**Практическая работа Нагрузочное тестирование с JMeter**

JMeter — это программное обеспечение для тестирования с открытым исходным кодом. Это 100% чистое Java-приложение для нагрузочного тестирования и тестирования производительности.

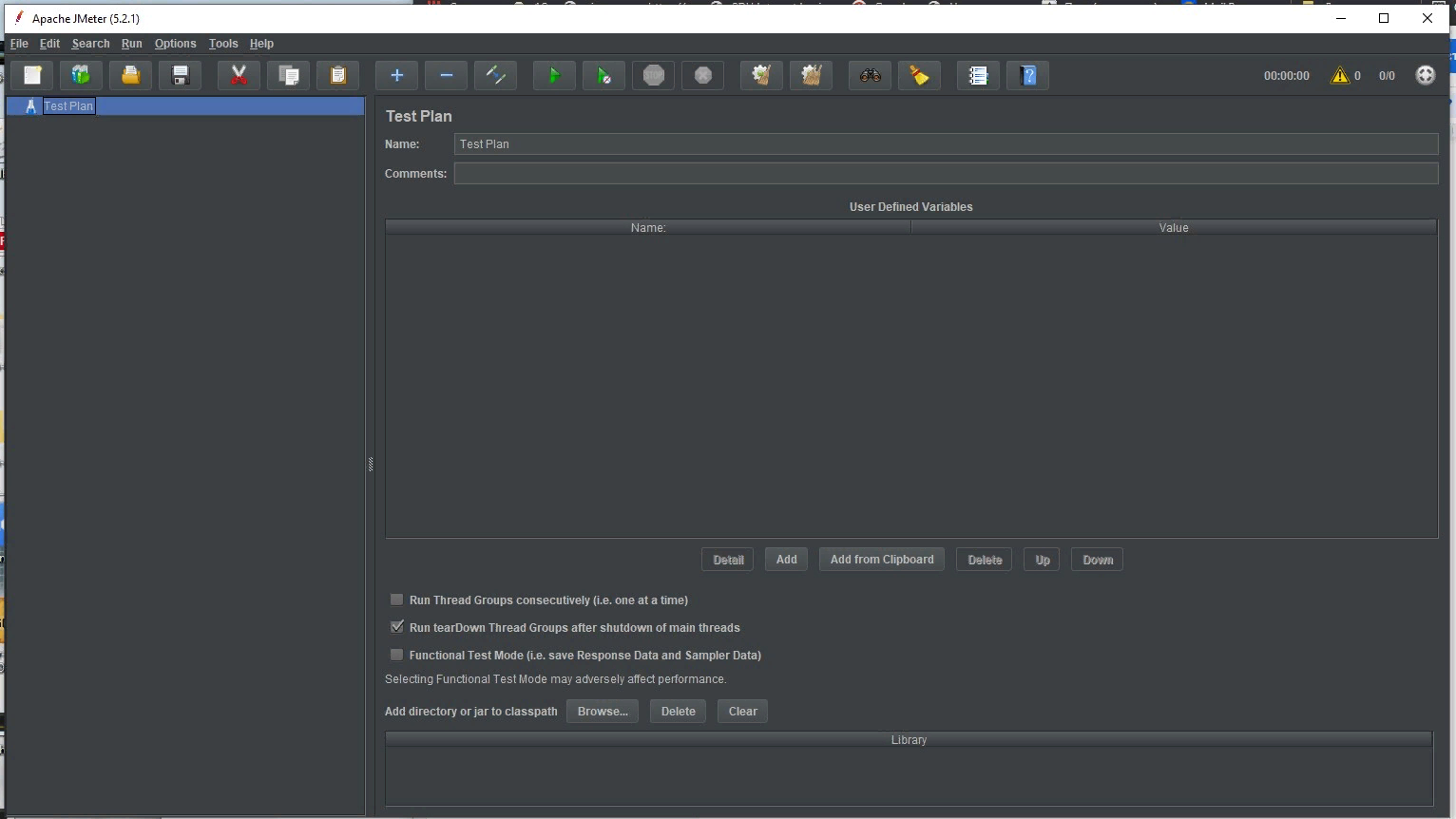
##### **Плюсы:**

* Легко установить
* GUI и non-GUI режим работы (из консоли)
* кроссплатформенность
* Поддержка протоколов (HTTP(S),SOAP,FTP)
* Возможность интеграций с Selenium
* Возможность автоматически генерировать тестовые скрипты

