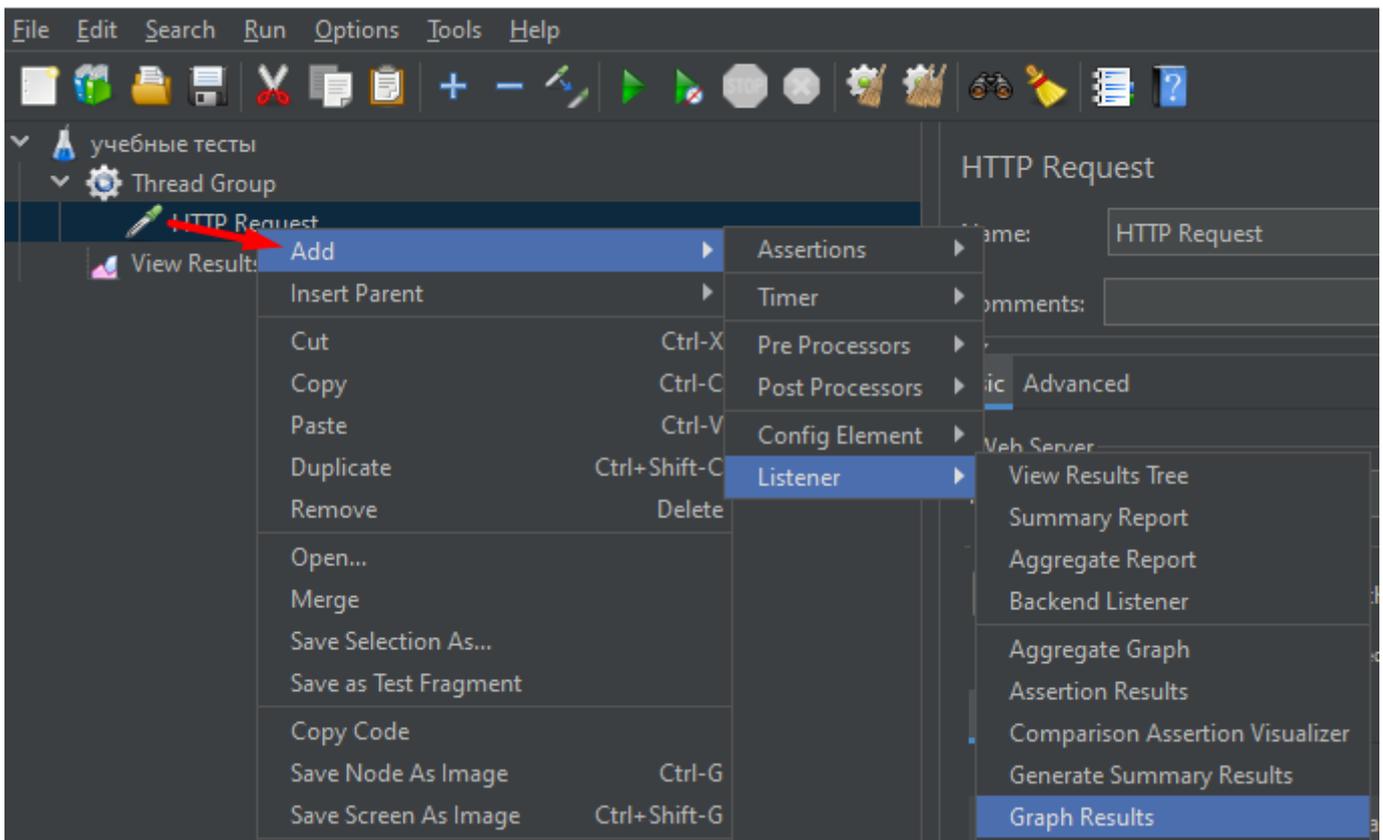
учебные тесты.jmx (C:\Users\Марина\Desktop\apache-jmeter-5.6.2\bin\учебные тесты.jmx) -

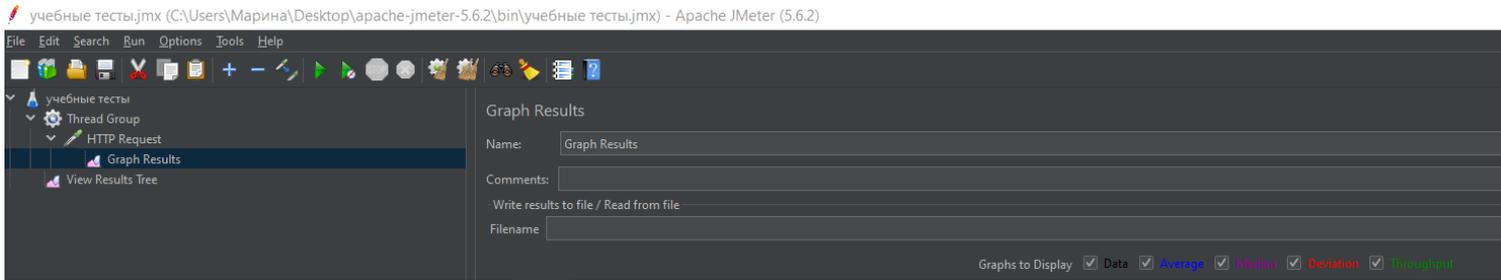


Здесь мы сможем наблюдать за результатами наших запросов на сайт, как они проходят успешно или нет

9. После чего добавим графические результаты. Для этого нажимаем правой кнопкой мыши на HTTP-запрос и выбираем график

учебные тесты.jmx (C:\Users\Марина\Desktop\apache-jmeter-5.6.2\bin\учебные тесты.jmx) - Ar

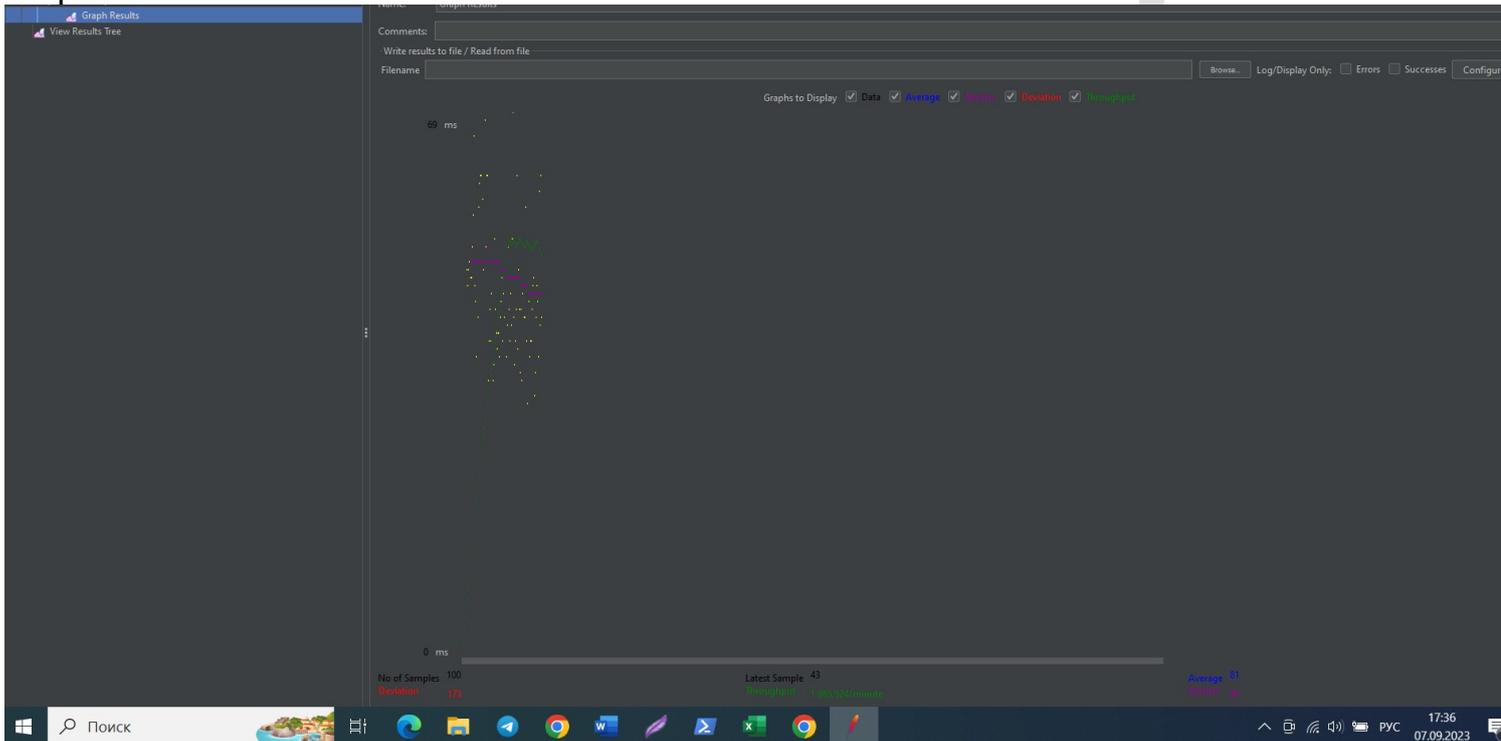




10. Запускаем наш тест нажимаем



Результат ответы смотреть на графике и на вкладке слушателя результатов дерева запросов



```
1 GET https://ozon.ru/product/yubka-shorty-mikmn-1041163834/?advert=nRaDj-vkm45Aov07jB05S1yZXVie38bGuz3mxNO-3ihmH-vMqCFGogG_cQlj7YDUuxJbMz4Jq11tapGxrdYxLo8K6KS2hJG18bCvVdPLsEgtLZ8ZjQ6HJyELA&avtc=1&avte=2&avts=1694094671&sh
```

```
2 GET data:
```

```
3
```

```
4
```

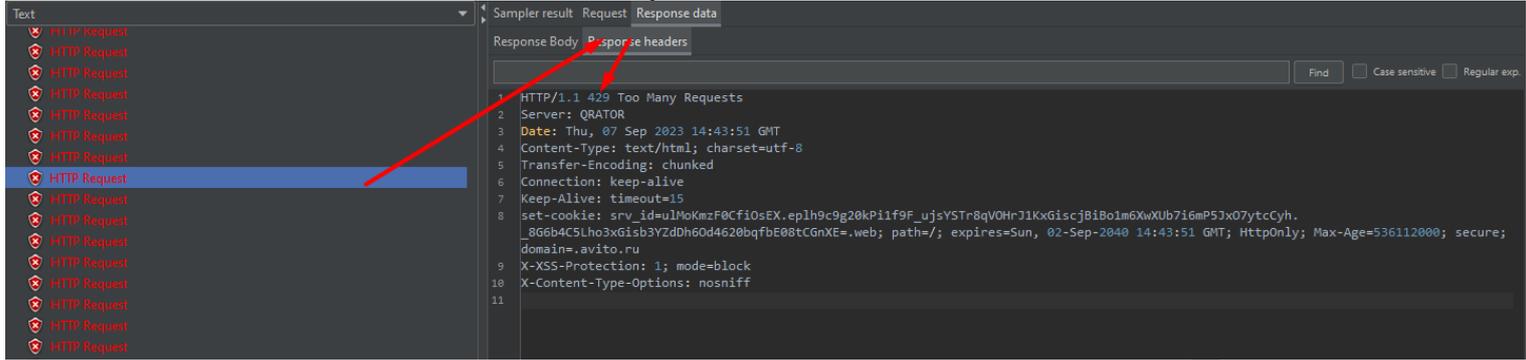
```
5
```

```
6 [no cookies]
```

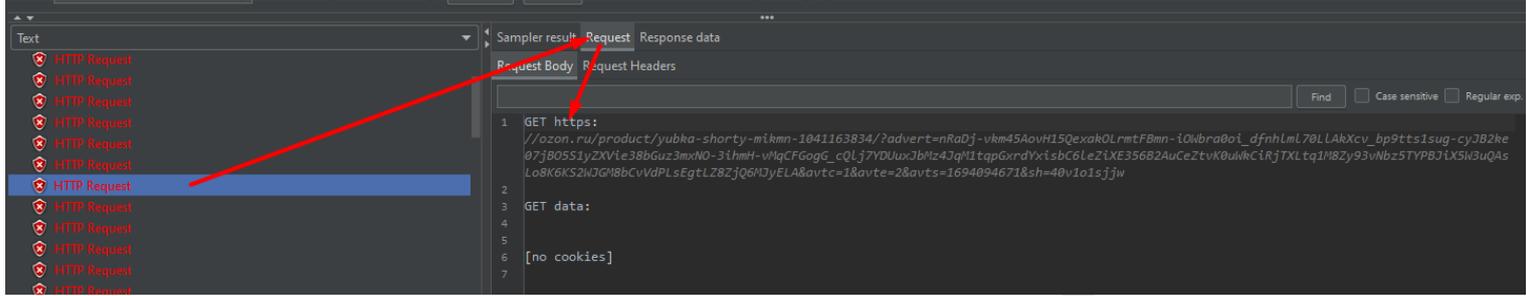
```
7
```

Анализы запросов:

