

10 шагов для запуска тестирования производительности с Apache JMeter.

Давайте представим, что Вы — единственный тестировщик на проекте, а то и во всей компании. Компания подписала контракт на разработку продукта, в котором очень важна производительность, а у Вас нет ни малейшего представления о том, с чего начать. Инструментов представлено большое количество, и выбрать какой-либо не так уж просто. Сейчас я предлагаю ознакомиться с [Apache JMeter](#) и запустить первый тест за 10 простых шагов. Предварительно нужно убедиться что установлена Java 8 или Java 9 на устройстве, с которого будут запускаться тесты.

1. Скачиваем JMeter с сайта jmeter.apache.org в нужном формате

Apache JMeter 4.0 (Requires Java 8 or 9.)

Binaries

[apache-jmeter-4.0.tgz md5 sha512 pgp](#)
[apache-jmeter-4.0.zip md5 sha512 pgp](#)

Source

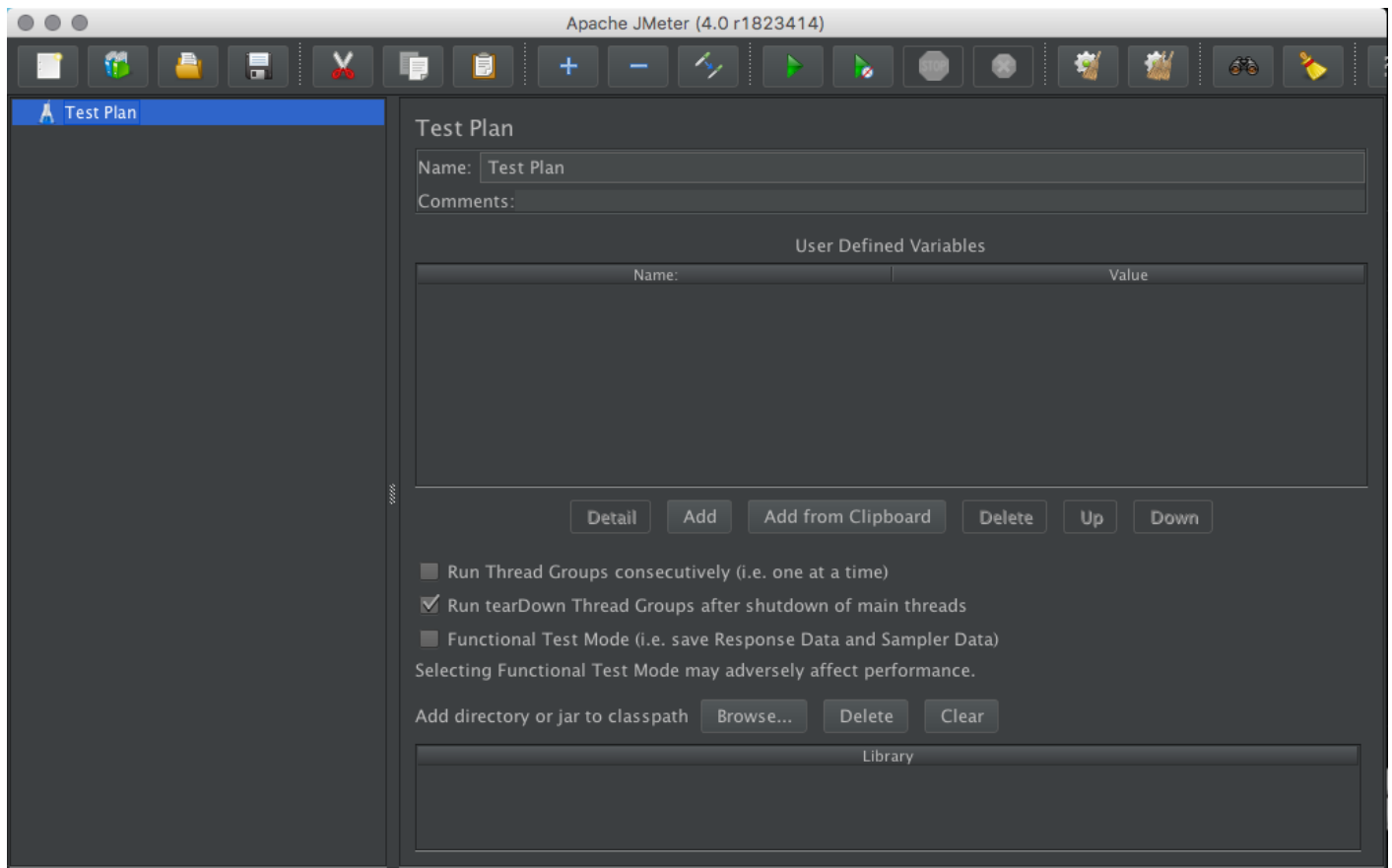
[apache-jmeter-4.0_src.tgz md5 sha512 pgp](#)
[apache-jmeter-4.0_src.zip md5 sha512 pgp](#)