##### **Минусы:** Ресурсозатратный при большой нагрузке и использовании различных репортов в GUI режиме

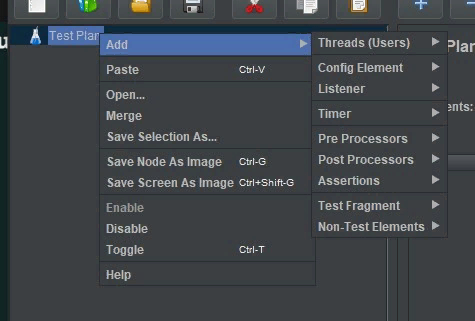
## **Установка JMeter.**

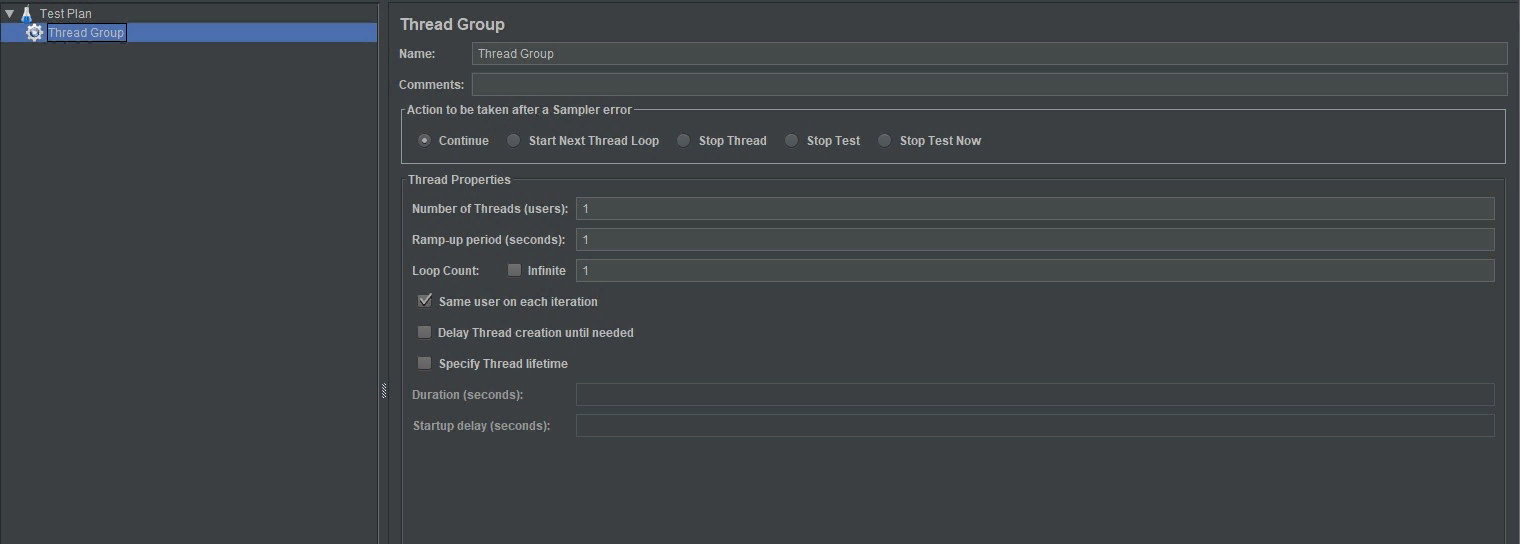
1. Для начала установить java 8+ : [Ссылка для скачивания java 8](https://www.oracle.com/java/technologies/javase/javase-jdk8-downloads.html)
2. Установить JMeter : [Ссылка для скачивания JMeter](https://jmeter.apache.org/download_jmeter.cgi)
3. Скачать и сделать распаковку JMeter
4. Запустить jmeter.bat из папки /bin



## **Основные элементы JMeter.**

1. Thread group - это набор потоков, выполняющих один и тот же сценарий. Установите количество итераций в конфигурации. Поведение потока определяется в соответствии с увеличением и уничтожается по истечении количества итераций в потоке.

