

Performance Testing with Apache Jmeter

Introduction

Автоматизированное тестирование:

- Замена ручного процесса средствами автоматизации для тестирования ПО
- Позволяет повторно запустить сценарии (регрессия)
- Также применяется для тестирования производительности, тестирования под нагрузкой, стресс-тестирования

Достоинства:

- Увеличение тестового покрытия
- Повышение точности
- Экономия времени

When to Automate?

- Длинные и критические проекты
- Проекты, требующие тестирования одних и тех же областей
- Приложение использует много пользователей
- ПО, используемое для автоматического тестирования, является стабильным по отношению к ручному тестированию
- Сжатые сроки тестирования

How to Automate?

Существует много доступных инструментов для тестирования:

- **Jmeter**
- Selenium
- TestComplete

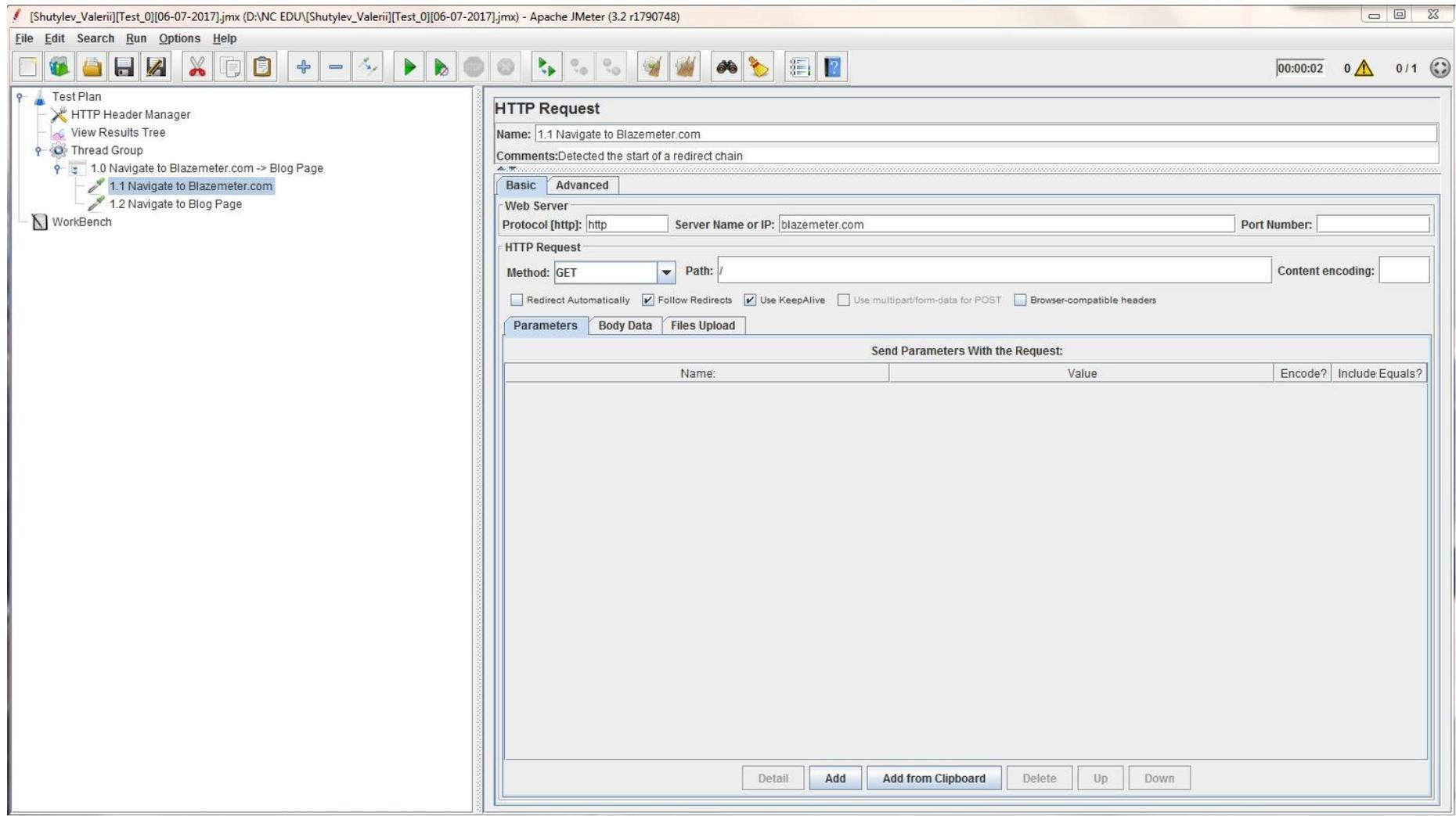
- Load Runner
- WATIR
- Visual Studio Test Professional