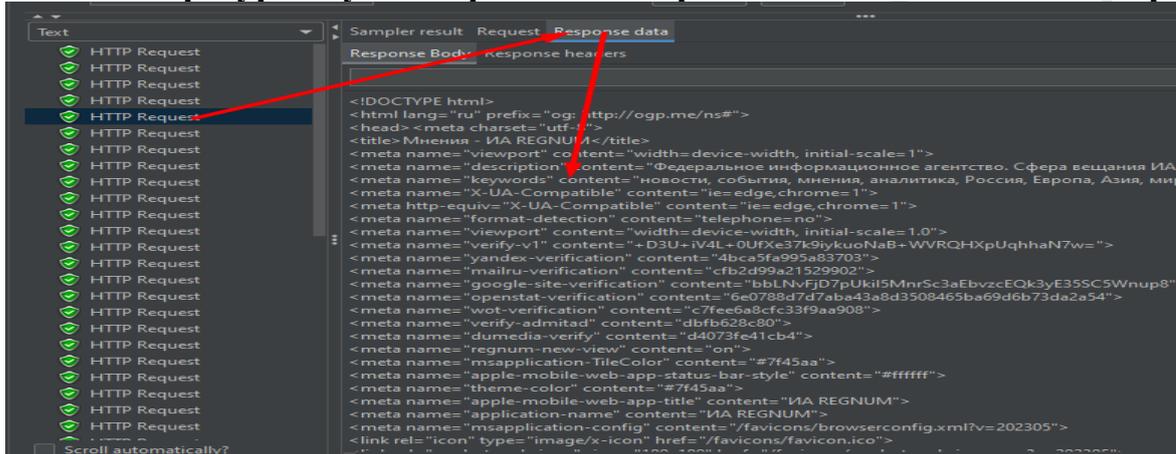
1. ответ от аzona 429 слишком много запросов



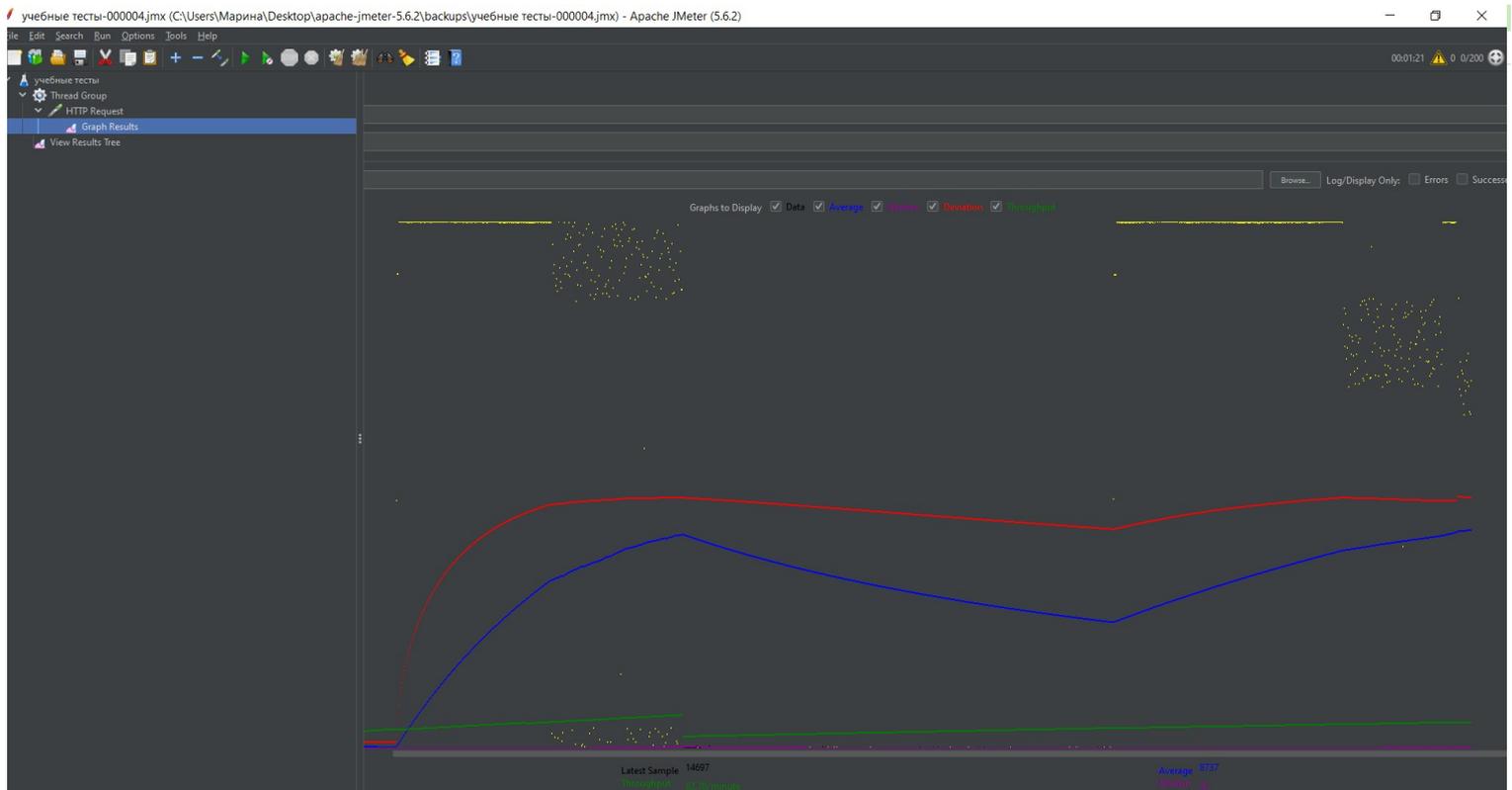
2. можем глянуть куда стреляли и на какой ресурс, здесь азон



3. ответ от ресурса куда отправляли запросы. Выводит все что на странице



4. Анализ по графику на 200 запросах в течении минуты



Задание 2 Создайте рядом еще один нагрузочный тест назовите его своим ФИО, данный нагрузочный тест должен отправлять запросы на любой вами выбранный ресурс (используйте не посещаемые сайты). Увеличьте нагрузку и проанализировать ответы. Проверьте на 200-500 пользователях, период нарастания 1-10 (в течении какого времени эти пользователи будут заходить на сайт) и количество раз для выполнения 1 раз. После чего поставьте минимальные параметры и средние. Сравните все 3 результата. Дать развернутый анализ по запросам, графику и его параметрам
 Какое количество отказов выдал сервер?

Очищаете при каждом запуске график и реквесты



Задание 2 Создайте рядом еще один нагрузочный тест назовите его Нагрузка на максималках, данный нагрузочный тест должен отправлять запросы на любой вами выбранный ресурс (возьмите яндекс или гугл серфисы или популярные магазины или соц сети, которые вам не жаль). Увеличьте нагрузку и проанализировать ответы.

Проверьте на:

- 2000-3000 пользователей
- период нарастания 10 (в течении какого времени эти пользователи будут заходить на сайт)
- количество раз для выполнения 1 раз.

Сравните все результаты с предыдущим сайтом. Дать развернутый анализ по запросам, графику и его параметрам.

Какое количество отказов выдал сервер?

Проверьте на:

- 20-50 пользователей
- период нарастания 100 (в течении какого времени эти пользователи будут заходить на сайт)
- количество раз для выполнения бесконечность, на 1 минуте остановите выполнение.

Задание 3 Рассчитать пропускную способность ресурса

Пропускная способность рассчитывается как количество запросов на единицу времени. *Время* рассчитывается от начала первой выборки до конца последней выборки. Сюда входят любые интервалы между выборками, поскольку предполагается, что они отражают нагрузку на сервер.

Формула: $\text{Пропускная способность} = (\text{количество запросов}) / (\text{общее время})$.

Контрольные вопросы:

1. Из каких частей состоит URL ссылка?
2. И чего состоит строка HTTP-запроса ?
3. Для чего используют нагрузочное тестирование? Приведите пример
4. Назовите 3 основные элемента управления группой потоков?
5. Какие два типа контроллеров имеет джиметр?
6. Что делает контроллер samplers?
7. Как вызвать и настроить параметры сохранения конфигурации резинатов
8. Где мы сможем отследить успешность выполнения запросов на сайт?

Практическая 17-18

Тема: автоматизированное нагрузочное тестирование ресурса с инструментом JMeter. Работа с группой потоков, слушателями запросов, контроллером Samplers, HTTP-запросом и его свойствами. Работа с POST запросами. Анализ ответов от сервера. Совмещение POST и GET запросов.

ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

HTTP-методы GET и POST — самые распространённые способы отправить или получить данные с сервера. Но в разных случаях оба метода могут быть небезопасными или неудобными в использовании. В этой заметке рассмотрим, какой метод когда использовать.

Как работает HTTP

GET — метод для чтения данных с сайта. Например, для доступа к указанной странице. Он говорит серверу, что клиент хочет прочитать указанный документ. На практике этот метод используется чаще всего, например, в интернет-магазинах на странице каталога. Фильтры, которые выбирает пользователь, передаются через метод GET.

-владелец ресурса: `htmlacademy.ru`

-метод: GET

-путь: `/tutorial/php/http-header`

POST — метод для отправки данных на сайт. Чаще всего с помощью метода POST передаются формы.

URL-адрес запроса: `https://htmlacademy.ru/consulting`

Метод запроса: POST

Код состояния: 200

Формат запроса

HTTP-запроса очень прост и состоит, по сути, из двух частей — заголовков и тела запроса или ответа.

Тело запроса — это информация, которую передал браузер при запросе страницы. Но тело запроса присутствует только если браузер запросил страницу методом POST. Например, если отправлена форма, то телом запроса будет содержание формы.

Запишитесь на консультацию

Мы любим отвечать на вопросы. Поэтому ждём ваших контактов, чтобы поскорее на всё ответить.

Имя *

Телефон *

Эл. почта *

Соглашаюсь на обработку персональных данных и получение информационных сообщений в соответствии с Политикой

Данные формы

source: intensive
tag: htmlcss#31
name: Test
phone: +79001234567
email: test@test.ru
agreement: 1

Форма на сайте

Данные в POST-запросе

Пример GET-запроса. Информация передаётся прямо в заголовке.

GET /blog/?
name1=value1&name2=value2
HTTP/1.1

Host: htmlacademy.ru

Пример POST-запроса. Информация передаётся в теле запроса:

POST /blog/ HTTP/1.1

Host: htmlacademy.ru

name1=value1&name2=value2

Ещё раз коротко

GET

- Фильтры в интернет-магазинах
- Передача параметров через ссылку
- Другие безопасные запросы

POST

- Любые формы с паролями или банковскими картами
- Формы заявок с персональными данными
- Отправка файлов

В каждый браузер встроены инструменты разработчика — они позволяют быстро отловить и исправить ошибки в разметке или в коде. С их помощью можно узнать, как построилось DOM-дерево, какие теги и атрибуты есть на странице, почему не подгрузились шрифты и многое другое.

В этом цикле статей мы разберём базовые возможности Chrome DevTools, но их будет вполне достаточно для начала обучения.