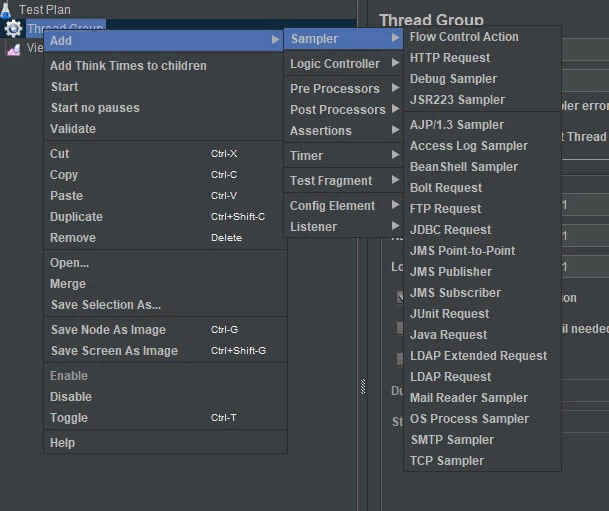




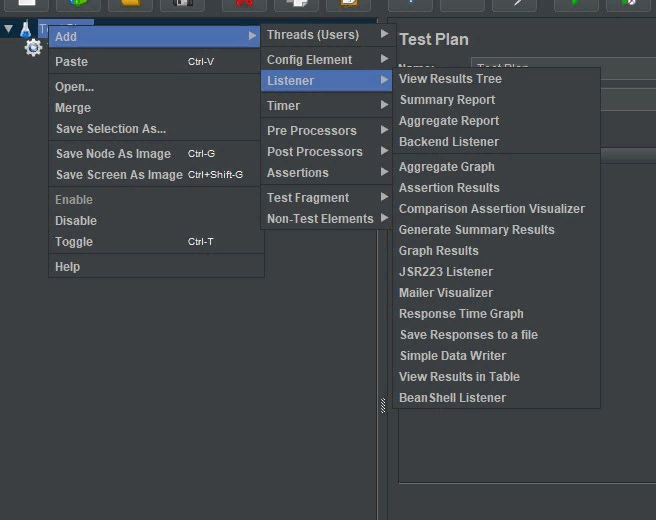
1. Samplers - добавляются как дочерние элементы групп потоков. Они используются для отправки на сервер различных типов запросов. После того, как запрос сэмплера обрабатывается сервером, его ответ возвращается в JMeter, и его можно просматривать и анализировать с точки зрения различных параметров производительности, таких как время отклика, количество обращений в секунду, пропускная способность.

Виды samplers:

* HTTP Request - Позволяет выполнять запросы к WEB-серверу по протоколу HTTP(S).
* FTP Request - Позволяет выполнять запросы к FTP-серверу.
* JDBC Request - Позволяет выполнять SQL-запросы к БД по JDBC API



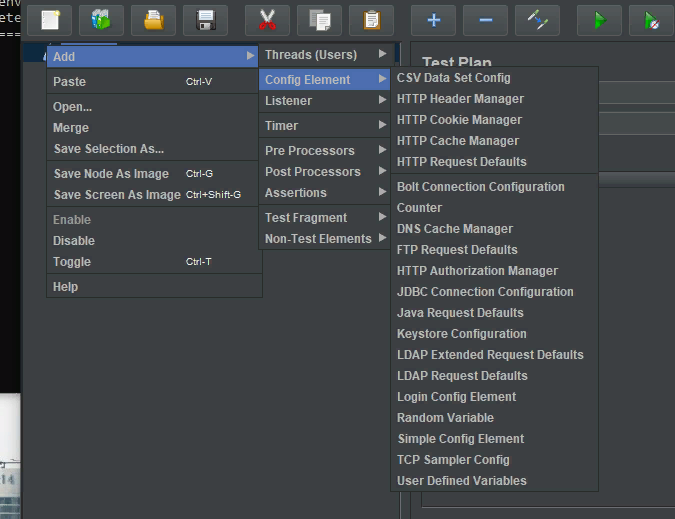
1. Listeners - слушатели показывают результат выполнения теста, они могут отображать результаты в другом формате, например, в виде графика или файла журнала.



Виды listeners:

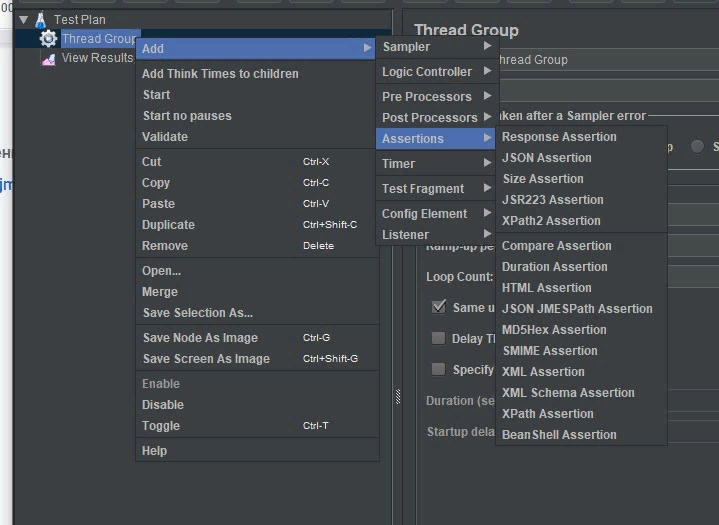
* View Result Tree - Позволяет просматривать детальные результаты выполнения запросов в виде списка.
* View Results in Table - позволяет просматривать результаты выполнения запросов в табличном виде.