Archives

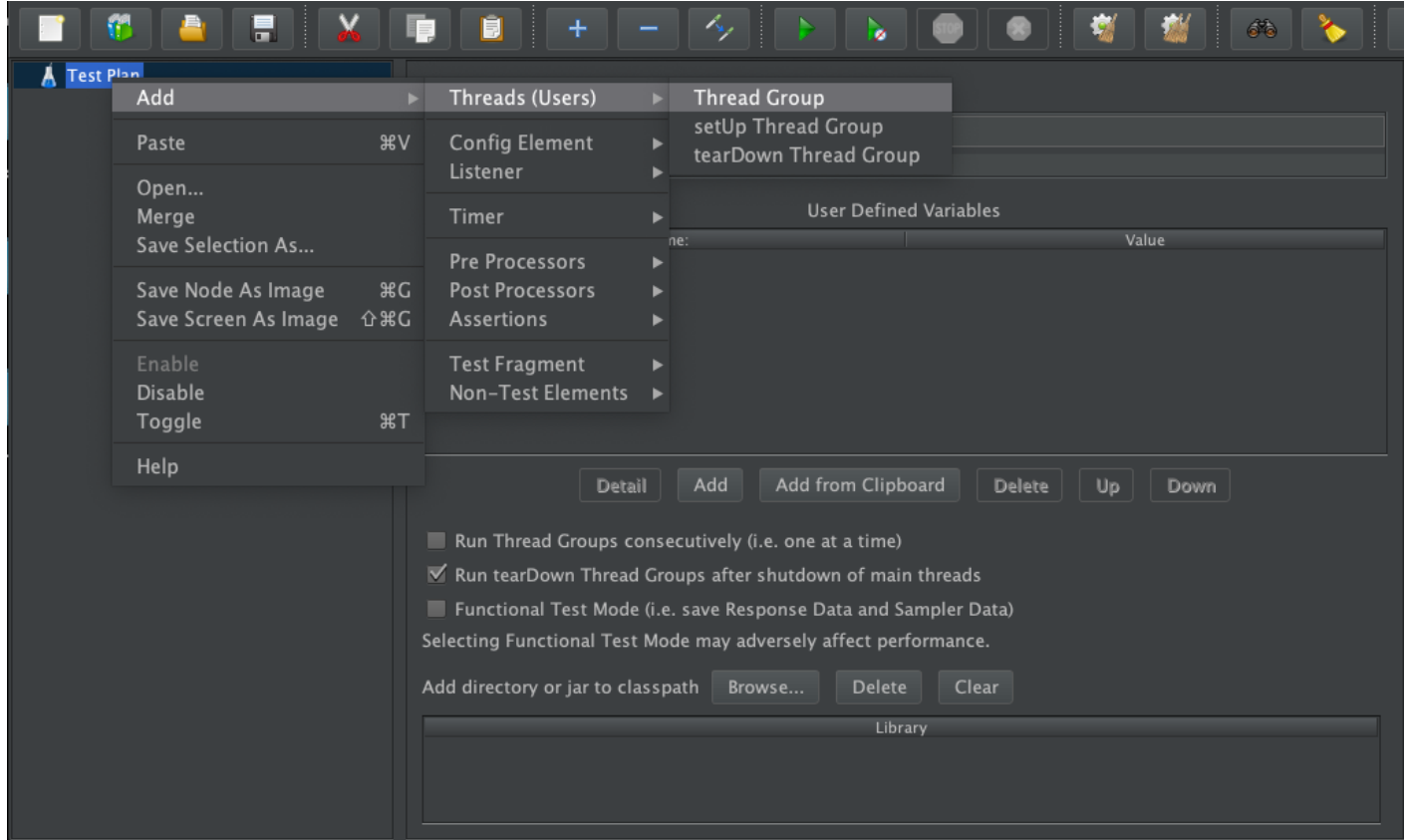
Older releases can be obtained from the archives.