Как открыть инструменты разработчика

macOS — Ctrl+Cmd+I

Windows — F12

Linux — Ctrl+Shift+I

ХОД РАБОТЫ

1. Запустить джиметр

Задание 1 создать в одной ветке нагрузочный тест для сайта <https://hobbygames.ru/nastolnie> используя GET метод и POST метод. Результат мы могли слушать запросы GET и POST запросы:

1.1 Тестовый вариант 1

- количество пользователей 5-10

- период нарастания, в который пользователи будут заходить на сайт 50

- количество раз бесконечно

1.2 Тестовый вариант 2

- количество пользователей 50-100

- период нарастания, в который пользователи будут заходить на сайт 3

- количество раз бесконечно

1.3 Тестовый вариант 3

- количество пользователей 100

- период нарастания, в который пользователи будут заходить на сайт 3

- количество раз 2000

1. создайте нагрузочный тест с GET методом, как в прошлой практической для любого ресурса, (без сильной нагрузки на ресурс в виде пользователей и количества обращений)

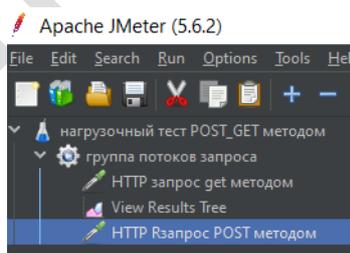
2. создайте в этом же тесте нагрузочный тест с методом POST

Для создания нагрузочного теста с методом POST, вам понадобится:

1 шаг

1 элемент группы потоков **Thread Group** (который у вас был создан при создании нагрузочного теста с методом GET)

1 элемент HTTP-запроса **HTTP Request**

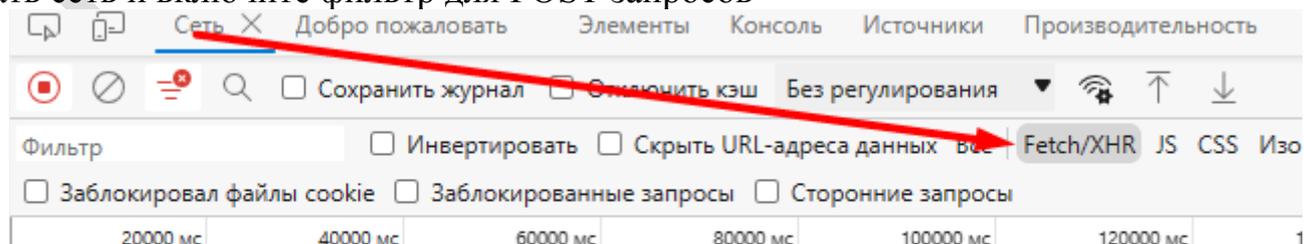


Должна получиться такая структура

Шаг 2

1. Перейти на сайт, открыть панель разработчика F12

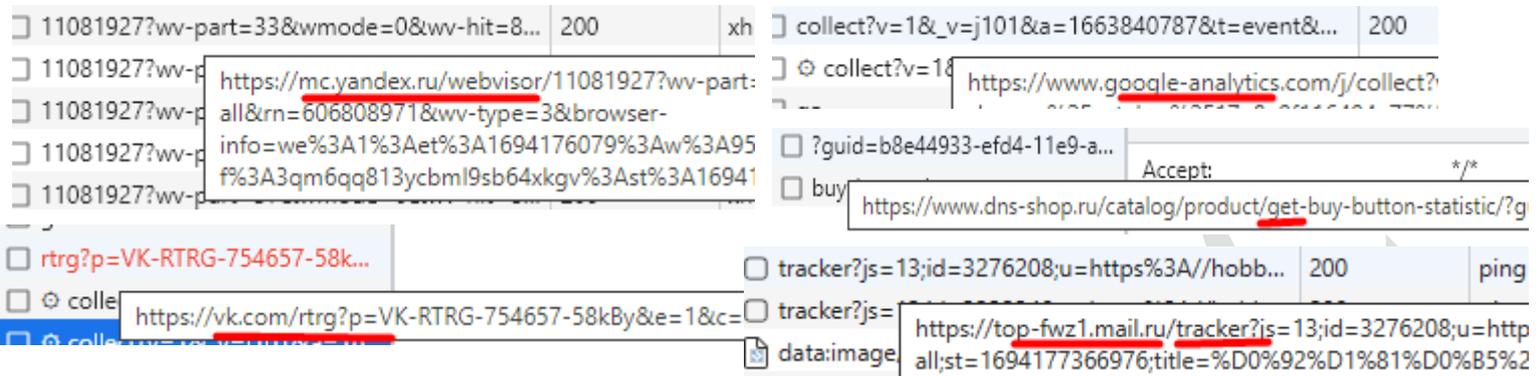
2. Выбрать сеть и включить фильтр для POST запросов



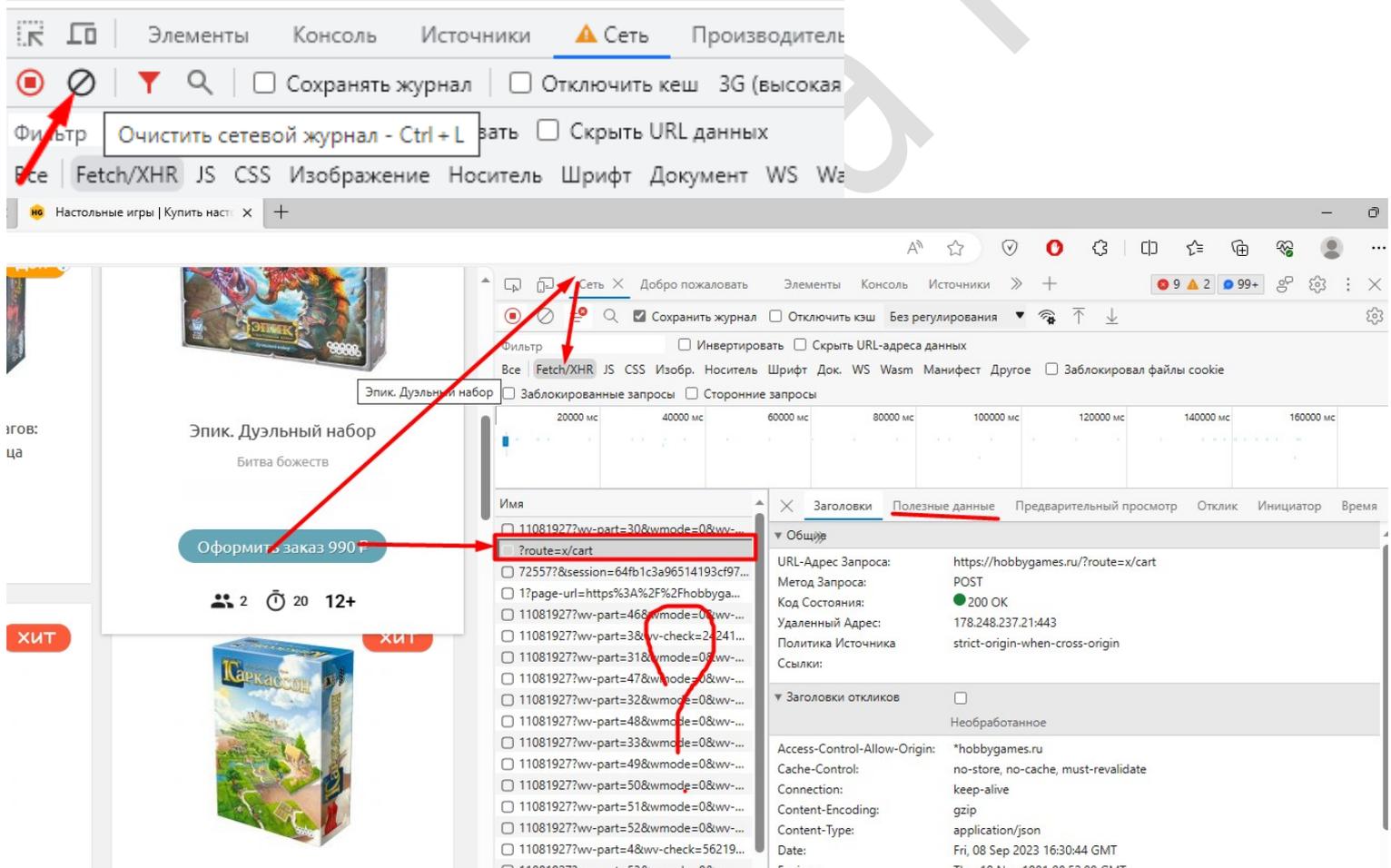
3. Нажать купить игру

5.1 после этого начинается анализ всех запросов, обрабатываем сервером

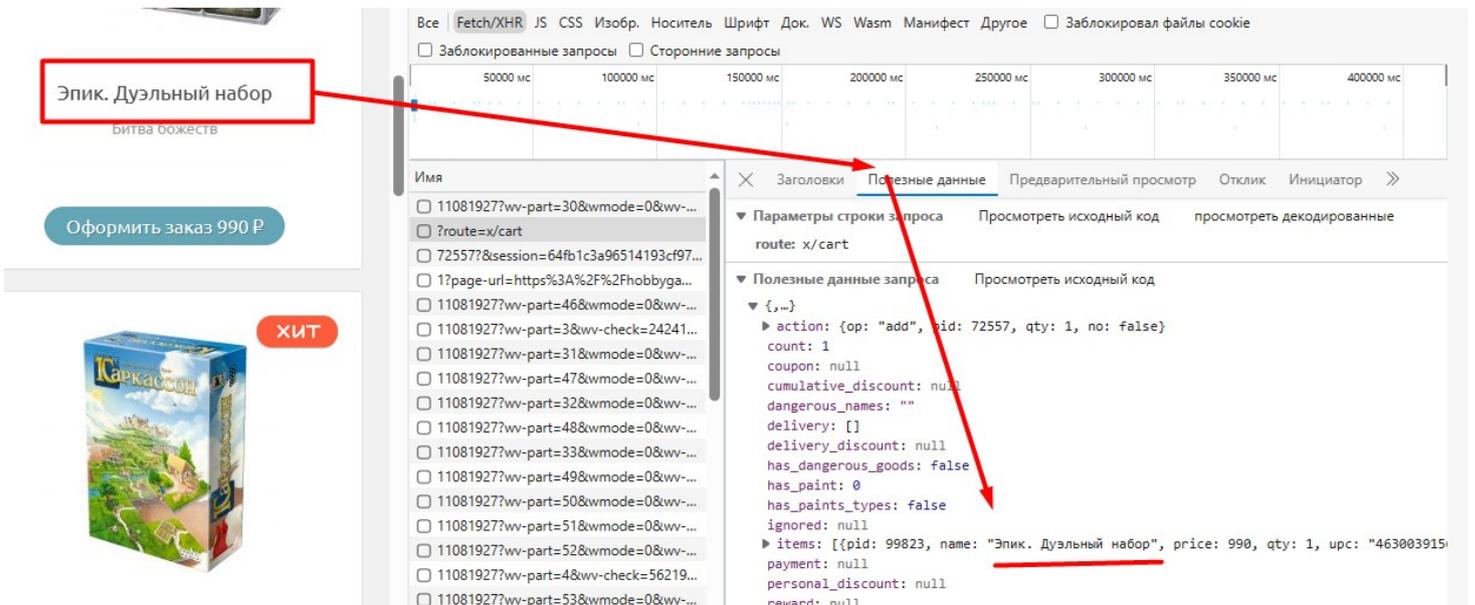
По умолчанию отбрасываем, **аналитику**, различные взаимодействия сбора статистики и отправки в сторонние сервисы, слежение за пользователем (запись всего чего делает пользователь на сайте-это для владельца важно, вдруг вы что-то купили и произошёл сбой у владельца должны храниться доказательства ваших действий и переходов), все **get запросы, трекеры** и методом исключения анализируем оставшиеся запросы помощью для анализа является вкладка **Полезные данные**



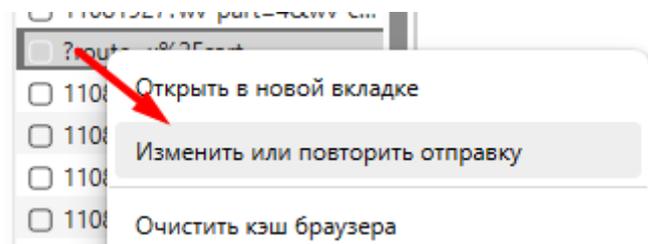
5.2 в данном случае остался только 1 запрос для более простого поиска используйте очистку истории и только после этого добавляйте в корзину товар