1. Configuration - элементы, которые устанавливают значения по умолчанию и переменные для последующего использования сэмплерами.



Виды configurations: - HTTP Request Defaults - Позволяет задавать параметры по умолчанию для HTTP-сэмплеров - JDBC Connection Configuration - Позволяет задавать настройки JDBC-подключения. - FTP Request Defaults - Позволяет задавать параметры по умолчанию для FTP-сэмплеров

1. Assertions- используется для проверки ответа на запрос, который вы отправили на сервер. Это процесс, в котором вы проверяете ожидаемый результат с помощью фактическим результатом запроса во время выполнения.



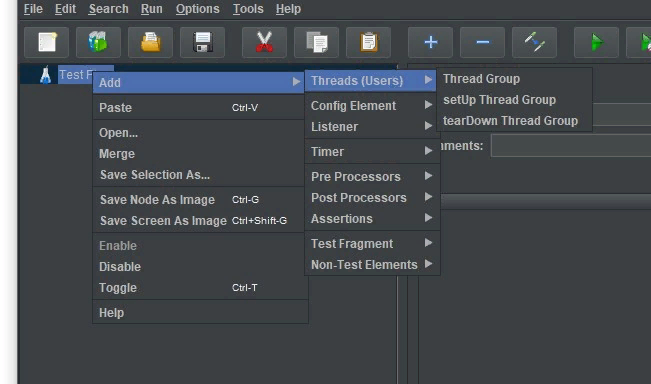
Виды assertions:

* Response Assertion - позволяет проверять содержимое запроса/ответа
* Duration Assertion - позволяет проверять длительность выполнения запроса
* Size Assertion - Позволяет проверять размер ответа

**Пример нагрузочного теста**

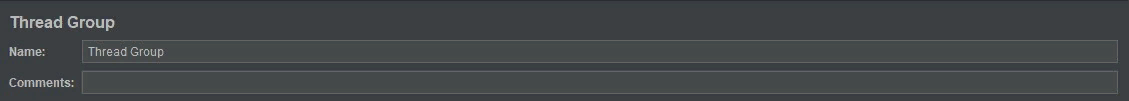
Шаг 1.

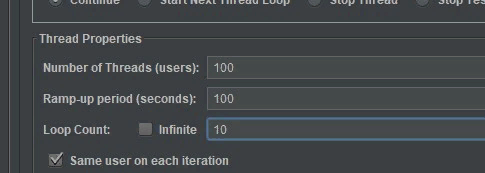
* Добавьте Thread Group
* Выберите план тестирования в дереве
* Щелкните правой кнопкой мыши «Test Plan» и добавьте новую группу потоков: Add -> Threads (Users) -> Thread Group



На панели управления thread group:

* Задать описание группы

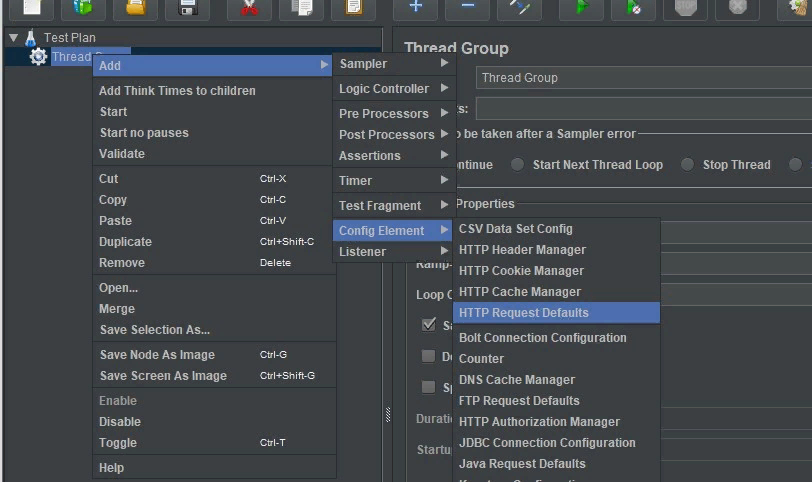
****



* Number of Threads: 100 (Количество пользователей, подключающихся к сайту: 100)
* Ramp-Up Period: 100 (Требуемое время для запуска всех потоков (пользователей).
* Loop Count: 10 (Количество циклов исполнения пользователями действий в Thread Group)