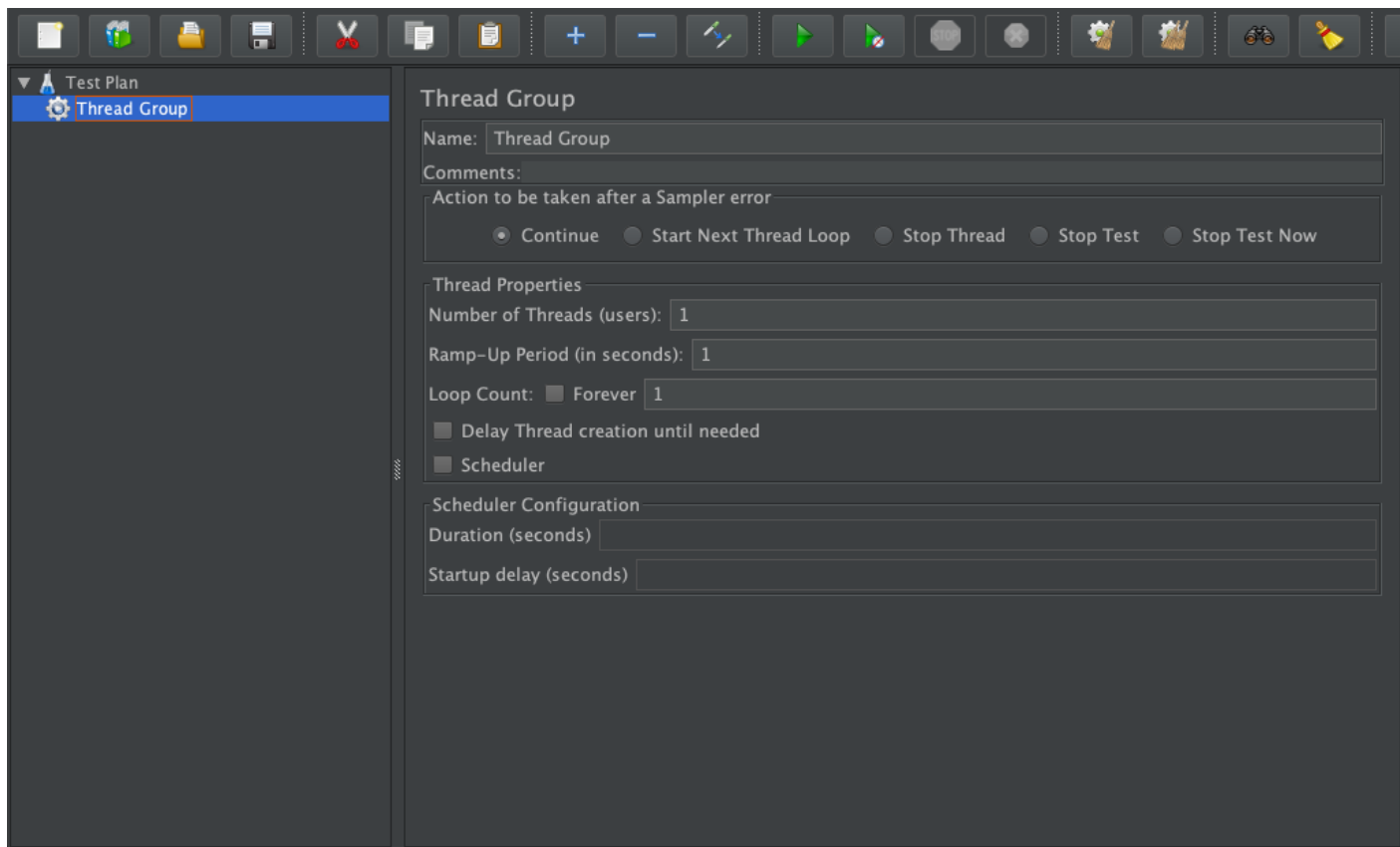
- [browse download area](#)
- [Apache JMeter archives...](#)
- [Apache Jakarta JMeter archives...](#)

2. Распаковываем архив. Путь к распакованной папке не должен содержать кириллицу или пробелы. Открываем распакованную папку, заходим в папку **/bin**, там находим сам **JMeter** (jmeter.bat для Windows jmeter.sh для Unix) и запускаем его. В результате будет представлен такой интерфейс:



3. Далее нам нужно добавить пользователей, которые будут ходить на страничку. Для этого снова выбираем тест план правым кликом и далее следуем в /Add/Threads (Users)/Thread Group. В итоге, в тест плане появляется следующий элемент:

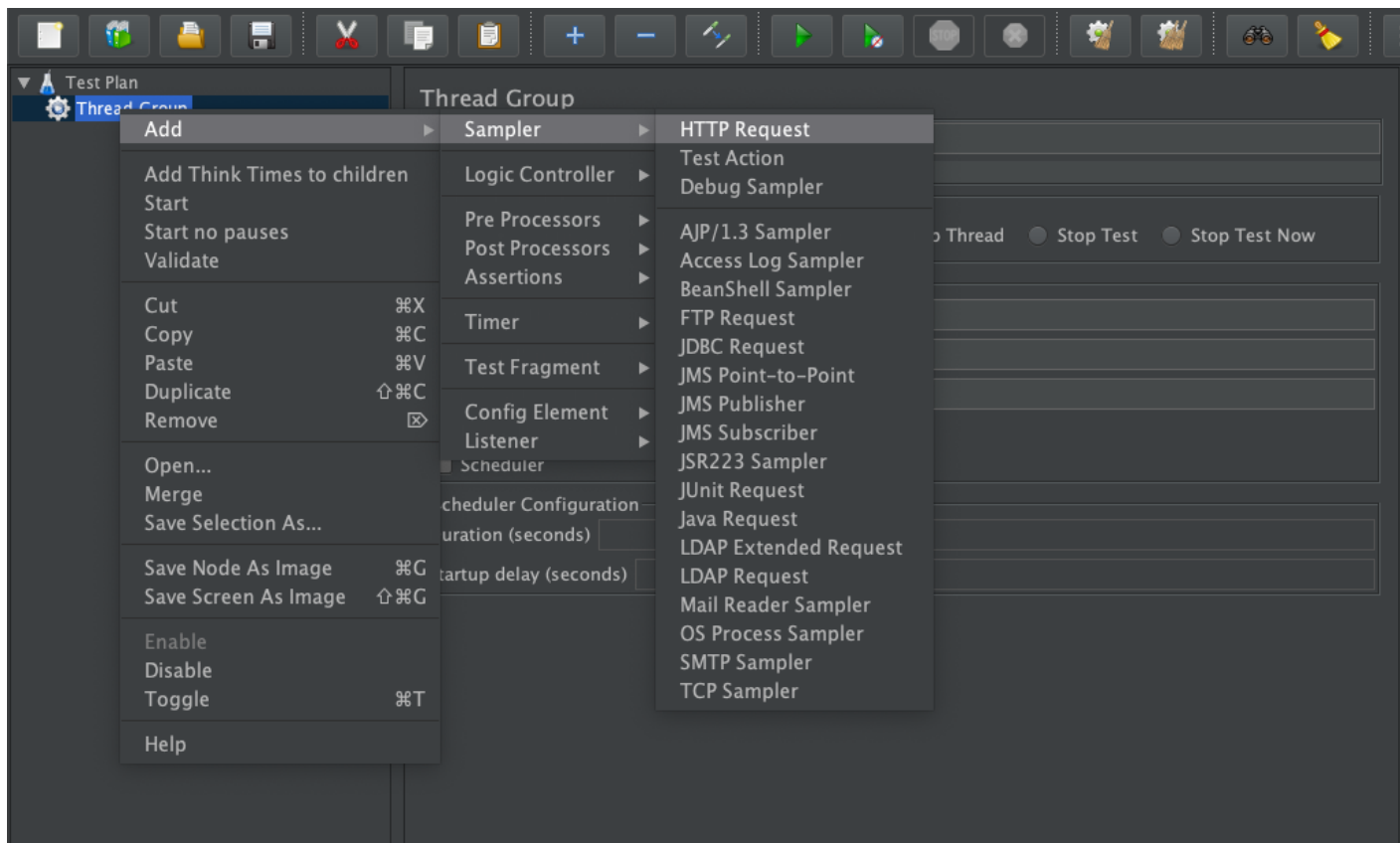




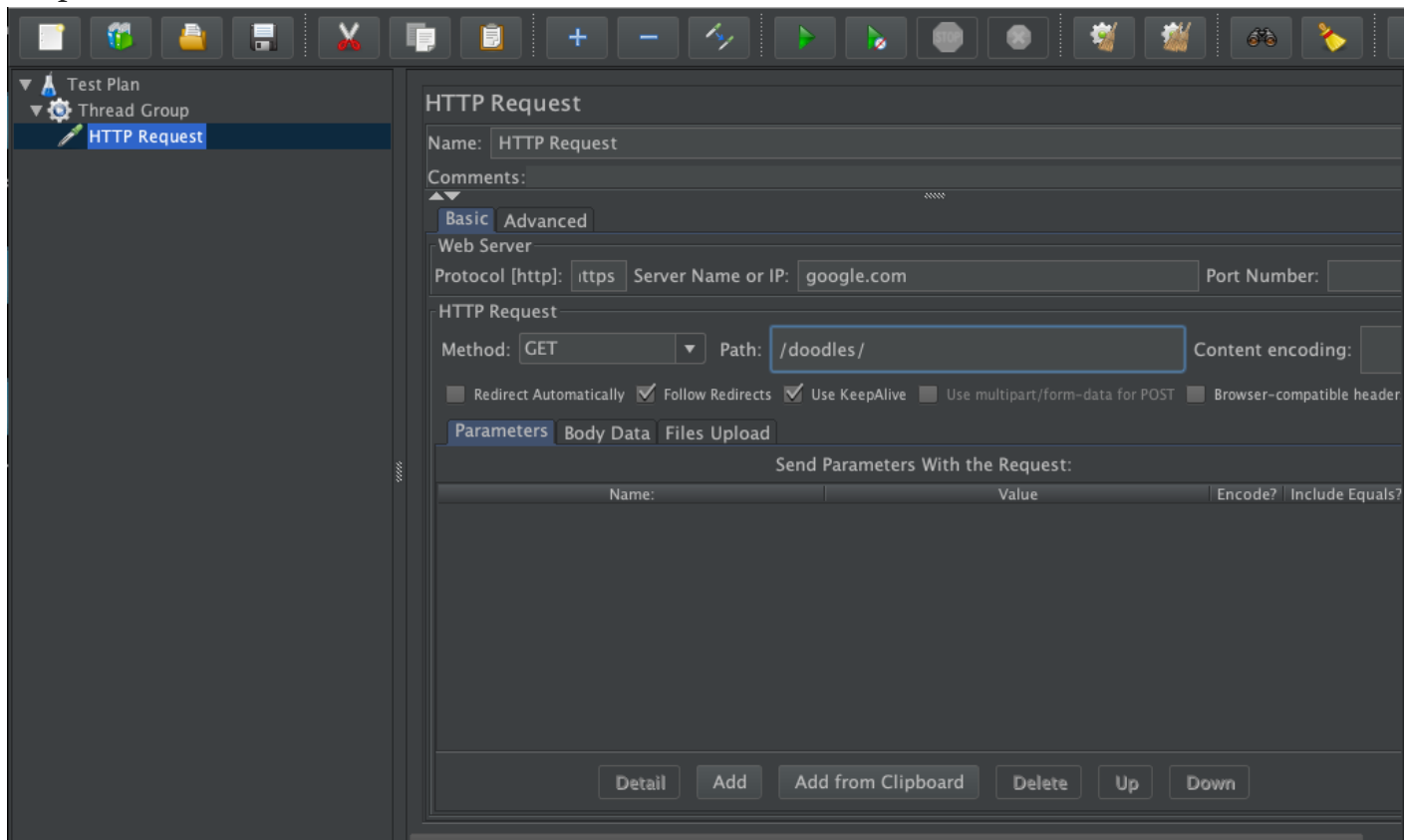
- Number of Threads — число пользователей, которые будут обращаться по запросу, указанному в HTTP Request Defaults.
- Ramp-Up Period — время, в течение которого будут прибавляться юзеры. Позволяет делать плавный и прогнозируемый старт.
- Loop Count — количество итераций. На скриншоте выше значение равно 1, то есть каждый пользователь сделает всего 1 запрос.