Совпадает



5.3 для проверки, тот это запрос или нет нужно выполнить следующие действия:



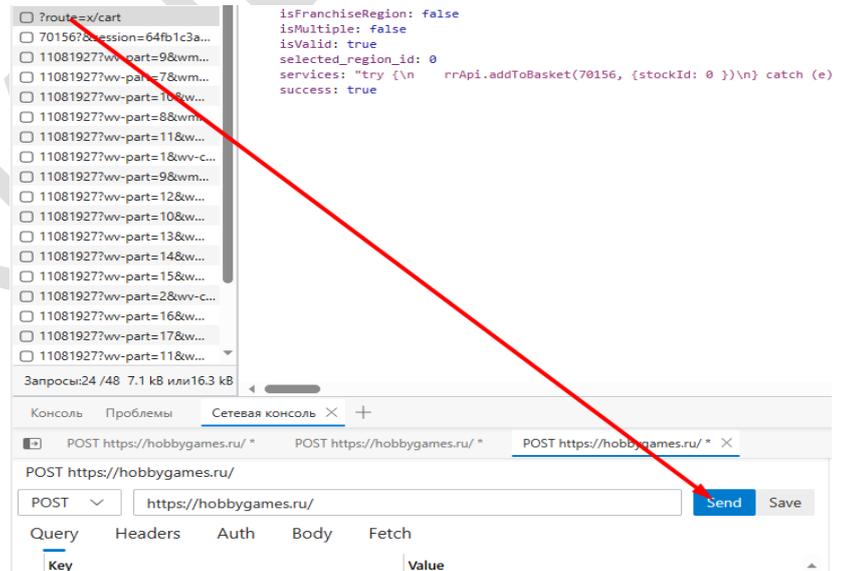
5.3.1 вы должны нажать на данный запрос правой кнопкой мыши и в меню выбрать Отправить повторный запрос для эджа

Иногда не все браузеры дают возможность такое сделать, откройте тогда в другом браузере или используйте постмен (данную программу для тестирования мы будем изучать позднее)

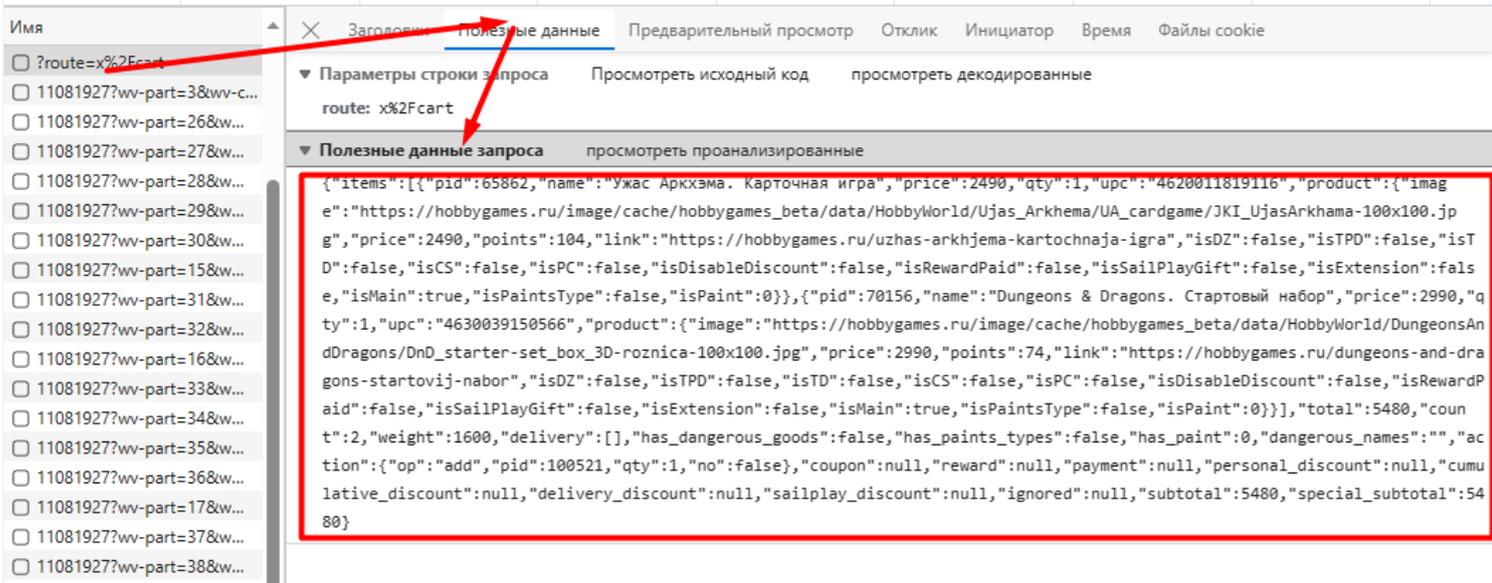
5.3.2 после чего внизу нажать **Send**

5.3.3 не закрывая страницу и не обновляя ее, вам нужно открыть корзину в новой вкладке и проверить, чтобы в корзине изменилось количество купленного товара.

5.5.4 если количество изменилась вы нашли нужный вам запрос, если нет ищите дальше. После того, как вы выбрали нужный запрос



5.3 выберите данный запрос **?route=x/cart** нажмите полезные данные и **Просмотреть исходный код**, после чего скопируйте весь код в BODY джиметра

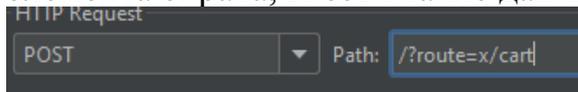


Шаг 3

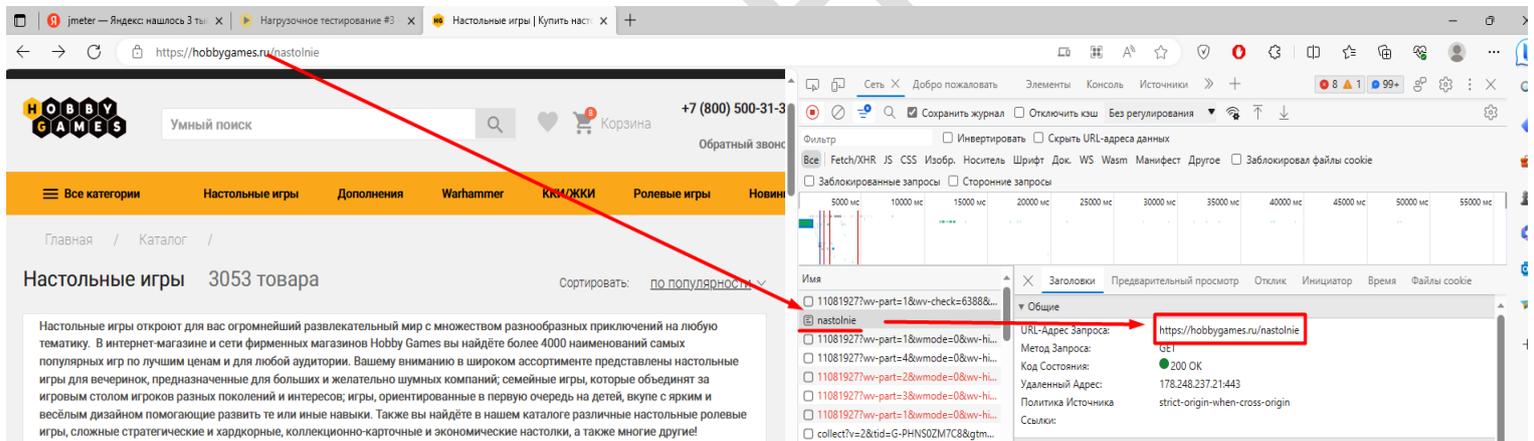
4. откройте джиметр

5. Нажать правой кнопкой мыши на **HTTP запрос POST методом**, после чего в свойствах данного

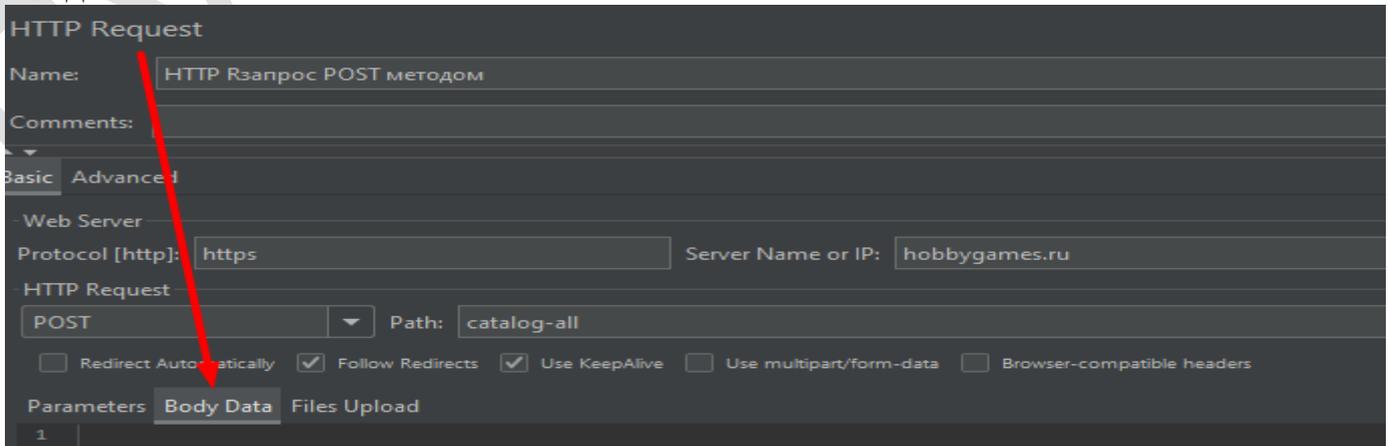
элемента справа, ввести такие данные, как :



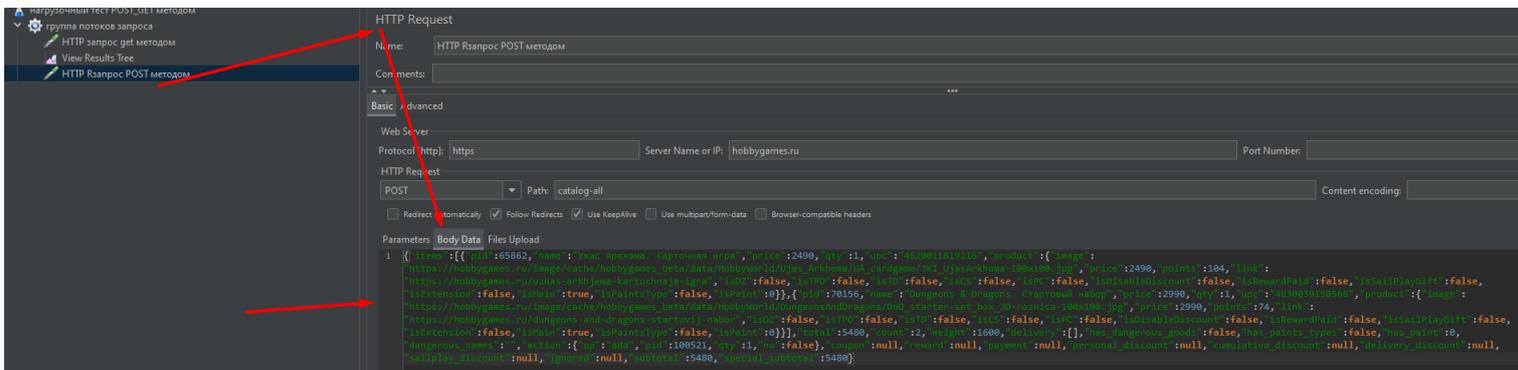
данные для POST запросов брать из панели разработчика, страницы