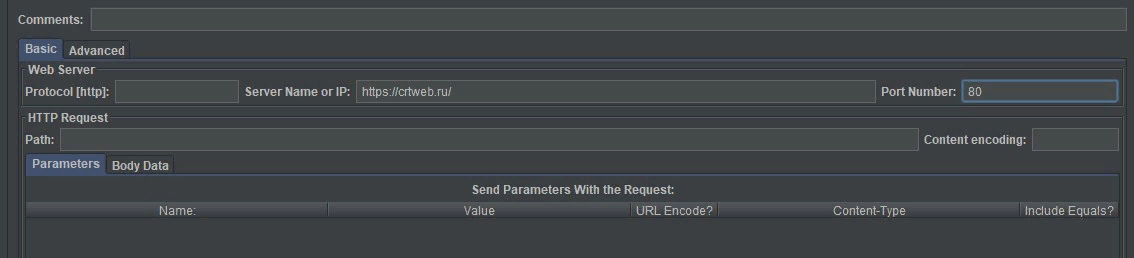
## Шаг - 2 : Создание тест - плана и конфигурации.

* HTTP request Default Этот элемент можно добавить, щелкнув правой кнопкой мыши thread group и выбрав: Add -> Config Element -> HTTP Request Defaults.

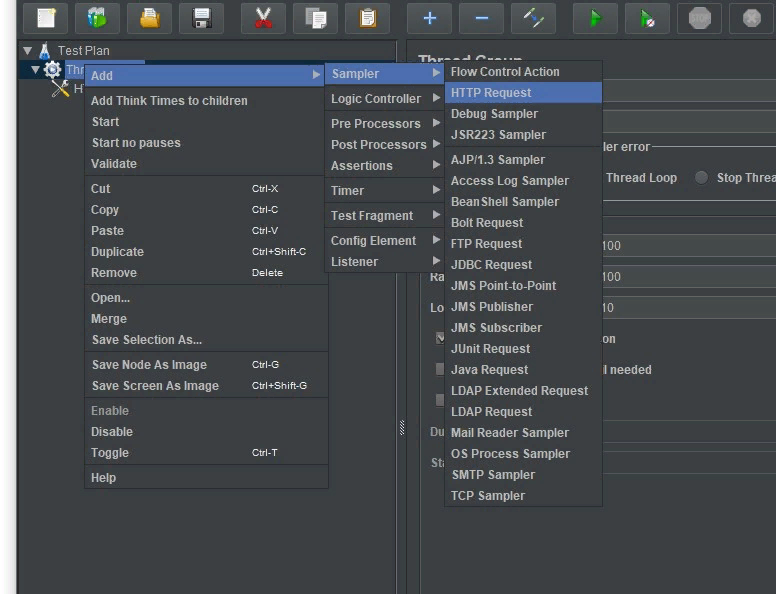
****

* HTTP Request Defaults - это очень простой и ключевой элемент JMeter. Этот элемент конфигурации используется, когда все запросы в сценарии JMeter отправляются на один и тот же сервер. Вы можете добавить один элемент HTTP-запроса по умолчанию в план тестирования с правильным именем или IP-адресом сервера в поле «Имя или IP-адрес сервера».

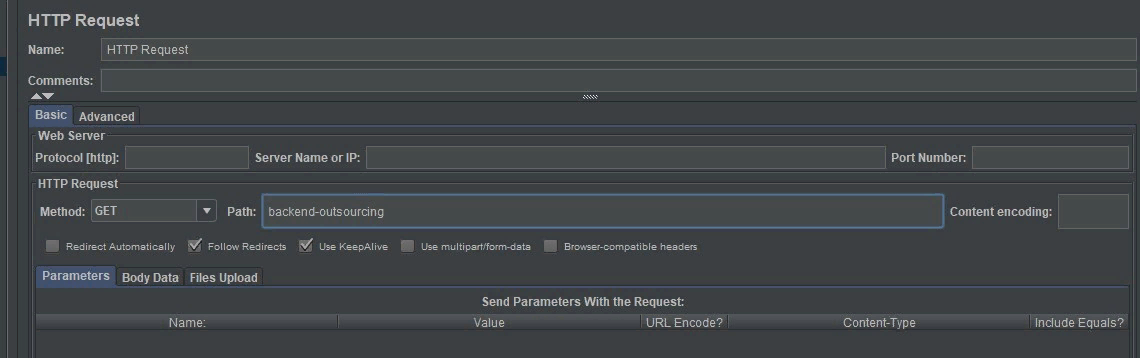
В панели управления HTTP-запроса по умолчанию введите имя тестируемого веб-сайта.В поле server name or IP указать адрес сайта.