Introduction - ™

- Может проводить загрузку и тест производительности для различных типов серверов - Web - HTTP, HTTPS, SOAP и т.д.
- Также используется для тестирования производительности Java-приложений, скриптов CGI, Java объектов, баз данных
- Jmeter может быть использован для имитации большой нагрузки на сервере, сети или объекте для испытания его на прочность, либо для анализа
- Jmeter позволяет создавать тестовые скрипты с валидацией правильности возврата ожидаемого результата (с помощью регулярных выражений).

**** Jmeter не является браузером!!!**

Example - ™



The screenshot shows the Apache JMeter GUI. The left sidebar displays a test plan tree with the following structure:

- Test Plan
 - HTTP Header Manager
 - View Results Tree
 - Thread Group
 - 1.0 Navigate to Blazemeter.com -> Blog Page
 - 1.1 Navigate to Blazemeter.com**
 - 1.2 Navigate to Blog Page
- WorkBench

The main window displays the configuration for the selected element, "1.1 Navigate to Blazemeter.com", under the "HTTP Request" tab. The configuration includes:

- Name:** 1.1 Navigate to Blazemeter.com
- Comments:** Detected the start of a redirect chain
- Basic Tab:**
 - Web Server:** Protocol [http]: http, Server Name or IP: blazemeter.com, Port Number: (empty)
 - HTTP Request:** Method: GET, Path: /, Content encoding: (empty)
 - Options: Redirect Automatically, Follow Redirects, Use KeepAlive, Use multipart/form-data for POST, Browser-compatible headers
- Parameters Tab:** A table for "Send Parameters With the Request" is currently empty.

Work start...