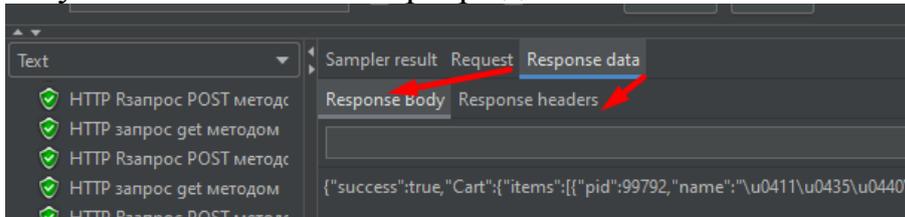
6. Нажать боди



7. Вставить туда скопированный код из панели разработчика

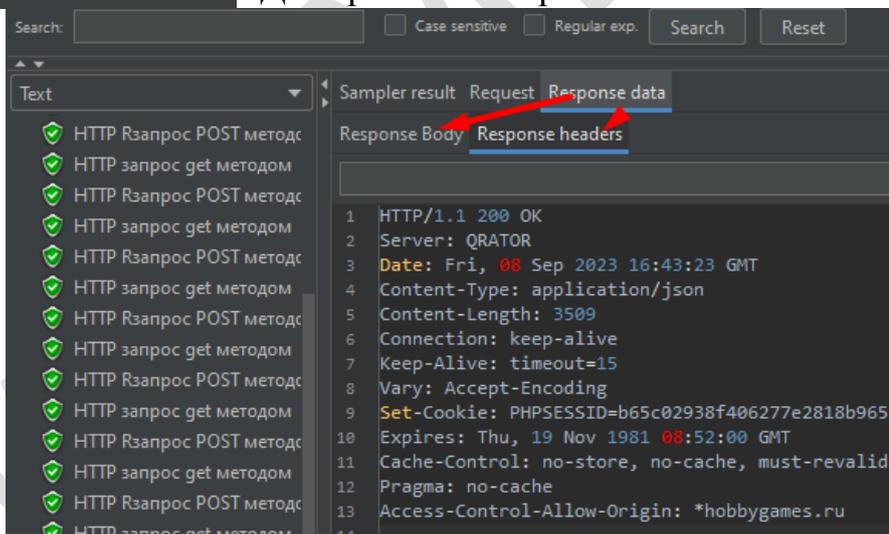


Результаты ответов от сервера делится на заголовки и тело :



здесь мы смотрим отработал наш запрос или нет и что в результате нам отвечает сервер.

Для сравнения верный ответ или нет

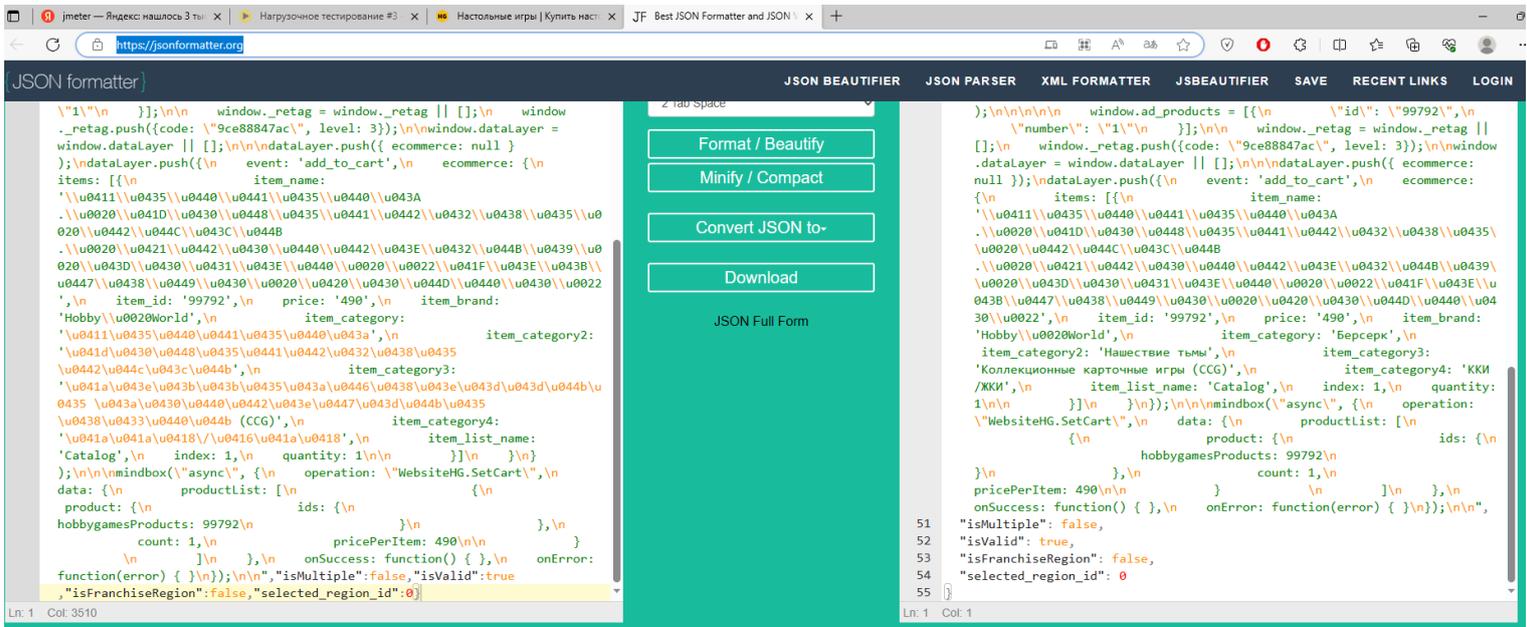


можно скопировать тело ответа.

Перейти в Панель разработчиков нажать предварительный просмотр и сравнить ответ сервера с вашим ответом(у вас не форматированный ответ)

Для идеального сравнения перейдите по ссылке [Best JSON Formatter and JSON Validator: Online JSON Formatter](#)

Слева вставьте тело ответа от вашего сервера а справа с панели разработчика с вкладки предварительный просмотр



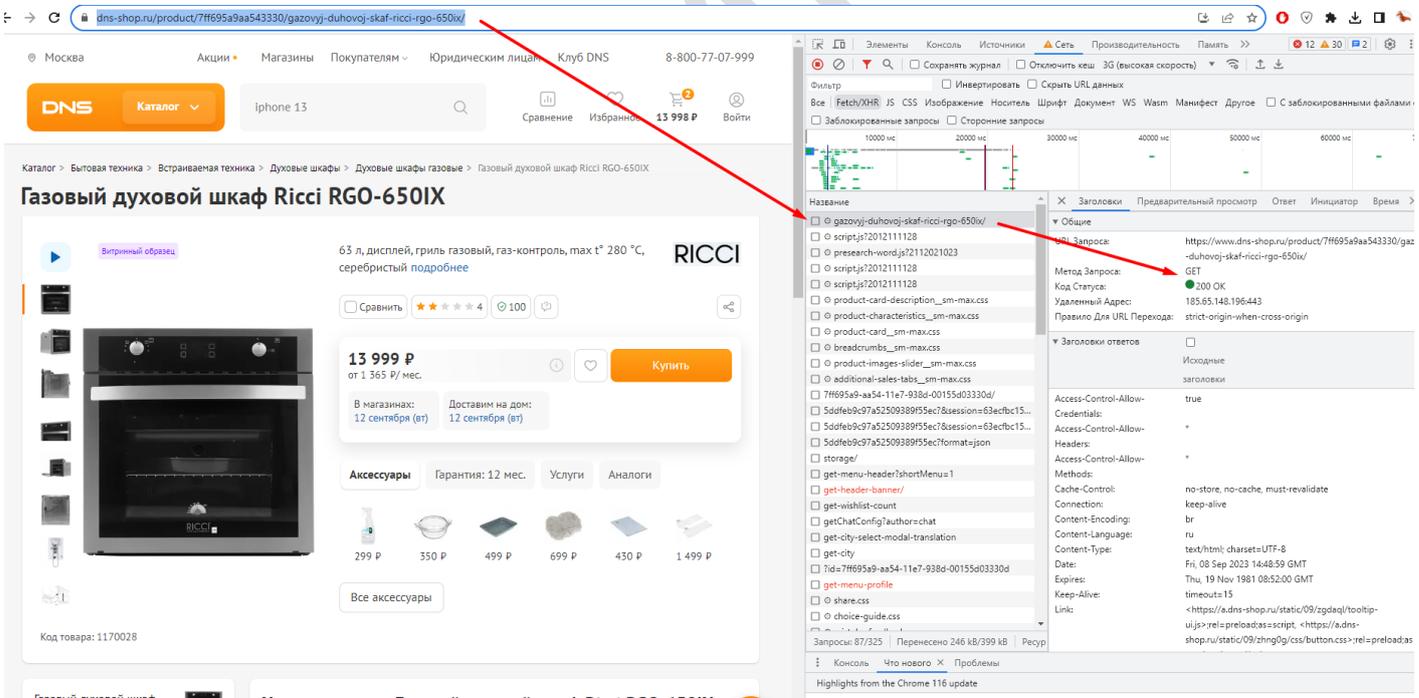
Задание 2 Добавить POST запросу график вывода для анализа результатов статистики.

Задание 3 Создайте нагрузочный тест в джиметре, только с POST методом отправки запросов для сайта DNS. Результаты вставьте и проанализируйте результат, что вам возвращает сервер и как обрабатываются запросы по графику.

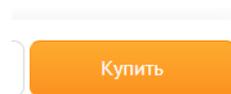
Предположите, почему сервер вам возвращает такие данные.

Пример Get запрос

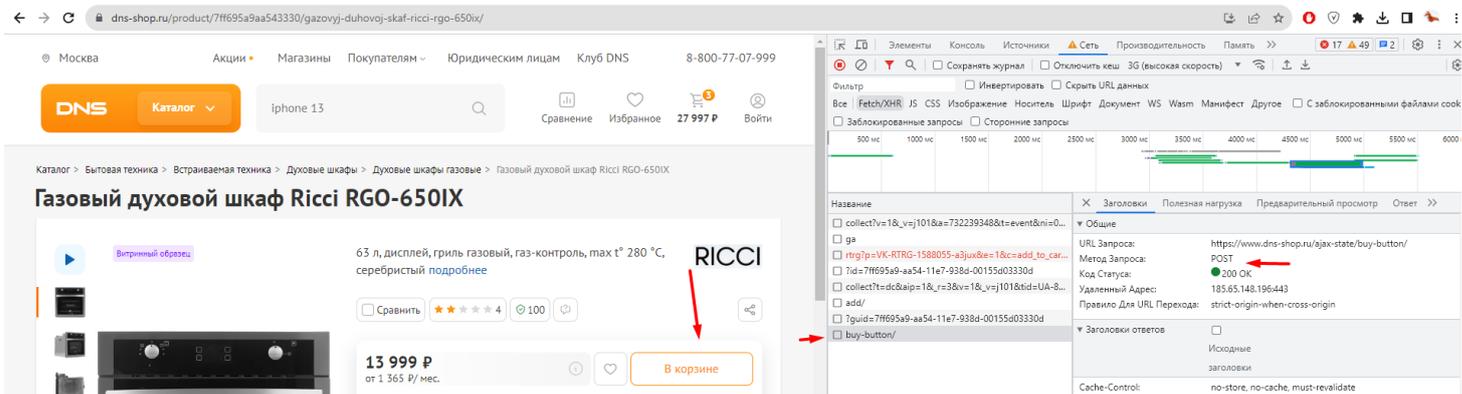
<https://www.dns-shop.ru/product/7ff695a9aa543330/gazovyj-duhovoij-skaf-ricci-rgo-650ix/>