****

* HTTP Request Этот элемент можно добавить, щелкнув правой кнопкой мыши Thread Group и выбрать : Add -> Sampler -> HTTP Request.



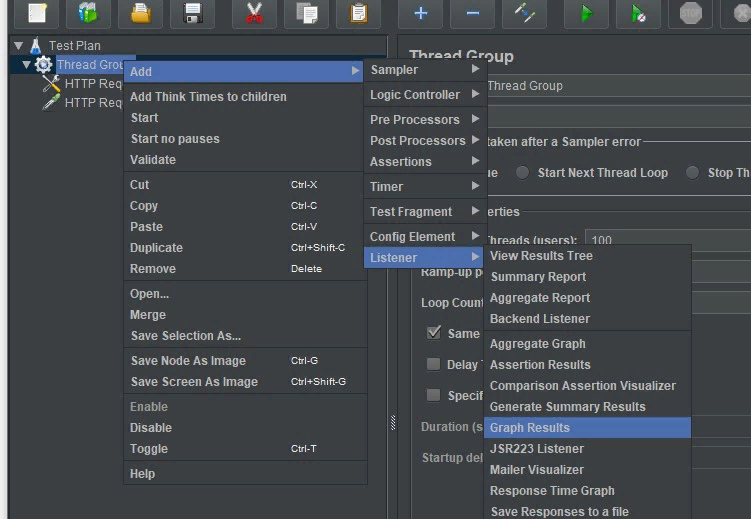
В панели управления HTTP-запросами поле «Path» указываете, какой URL-запрос вы хотите отправить на сервер сайта. https://google.com

****

Например, если вы введете «calendar» в поле «Path». JMeter создаст URL-запрос https://google.com/calendar/ к серверу google.com .

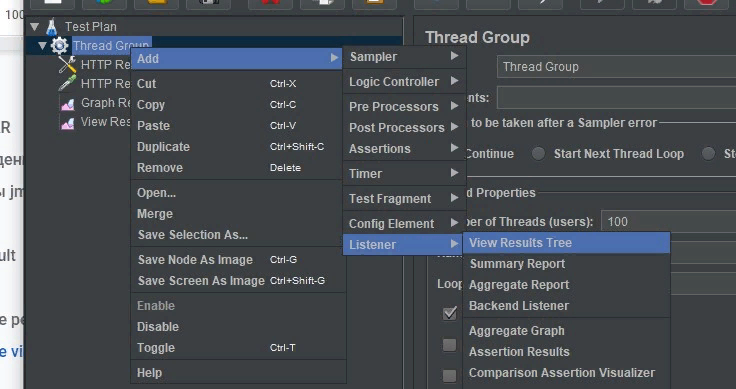
## Шаг - 3 : Отчётность.

JMeter может отображать результат теста в формате графика. Щелкните правой кнопкой мыши Test Plan, Add -> Listener -> Graph Results.