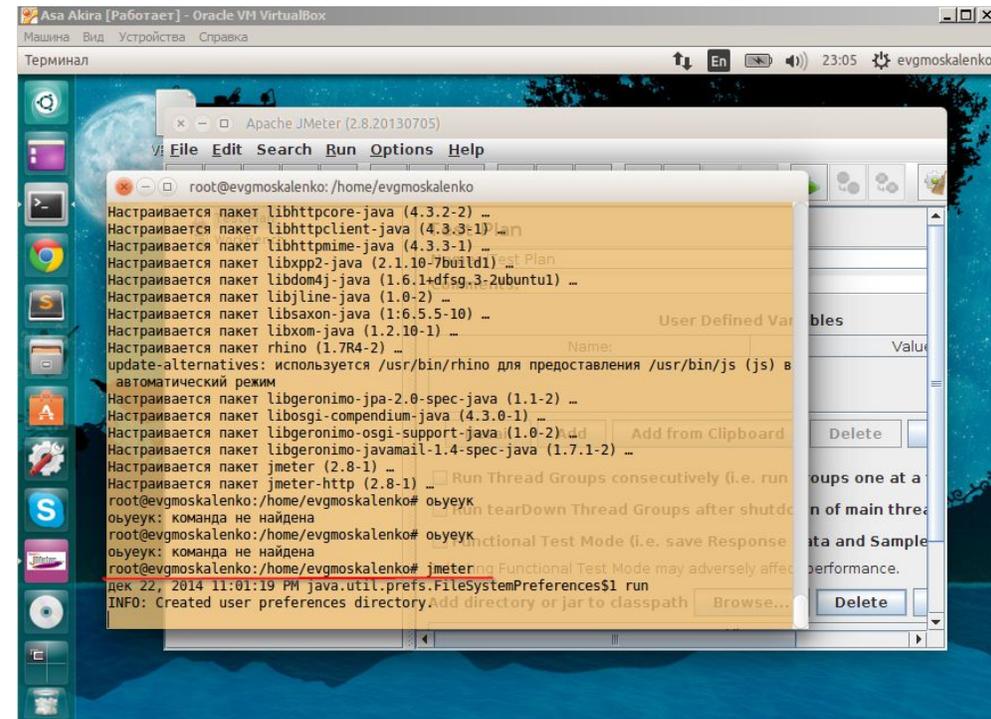
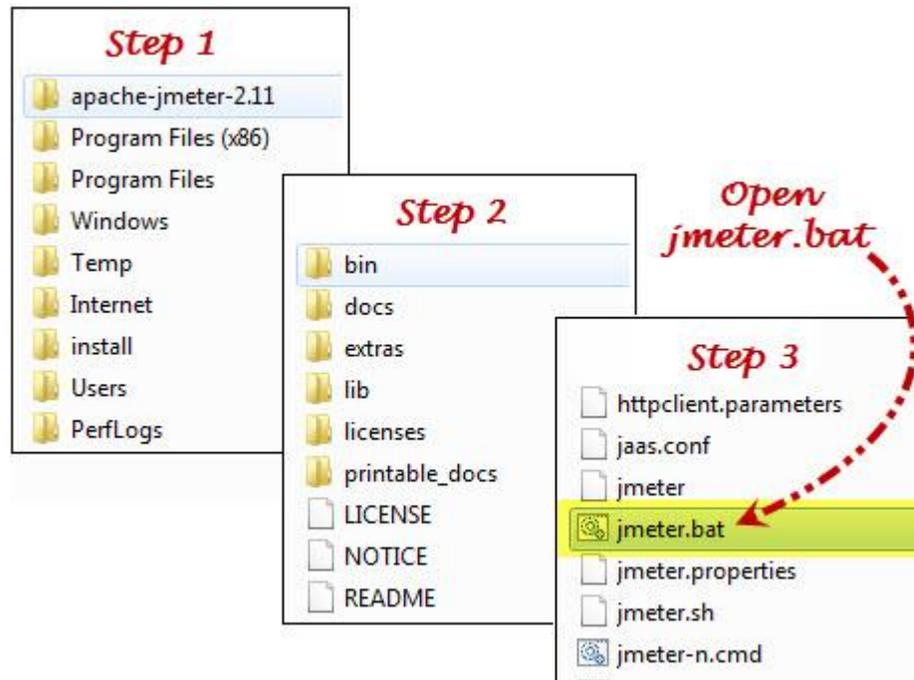
...с чего начать?

- Скачайте последнюю версию <http://jmeter.apache.org>
- Разархивируйте ZIP архив в любом удобном месте

Jmeter running...

Для запуска Jmeter необходимо запустить:

- jmeter.bat (Windows)
- jmeter.sh (Unix)



Building a basic Test Plan...

- 5 пользователей «навигируются» на 2 разные страницы сайта.
- Тест запускается 2 раза
- Каждая навигация – HTTP запрос.
- Общее количество HTTP запросов в тесте :
5 пользователей * 2 запроса * повтор 2 раза = 20
- Использоваться будут элементы:
Thread Group, HTTP Request, HTTP Request Defaults.