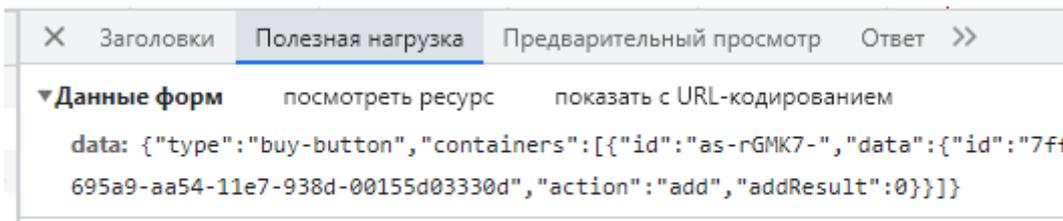
Что бы показать Post запрос жмем



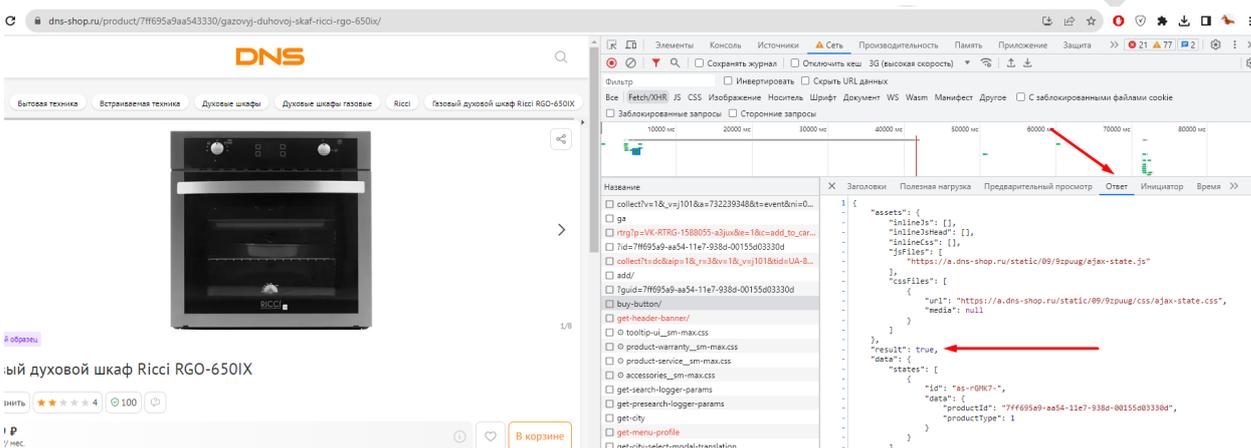
купить кнопку



Данные которые мы отправили с вкладки полезная нагрузка



На вкладке ответ можно посмотреть ответ от сервера



Результатом стало добавление в корзину товара.

Задание 4 Создайте нагрузочный тест только с POST методом отправки запросов для сайта VK, отправки смс с текстом «Купить слона». Результаты вставьте и проанализируйте на минимальных и средних показателях. Анализ по графику выполнить. И показать, что вам возвращает сервер и что возвращается в панели разработчика на вкладке предварительный просмотр.

Задание 5 создайте get запрос в поисковую строку яндекс с текстом «крутая киношка»

Что вам возвращает сервер?

Если вам нужно 5 оценка

Задание 6 найдите ресурс и отправьте любой пост запрос, джиметром должны обрабатываться и отдаваться положительные ответы от сервера, а не блокируются ресурсом и возвращают NULL.

контрольные вопросы:

1. чем отличается Get и POST запросы?
2. Как в джиметре создать нагрузочный тест POST методом?
3. Что мы вставляем в тело запроса POST методом?
4. Какие данные при анализе запросов на сервер при работе с пост запросами мы откидываем?
5. Как включить фильтр по пост запросам в панели разработчика?
6. Как проверить нужный ли нам пост запрос?
7. Как скопировать код из панели разработчика для вставки в body джиметра?

8. Как вызвать панель разработчика?
9. Как очистить истории движения наших переходов в панели разработчика?
10. Из чего состоит HTTP-запрос? перечислите полный набор параметров.
11. Приведите по несколько примеров POST и GET запросов

Пушкина М.С.

Практическая 19-20

Тема: автоматизированное нагрузочное тестирование ресурса с инструментом JMeter. Работа с POST методом. POST запросы с сохранением cookie файлов. Автоматическое заполнение HTTP CookieManager. Анализ ответов от сервера.

ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

Cookie (куки) — это небольшие текстовые файлы, сгенерированные сайтами. В них записана информация о посещении сайта. Эти файлы хранятся на компьютере.

Cookie нужны, чтобы идентифицировать пользователя. Когда вы вернётесь на сайт, он узнает вас и автоматически подстроится. Формы регистрации будут заполнены, языковые и региональные настройки заданы. Если на сайте вы выбирали товары, то новые будут предложены на основе ваших предпочтений.

Вот как это работает. После того как вы вводите адрес нужной страницы в браузере, браузер ищет на устройстве файл cookie этого сайта. Если этот файл найден, то он отправляется на сервер ресурса. Сайт получает его и начинает использовать. Если браузер не находит cookie, сайт считает вас новым посетителем и просит разрешения создать файлы на устройстве.

Какая информация хранится в cookie

Информация, которая хранится в cookie-файлах, может быть практически любой. Это зависит от того, что нужно сайту, который вы посещаете. У каждого сайта свой набор cookie, но больше всего информации собирают и хранят интернет-магазины. Обратим внимание на основные данные:

- **Индивидуальные настройки.** Это могут быть язык, город, масштаб страницы, валюта. Cookie избавляют вас от необходимости заново настраивать сайт при каждом входе.

- **Данные авторизации.** Если на сайте предусмотрен личный кабинет, то cookie хранят логин и пароль.

- **Личные данные.** Сайты могут хранить номера телефонов, паспортные данные, адреса, платёжные данные. Всё для того, чтобы вы не вводили одно и то же каждый раз при покупках или заказе услуг.

- **Данные об устройстве, с которого вы зашли на сайт.** Эта информация нужна владельцам сайтов для оптимизации работы своих ресурсов. Сохраняется время сессии на сайте, модель устройства, версия браузера.

- **Переходы и клики.** Когда вы заходите на сайт, вам присваивается идентификатор, который хранится в файлах cookie. По нему можно отследить ваше перемещение по сайту или клики на баннеры с рекламой. Эта информация нужна, чтобы сделать сайты более удобными.

- **HTTP** - это протокол или определенный набор правил для доступа к ресурсам в Интернете. Ресурсы могут означать что угодно, от HTML-файлов до данных из базы данных, фотографий, текста и так далее.

- Эти ресурсы становятся доступными для нас через API, и мы отправляем запросы к этим API через протокол HTTP. API расшифровывается как интерфейс прикладного программирования. Это механизм, который позволяет разработчикам запрашивать ресурсы.

Пользовательские куки

Куки в джиметре, добавленные с помощью CURL формата подгруженные автоматически

HTTP CookieManager

Comments: Created from cURL on 2023-09-08T23:56:43.91

Options

Clear cookies each iteration?

Use Thread Group configuration to control cookie clearing

standard

User-Defined Cookies

Name:	Value	Domain
SESSIONID	rr9a06pq8hgbb04k7a6foc02eq	shop.miratorg.ru
clientid	20230908205100	shop.miratorg.ru
BITRIX_SM_SALE_UID	81963d69403468bce67cc15b29f1a91	shop.miratorg.ru
session-cookie	1782fdc976e33d74371fc6ddeb261f52d3132cd9d3311...	shop.miratorg.ru
deduplication_cookie	yandex	shop.miratorg.ru
rpvid	239599175652589	shop.miratorg.ru
_ym_uid	1694195462108531955	shop.miratorg.ru
_ym_d	1694195462	shop.miratorg.ru
rcuid	64fb1c3a96514193cf978df1	shop.miratorg.ru
advcake_track_id	7e6451d9-b405-4a5e-af4d-aabbf9edb135	shop.miratorg.ru
advcake_session_id	4633c68a-534d-28e4-d668-2e0638b236d9	shop.miratorg.ru
advcake_track_url	https%3A%2Fshop.miratorg.ru%2Fproduct%2Fp...	shop.miratorg.ru
advcake_utm_partner	yandex	shop.miratorg.ru
advcake_utm_webmaster	astat_7%7Cdsa_7%7Ccid_90896347%7Cgid_52556386...	shop.miratorg.ru
tmr_lvuid	1ec7cfc066e7b6dcec11c15a473065d	shop.miratorg.ru
tmr_lvuidTS	1694195462821	shop.miratorg.ru
_gid	GA1.2.569912213.1694195463	shop.miratorg.ru
_ym_isad	1	shop.miratorg.ru
tt_deduplication_cookie	yandex	shop.miratorg.ru
BX_USER_ID	5086a26bb3c0aac9177b11df5de2bd28	shop.miratorg.ru
BITRIX_CONVERSION_CONTEXT_s1	%7B%22ID%22%3A5%2C%22EXPIRE%22%3A1694206...	shop.miratorg.ru
cat_sort	popular	shop.miratorg.ru
cat_count	30	shop.miratorg.ru
skwb24_popups_pp1	Y	shop.miratorg.ru
_ym_visorc	w	shop.miratorg.ru
rwpswu	true	shop.miratorg.ru
_gat_UA-177455897-1	1	shop.miratorg.ru
_gat_gtag_UA_177455897_1	1	shop.miratorg.ru
rr-viewitemid	80468	shop.miratorg.ru
rreviewed	124253%2C80468	shop.miratorg.ru
_ga	GA1.2.353115491.1694195462	shop.miratorg.ru
tmr_detect	1%7C1694204161896	shop.miratorg.ru
BITRIX_SM_skyweb24Popup_1	1694204162	shop.miratorg.ru
cto_bundle	ocekq19BWW10dFR2RDKiMkRRlpZRUhVNUc4emxkW...	shop.miratorg.ru
rbasket	124253%2C124253%2C124253%2C124253%2C124253...	shop.miratorg.ru
rflvt	1694204162674	shop.miratorg.ru
_ga_RP7HK1VQ2V	GS1.1.1694201756.2.1.1694204177.15.0.0	shop.miratorg.ru

CURL

— это утилита командной строки для передачи данных с или на удаленный сервер с использованием одного из поддерживаемых протоколов. Он установлен по умолчанию в macOS и большинстве дистрибутивов Linux.

CURL используется разработчиками для тестирования API, просмотра заголовков ответов и выполнения HTTP-запросов.

В этой статье мы собираемся объяснить, как использовать cURL для выполнения запросов POST. Метод HTTP POST используется для отправки данных на удаленный сервер.

Заголовки запросов в джиметре подгруженные автоматически

HTTP HeaderManager

Comments: Created from cURL on 2023-09-08T23:56:43.91

Headers Stored in the Header Manager

Name:	Value
authority	shop.miratorg.ru
accept	*/*
accept-language	ru,en;q=0.9,en-GB;q=0.8,en-US;q=0.7
bx-ajax	true
content-type	application/x-www-form-urlencoded
origin	https://shop.miratorg.ru
referer	https://shop.miratorg.ru/product/voda-arkhyz-pitevaya-negaz-5l-pet-rossiya/
sec-ch-ua	"Chromium";v="116", "Not)A;Brand";v="24", "Microsoft Edge";v="116"
sec-ch-ua-mobile	?0
sec-ch-ua-platform	"Windows"
sec-fetch-dest	empty
sec-fetch-mode	cors
sec-fetch-site	same-origin
user-agent	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
x-ajax-token	e7eebe2e05e9a9a0dab591e526c9f84984b1d6f1417a38a595da22b60e8707ac
x-bitrix-csrf-token	1e45d9153570b25564ed294f317f50da
x-bitrix-site-id	s1
x-requested-with	XMLHttpRequest
Accept-Encoding	gzip, deflate

The screenshot shows the Chrome DevTools Network tab. A list of requests is visible on the left, with one request selected. The right pane shows the details of this request, including the 'Заголовки запросов' (Request Headers) section, which is highlighted with a red box. The headers include:

authority:	shop.miratorg.ru
method:	POST
path:	/bitrix/services/main/ajax.php?action=miratorg%3Aaction.app.ajax.request
scheme:	https
Accept:	*/*
Accept-Encoding:	gzip, deflate, br
Accept-Language:	ru,en;q=0.9,en-GB;q=0.8,en-US;q=0.7
Bx-Ajax:	true
Content-Length:	60
Content-Type:	application/x-www-form-urlencoded
Cookie:	SESSIONID=rr9a06pq8hgbb04k7a6foc02eq; clientid=20230908205100; BITRIX_SM_SALE_UID=81963d69403468bce67fcc15b29f1a91; session-cookie=1782fdcb976e33d74371fc6dbeb261f52d3132cd9d331190588142df1a6a265dff2a5725615b013bf30343ca5682a805; deduplication_cookie=yandex; deduplication_cookie=yandex; rmid=230500175652580; um_uid=1604105462108521055

ХОД РАБОТЫ

1. запустите джиметр
2. переименуйте группу запросов в ТЕСТ_POST_ЗАПРОС
3. перейдите на сайт [Вода питьевая негазированная Tassay 5л, цена - купить с доставкой в официальном интернет-магазине Мираторг \(miratorg.ru\)](https://www.miratorg.ru/)
4. вызовите панель разработчика f12
5. добавьте себе воду в корзину

Задание 1 создать post запрос на добавление 10 бутылок воды в корзину пользователя. Найти запрос в панели разработчика, проверить, точность запроса. В запросе указать количество 10 и отправить на сервер. Выполнить через панель разработчика.