## Шаг - 4 : Запуск и анализ результатов.

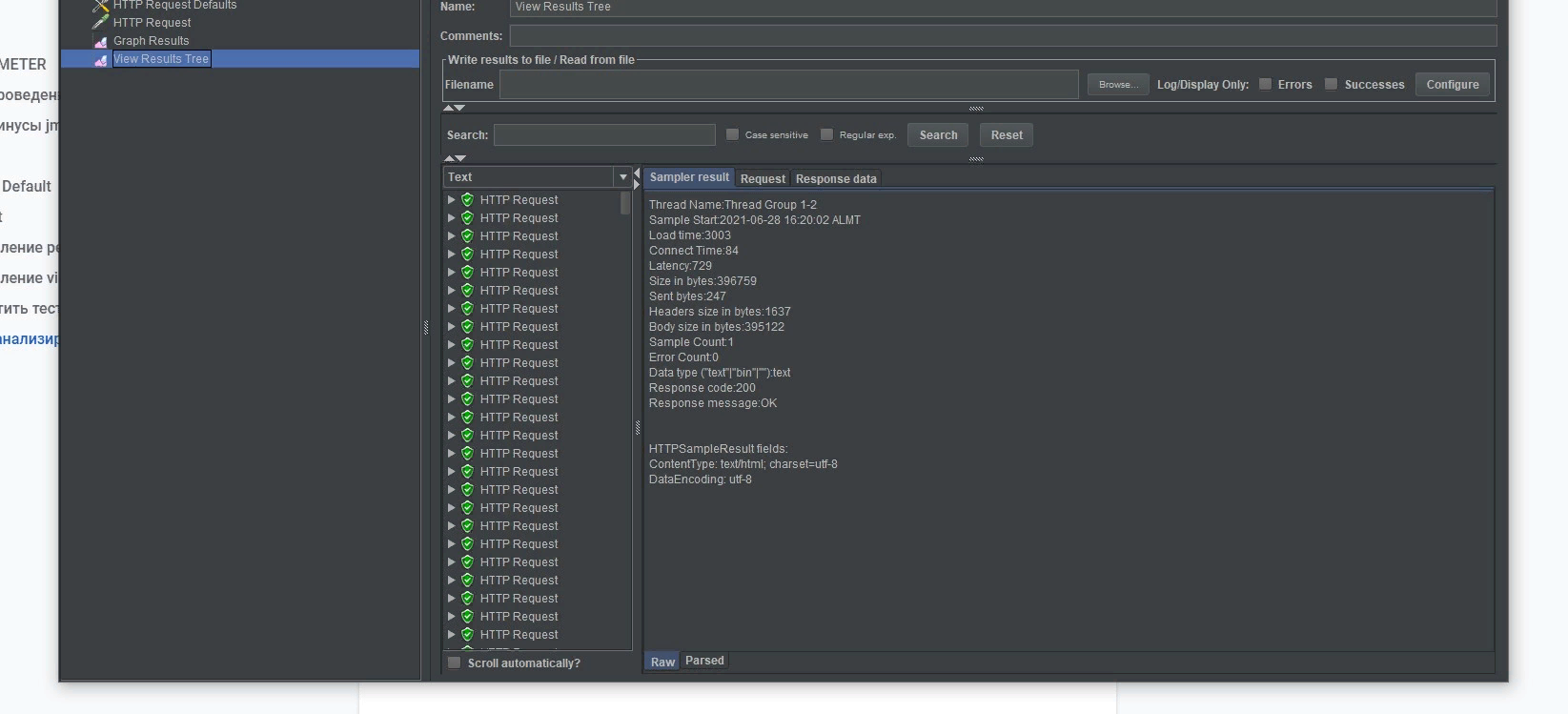
## JMeter может отображать результат теста в формате таблиц. Щелкните правой кнопкой мыши Test Plan, Add -> Listener -> View Result Tree. View result tree - это, инструмент для отладки отправленных запросов и полученных ответов.