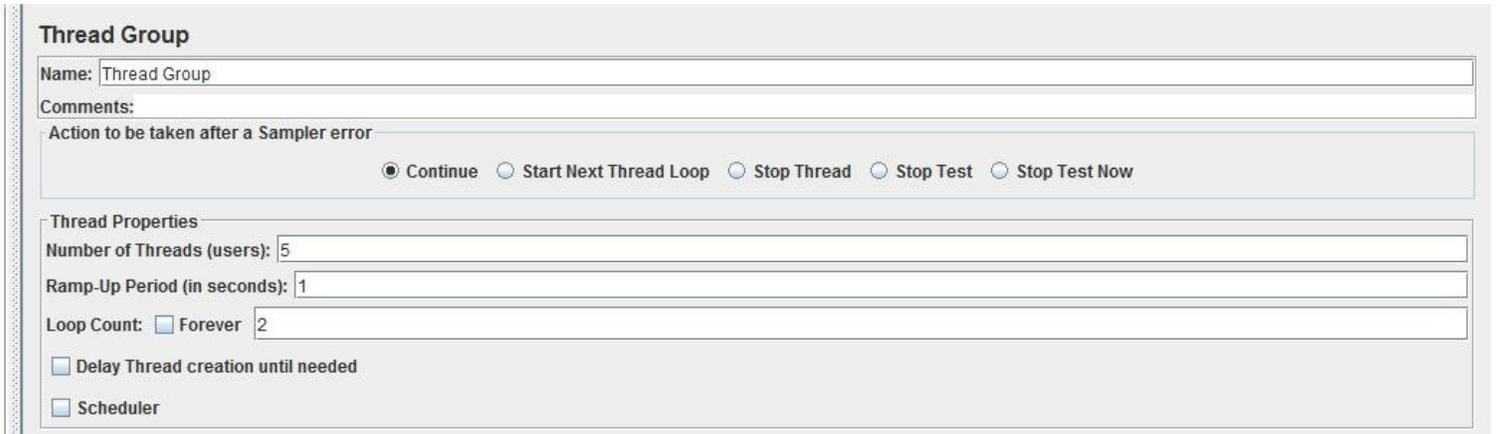
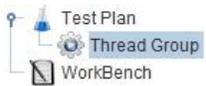
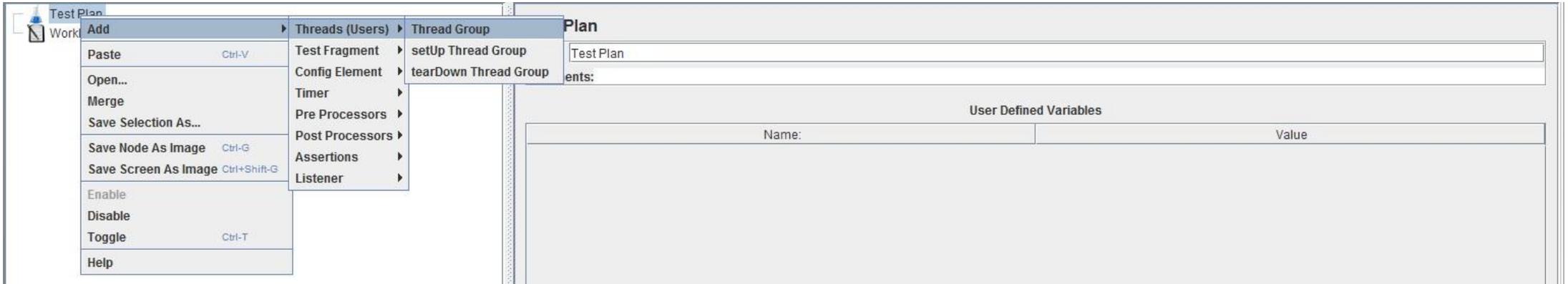
**Another Elements of a test plan:*

<http://jmeter.apache.org/usermanual/index.html>

Thread Group

- Добавляем Thread Group: **Add** **Thread Group**
- Устанавливаем количество пользователей (**Threads**): 5
- **Ramp-Up Period**: задает Jmeter задержку между стартом каждого пользователя
 - Если имеется 5 пользователей и 5 секунд Ramp-Up Period, то задержка между пользователями составит 1 секунда ($5 \text{ users} / 5 \text{ seconds} = 1 \text{ user per second}$)
- **Loop Count** (сколько раз повторять тест): устанавливаем значение 2