1. найдите данный пост запрос в панели разработчика
2. отправьте пост запрос на сервер с количеством 10

Сеть × Добро пожаловать Элементы Консоль Источники >> + 26 1 99+

Сохранить журнал Отключить кэш Без регулирования

Фильтр Инvertировать Скрыть URL-адреса данных

Все **Fetch/XHR** JS CSS Изобр. Носитель Шрифт Док. WS Wasm Манифест Другое Заблокировал файлы cookie

Заблокированные запросы Стронние запросы

Имя

- 47692048?wv-part=3&wmode=0&wv-hit=1683464...
- 47692048?wv-part=4&wmode=0&wv-hit=1683464...
- 33b41e96b9a2d9982668e8751e608a19
- ajax.php?action=miratorg%3Aaction.app.ajax.request
- ajax.php?action=miratorg%3Aaction.app.ajax.request**
- rtrg?p=VK-RTRG-1205133-4ry4s&products_events=...
- ajax.php
- ajax.php
- 47692048?wv-part=5&wmode=0&wv-hit=1683464...

Запросы: 44 / 106 19.9 kB или 42.8 kB передано Ресурсы: 1

Консоль Проблемы **Сетевая консоль** × +

▼ Коллекции

Вы не определили ни одной коллекции. Нажмите кнопку выше, чтобы импортировать коллекцию с диска или создать новую пустую коллекцию.

▼ Среды

Вы не определили ни одной среды. Нажмите кнопку выше, чтобы импортировать среду с диска или создать новую пустую среду.

POST https://sh.../main/ajax.php * ×

POST https://shop.miratorg.ru/bitrix/services/main/ajax.php

POST **Send** Save

Query Headers Auth **Body** Fetch

<input checked="" type="checkbox"/>	args[product_id]	80468	<input type="button" value="x"/>
<input checked="" type="checkbox"/>	args[quantity]	10	<input type="button" value="x"/>
	Key	Value	

Computed URL: https://shop.miratorg.ru/bitrix/services/main/ajax.php?action=miratorg%3Aaction.app.ajax.request

Preview **Body** Headers Cookies

```
1 [{"status": "success", "data": {"80468": {"id": "80468", "quantity": 10, "name": "\u0412\u043e\u0434\u0430 \u0430\u0440\u0445\u0438\u0437 5\u043b"}}]
```

Status: 200 Time: 185ms Size: 505 bytes [Download](#)

3. После чего откройте новую вкладку, перейдите в корзину и после отправки запроса с количеством 10 шт обновите корзину!
если результат такой

Уменьшайте количество, значит на сайте нет такого количества товара.

4. Уменьшили пост запрос до 5шт

Извините, некоторые из выбранных товаров отсутствуют.

Товар временно отсутствует

1 товаров на 1399 Р (50 кг) [Удалить все](#)

Вода питьевая негазированная Архыз 5л
1 уп / 5 кг 139.9 Р за уп

напитки, соки, вода избранное

Вода питьевая негазированная Архыз 5л

1 уп / 5 кг В наличии 5 шт.

Цена **139,9 Р**

ТОВАР В КОРЗИНЕ - 2 шт +

Вода Архыз природная питьевая негазированная 2 штуки по 5 л, для детского питания.
Для детей старше 3 лет.
В одном заказе мы доставляем не более 24 бутылей питьевой воды емкостью от 5 л

Имя

Заголовки **Параметры строки запроса** Предварительный просмотр Отклик

Параметры строки запроса: `action: miratorg.action.app.ajax.request`

Данные формы: `method: basket.update`
`args[product_id]: 80468`
`args[quantity]: 2`

Консоль Проблемы Сетевая консоль

Коллекции: POST `https://shop.miratorg.ru/bitrix/services/main/ajax.php`

POST `https://shop.miratorg.ru/bitrix/services/main/ajax.php`

Query	Headers	Auth	Body	Fetch
<code>args[product_id]</code>				80468
<code>args[quantity]</code>				5
Key				Value

Ваш заказ
1 товаров на 699.5 Р (25 кг) Удалить все



Вода питьевая негазированная Архыз 5л

1 уп / 5 кг 139.9 Р за уп

Доступно 5 шт

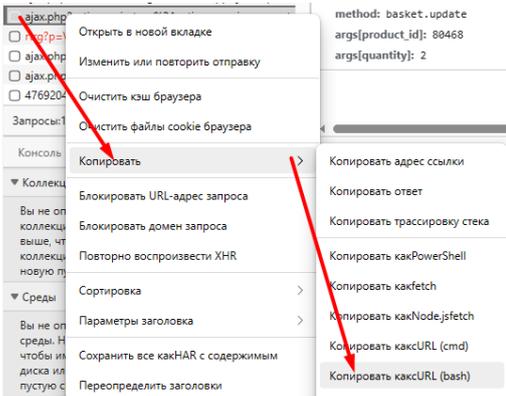
699.5 Р

- 5 +

Задание 2 создать post запрос на добавление 10 любых других товаров другой категории на данном сайте. Найдите запрос в панели разработчика, проверьте, точность запроса. В запросе укажите количество 10 и отправьте на сервер. Выполните через панель разработчика. Определите максимальное количество выбранного товара на сайте.

Задание 3 создать post запрос на добавление 4(определили из 1-го задания) бутылок воды в корзину пользователя. Найдите запрос в панели разработчика, проверьте, точность запроса. В запросе укажите количество 4 и отправьте на сервер. После тоже самое отработайте в джиметре, проверяя ответы сервера с использованием cookie и автоматической генерацией запроса.

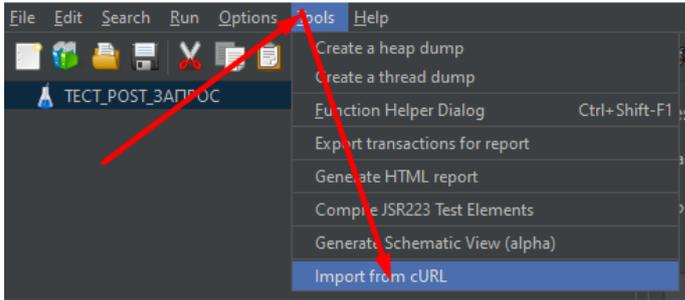
1. Перейдите на сайт найдите post запрос с добавлением товара в корзину, где мы отправляли 5 шт
 - 4.1 нажмите на данный запрос правой клавишей
 - 4.2 после чего нажмите на него левой клавишей и выберите копировать и из списка выберите копировать URL bash



```
method: basket.update
args[product_id]: 80468
args[quantity]: 2
```

2. Откройте джиметр
3. Нажмите на ТЕСТ_POST_ЗАПРОС

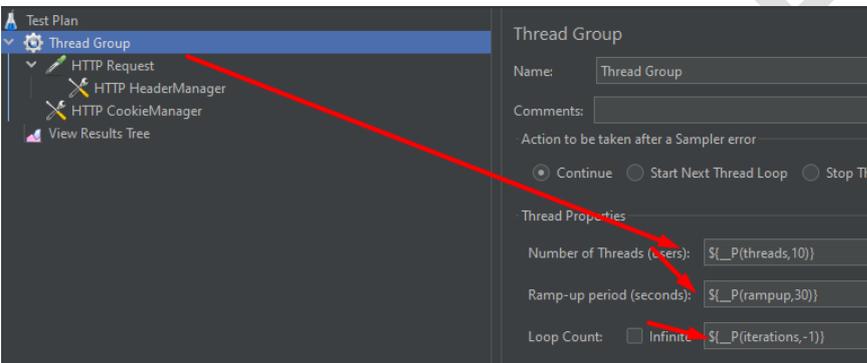
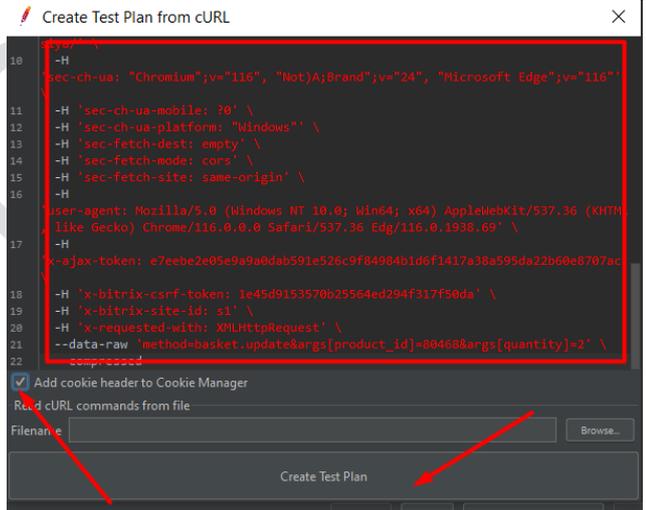
4. группа потоков запроса.jmx (C:\Users\Марина\Desktop\apache-jme



Откройте вкладку TOOL

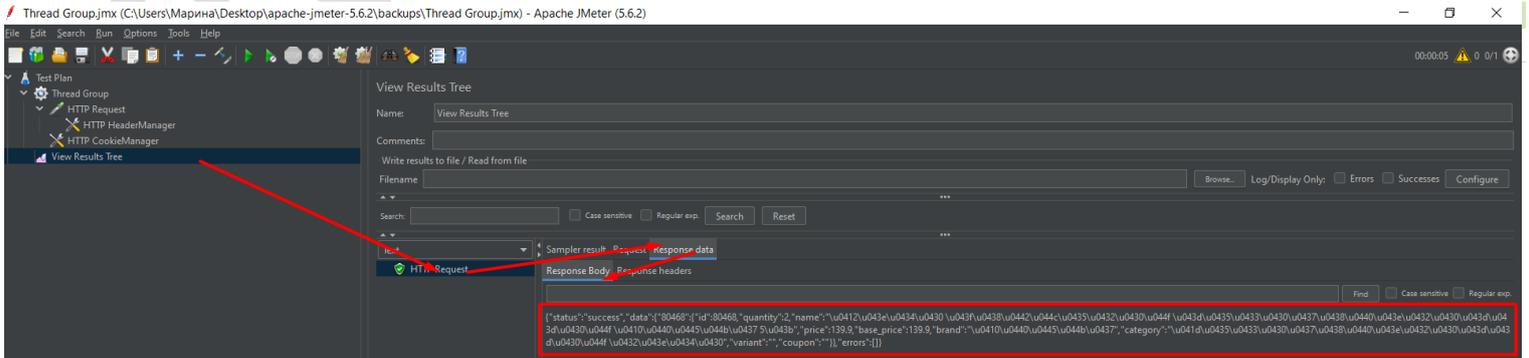
- 4.1 откроется диалоговое окно. Вставьте туда скопированный URL-bush
- 4.2 поставьте галочку напротив cookie
- 4.3 сохранитесь