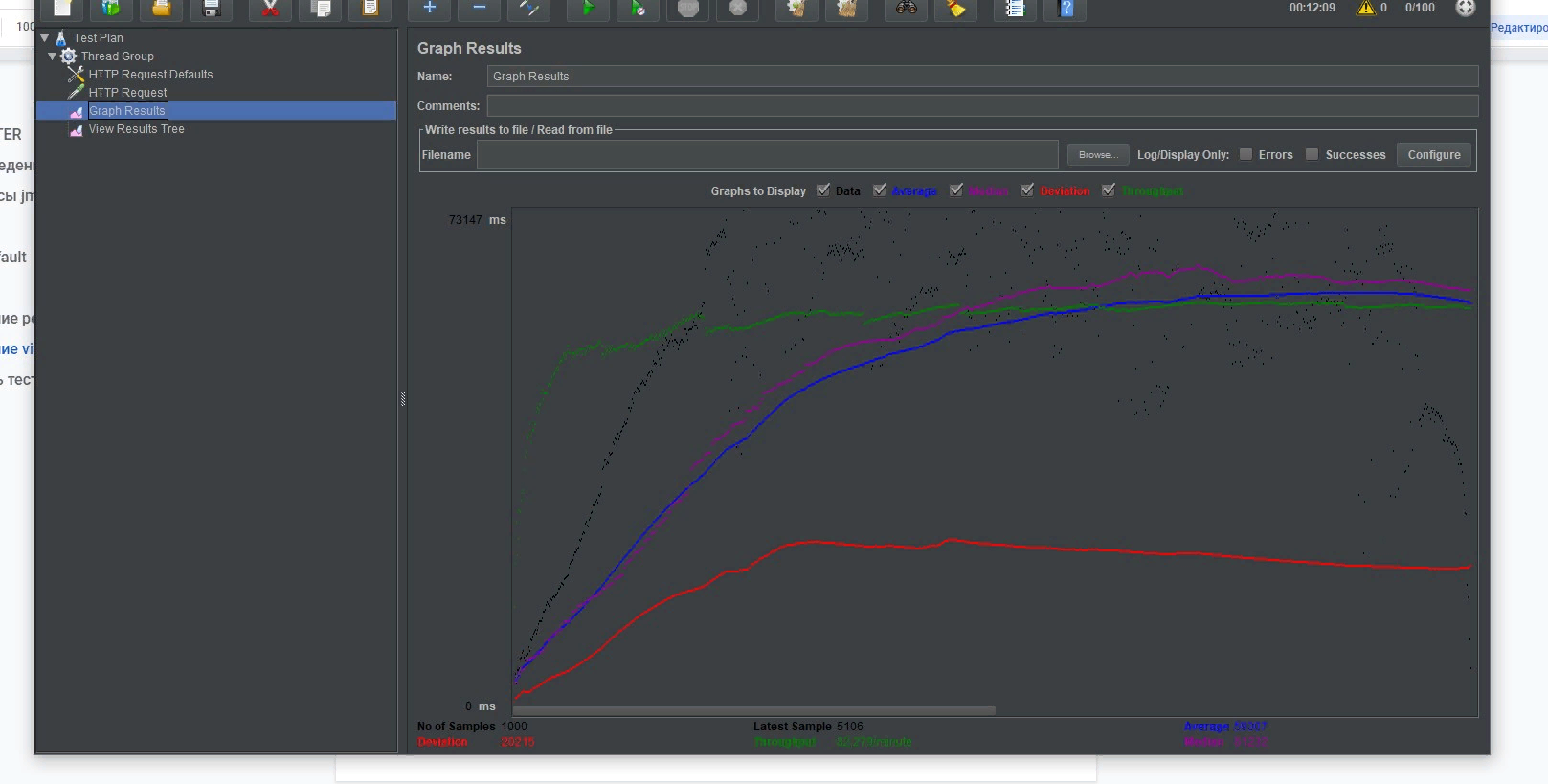
****

Запустите процесс тестирования.

Вы увидите отображение результатов теста на графике в реальном времени.

На рисунке ниже представлен график плана тестирования, где смоделировали 100 пользователей, которые заходили на сайт. Ниже указаны весь список запросов, всех пользователей, каждый запрос можно детальнее изучить.

****

****

Внизу изображения представлены следующие статистические данные, представленные цветами:

- Черный : общее количество отправленных текущих образцов.

- Синий : текущее среднее значение всех отправленных образцов.

- Красный : текущее стандартное отклонение.

- Зеленый : пропускная способность, которая представляет количество запросов в единицу времени (секунды, минуты, часы), которые отправляются на ваш сервер во время теста.

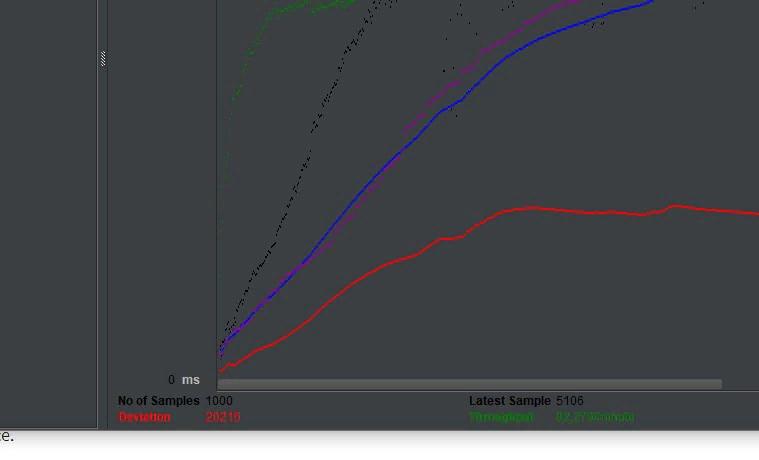
Data — время отклика каждой отдельной единицы данных т.е. каждого проверенного url.

Average — усредненное время отклика, объективный график изменения нагрузки.

Median — значение медианы Deviation — погрешность, стандартное отклонение.

Throughput — пропускная способность выполняемых запросов.

Давайте проанализируем производительность сервера сайта на рисунке ниже.



Для анализа производительности тестируемого веб-сервера следует ориентироваться на 2 параметра.

* Throughput (Пропускная способность)
* Deviation (Отклонение)

В этом тесте пропускная способность сервера сайта составляет 82279 в минуту. Это означает, что сервер сайта может обрабатывать 82279 запросов в минуту. Хорошая или плохая производительность сервера уже зависит от требования к приложению.

**Самостоятельно:**

1. Протестируйте еще один сайт (на ваш выбор) и проведите анализ полученных результатов.
2. Составьте отчет по проделанной работе