Дополнение: Можно выставить базовые параметры, указанные выше на любые значения, которые хочется испытать, но для начала рекомендую обойтись небольшими величинами, например, 10 пользователей и пару циклов. Выставление огромного количества пользователей в одном треде может проглотить все мощности вашего устройства и оно просто перезагрузится :)

4. Для запуска теста нужно указать инструменту адрес сервера, который будет подвержен нагрузке с типом запроса. Это делается через правый клик по уже добавленному Thread Group и выбор Add/Sampler/HTTP Request



5. После добавления элемента можно указывать адрес сервера. Прежде чем ходить на чьи-то реальные/тестовые серверы и наводить панику на команды разработки и поддержки, стоит попробовать свои силы на простейшем примере поисковой системы. В данном случае я предлагаю сходить на [google.com](https://www.google.com/doodles/), а точнее, на страницу, так называемого, “Мне повезёт!” (<https://www.google.com/doodles/>). Для этого в протоколе указываем **https**, в адресе сервера **google.com**, а в пути расположение страницы **/doodles/**, ну и не забываем про тип запроса **GET**.



6. Тест нужно сохранить в папку **/bin** (там же, где лежит сам JMeter), далее, запустить терминал/командную строку и перейти в папку **/bin**, из которой на втором шаге запускали JMeter. Графическую оболочку лучше закрыть перед запуском теста. Запуск теста осуществляется командой:

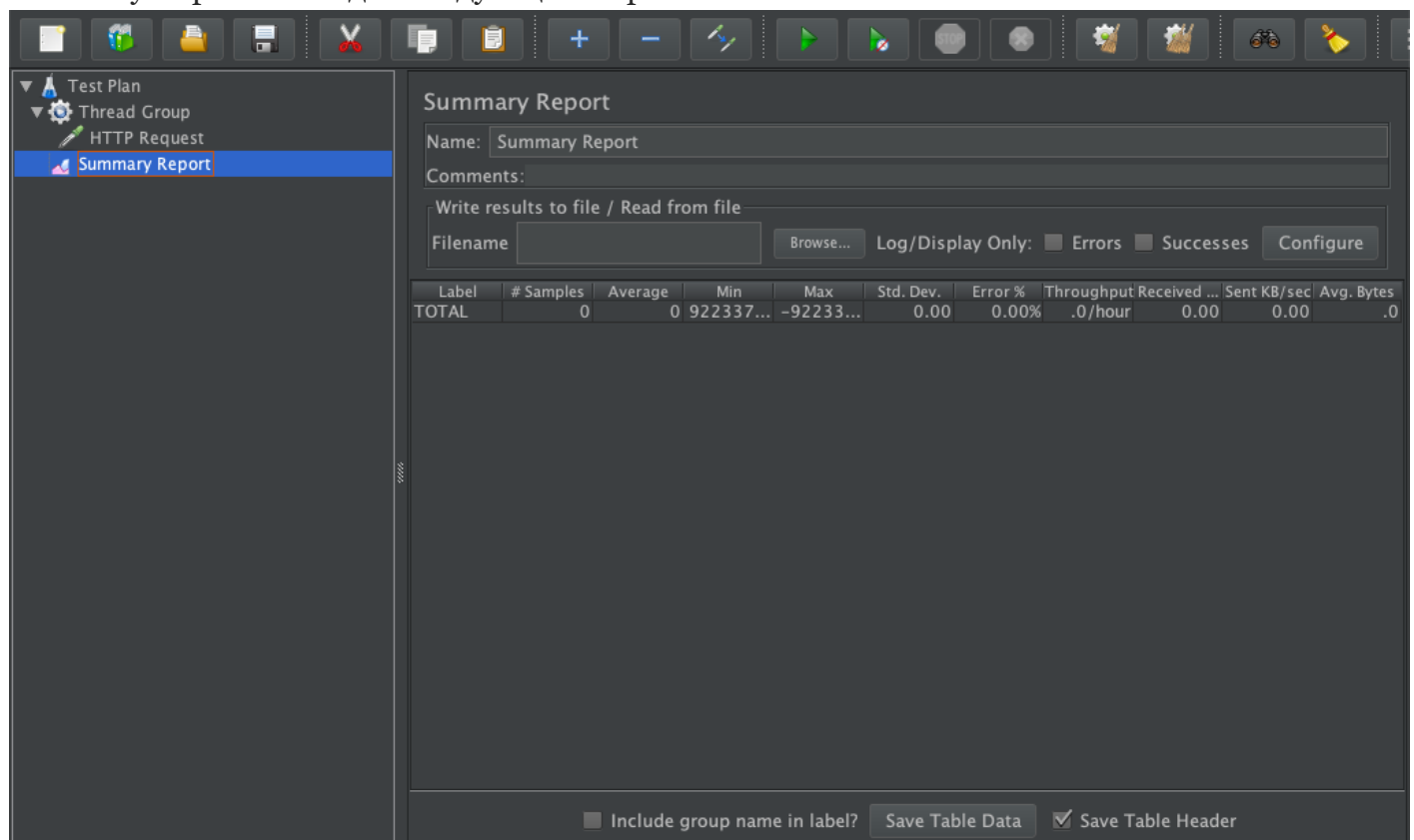
- `jmeter -n -t test_name.jmx -l log.jtl` (Windows)
- `sh jmeter.sh -n -t test_name.jmx -l log.jtl` (Unix)