4.4 группу потоков измените на минимальные параметры 1

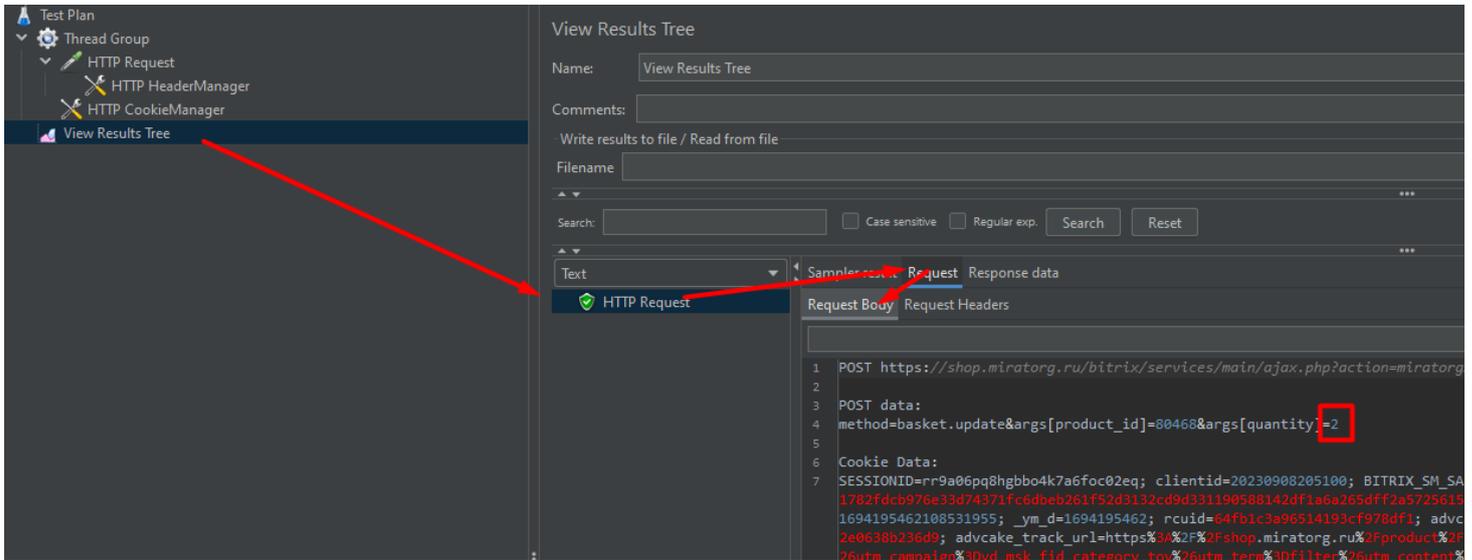


4.5 запустите поток запросов

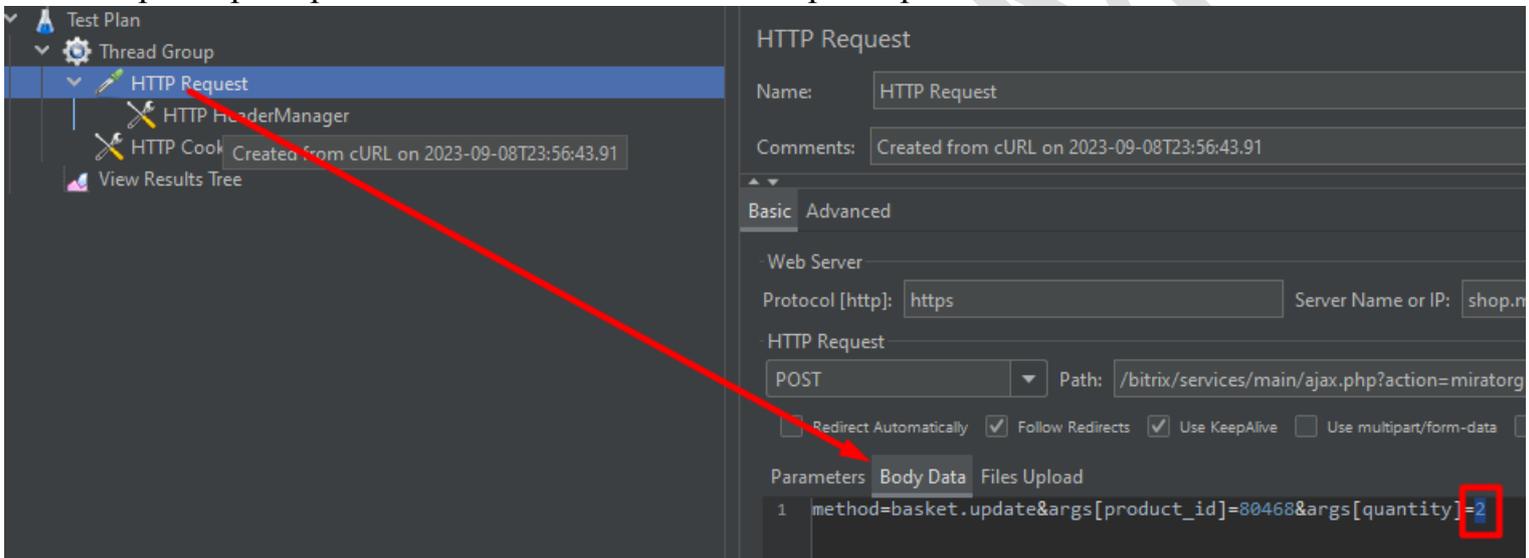
5. откройте ответы сервера , тело запросов. Т.е запрос на сервер отработал



5. Видим какое количество добавляем в корзину 2

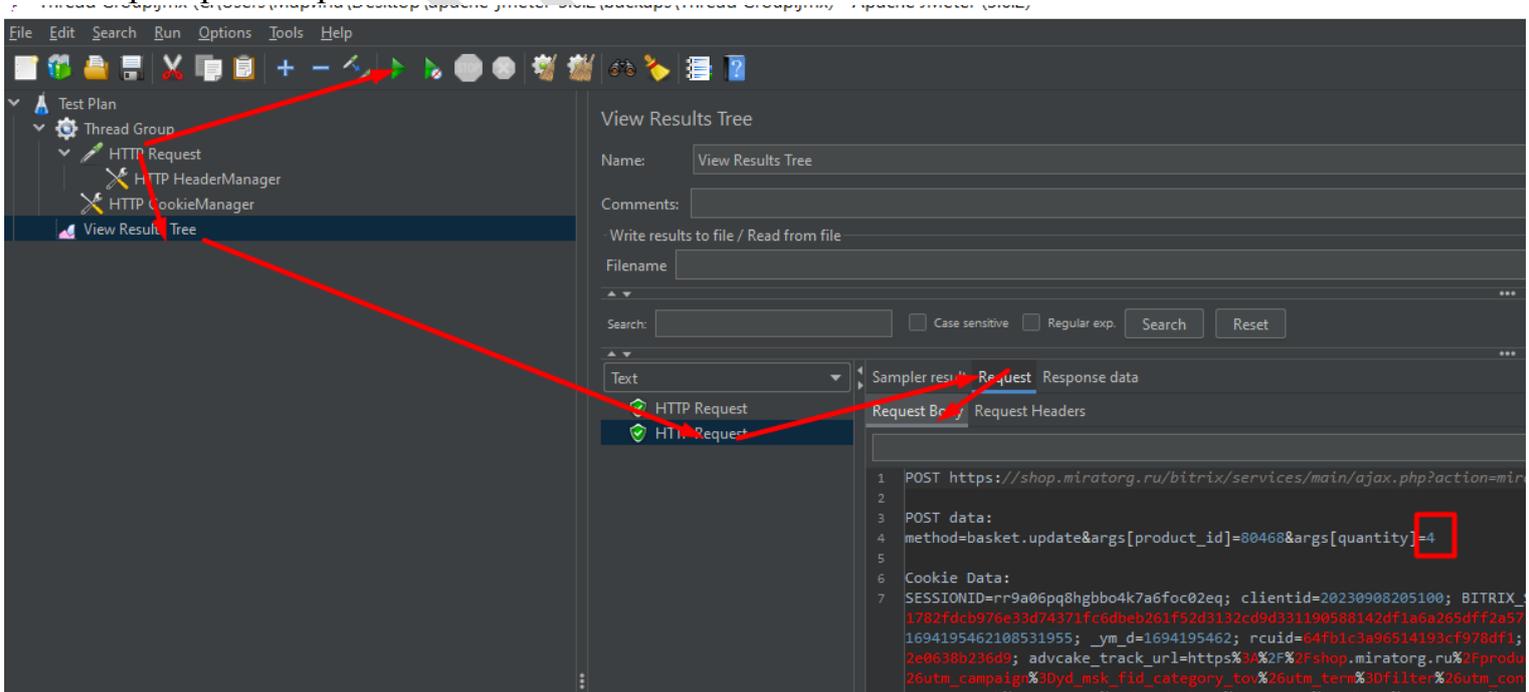


6. Изменить количество товаров с 2 на 4, для этого выберите HTTP request, выберите параметр запроса тело и измените в нем параметр с 2 на 4

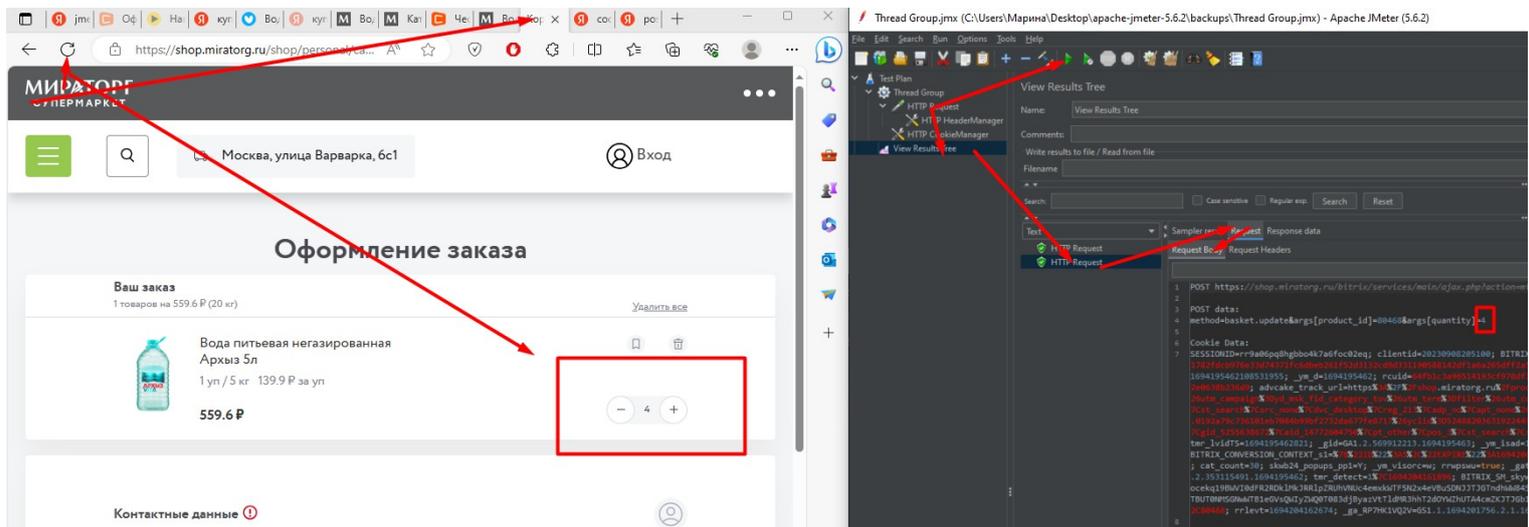


7. Запустите запрос

8. Проверьте отработал ли он



9. Перейдите на сайт и проверьте корзину количество товара должно изменится на 4 шт



Задание 4 создать post запрос на добавление максимального количества товара в корзину(определяем из 2-го задания) пользователя. Найти запрос в панели разработчика, проверить, точность запроса. В запросе указать количество и отправить на сервер. После тоже самое обработать в джиметре, проверяя ответы сервера с использование cookie и автоматической генерацией запроса.

параметры **Thread Group** поставить 20, 3, 2.

Результаты запроса показать, как обрабатывает одновременно на сайте и в джиметре.

Контрольные вопросы:

1. что такое куки?
2. что такое HTTP?
3. Как создать автоматически POST
4. что такое CURL?
5. Как с помощью пост запроса и CURL создать автоматический запрос на сервер?
6. Где в джиметре посмотреть заголовки которые передается с ресурса?
7. Как изменить количество товара через пост запрос?
8. Где глянуть, что ответил нам сервер в ответ на отработанный пост запрос?
9. Как очистить поле выполненных запросов?
10. Как импортировать CURL в джиметр, где его взять?