В примерах команд выше **test_name.jmx** — имя сохранённого теста, а **log.jtl** — имя файла, в который будут сохранены результаты теста. Ключ **-n** нужен для запуска инструмента в режиме без интерфейса (чтобы повысить производительность), ключ **-t** указывается перед именем сохраненного файла с тестом, ключ **-l** указывается перед именем файла, который будет создан в ходе прохождения теста и содержать отчёт.

7. Запускаем тест и ждём его завершения.

```
MacBook-Pro-Artem:bin artemmezdryn$ sh jmeter.sh -n -t doodles_test.jmx -l doodles_log.jtl
Creating summariser <summary>
[Created the tree successfully using doodles_test.jmx
Starting the test @ Mon Aug 27 11:22:50 OMST 2018 (1535347370594)
Waiting for possible Shutdown/StopTestNow/Heapdump message on port 4445
summary =      20 in 00:00:04 =      5.4/s Avg:  1388 Min:   694 Max:  2396 Err:      0 (0.00%)
Tidying up ...   @ Mon Aug 27 11:22:54 OMST 2018 (1535347374895)
... end of run
```

8. Далее, можно запустить JMeter командой **jmeter** (Windows) или **sh jmeter.sh** (Unix) отсюда же, из терминала/командной строки, или по старинке из папки **/bin**. После запуска нужно открыть сохранённый тест (в моём случае это `doodles_test.jmx`) и для него добавить отчёты. Самый простой отчёт — Summary Report, который добавляется через правый клик по Test Plan и, далее, Add/Listener/Summary Report. Listener'ы можно добавлять и в ходе настройки теста. Итак, Summary Report выглядит следующим образом:



The screenshot shows the JMeter Summary Report window. The left sidebar displays the Test Plan tree with 'Summary Report' selected. The main window shows the 'Summary Report' configuration, including fields for Name, Comments, and a section for writing results to a file. Below this is a table of test results.

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput Received ...	Sent KB/sec	Avg. Bytes
TOTAL	0	0	922337...	-92233...	0.00	0.00%	.0/hour	0.00	0.00

At the bottom of the window, there are checkboxes for 'Include group name in label?' and 'Save Table Header', and a 'Save Table Data' button.

9. Чтобы посмотреть результаты теста, нужно нажать Browse..., найти файл с логами в формате *.jtl, который был прописан в шаге 6, и открыть его. В результате будет следующее:

Summary Report

Name: Summary Report

Comments:

Write results to file / Read from file

Filename: meter-4.0/bin/doodles.jtl Log/Display Only: ☐ Errors ☐ Successes

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received ...	Sent KB/sec	Avg. Bytes
HTTP Re...	20	1084	638	1462	264.74	0.00%	6.4/sec	1414.92	1.56	225445.8
TOTAL	20	1084	638	1462	264.74	0.00%	6.4/sec	1414.92	1.56	225445.8

☐ Include group name in label? ☒ Save Table Header

10. Что можно понять из данного отчёта?

- Было сделано 20 запросов (Samples) по указанному в конфиге адресу (10 пользователей по 2 раза каждый)
- Среднее время ответа составило чуть меньше чем 1,1 секунды (Average); минимальное время ответа чуть менее чем 0,7 секунды (Min); максимальное время ответа чуть менее чем 1,5 секунды (Max).
- В секунду проходило 6,4 запроса (Throughput).
- Std Dev — показатель стандартного отклонения. Насколько я знаю, им мало кто пользуется, но, раз уж он есть, то... Этот показатель позволяет оценить насколько сильно значения из выборки (результата тестового прогона) отличаются от рассчитанного среднего значения.
- Error %—количество ошибок в процентах, которые вернул сервер
- Received и Sent KB/sec — количество полученных и отправленных данных
- Avg.Bytes — среднее количество полученных данных