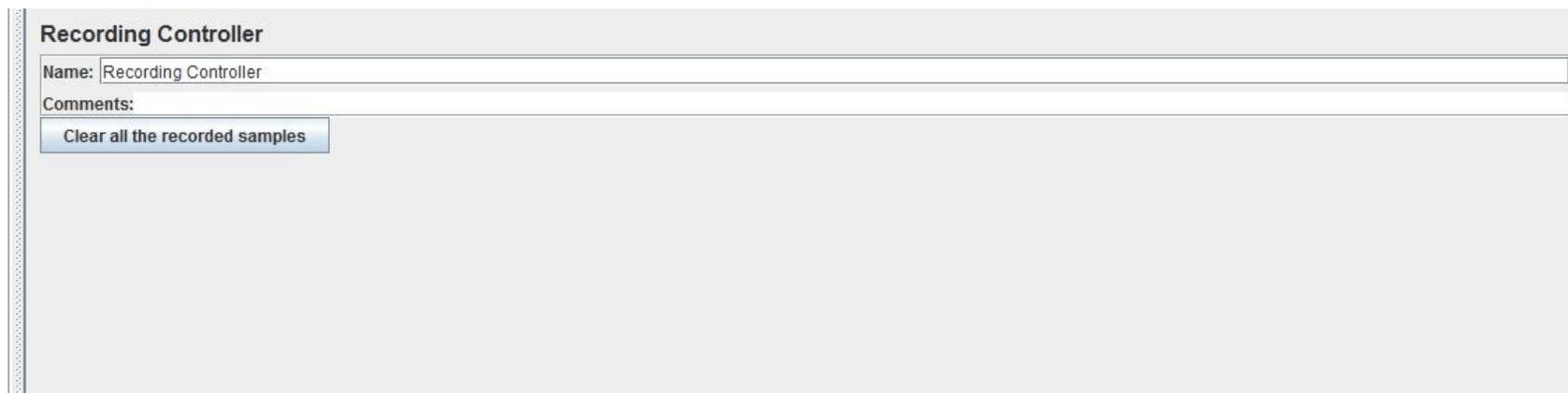
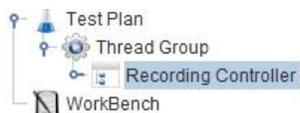
Example: Thread Group



Recording Controller

- Добавляем Thread Group: **Add menu, выбираем**
Add **Logic Controllers** **Recording Controller**

Необходим в качестве семпла для записи сценария с использованием **HTTP(S) Test Script Recorder**



HTTP(S) Test Script Recorder

- Данный элемент позволяет Jmeter записывать действия пользователя (реквесты) во время навигации по веб-страницам
- Добавляется HTTP Proxy Server элемент на панели **Workbench**
Add **Non-Test Elements** **HTTP(S) Test Script Recorder**

HTTP(S) Test Script Recorder

Настройка HTTP(S) Test Script Recorder:

- Указать Порт для Proxy Server (по умолчанию 8888)
- Указать в какой контроллер добавлять записанные реквесты (по умолчанию Recording Controller)
- Указать шаблон для исключения определенных типов файлов (png, js, CSS и т.д.)

Example: HTTP(S) Test Script Recorder



HTTP(S) Test Script Recorder

Name: HTTP(S) Test Script Recorder

Comments:

State

Global Settings

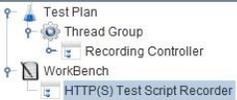
Port: 8888 HTTPS Domains :

Test Plan Creation Requests Filtering

Test plan content

Target Controller: Use Recording Controller

Grouping: Do not group samplers Capture HTTP Headers Add Assertions Regex matching



HTTP(S) Test Script Recorder

Name: HTTP(S) Test Script Recorder

Comments:

State

Global Settings

Port: 8888 HTTPS Domains :

Test Plan Creation Requests Filtering

Content-type filter

Include: Exclude:

URL Patterns to Include

URL Patterns to Include

URL Patterns to Exclude

URL Patterns to Exclude

View Result Tree

View Result Tree показывает дерево всех ответов, позволяя просмотреть ответ для любого семпла.

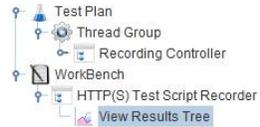
В дополнение к отображению ответа можно определить время, затраченное для получения ответа, и некоторые коды ответов.

Используется для отладки Сценария.

Чтобы добавить View Result Tree, необходимо выполнить

Add Listener View Result Tree

Example: View Result Tree



View Results Tree

Name: View Results Tree

Comments:

Write results to file / Read from file

Filename: Browse... Log/Display Only: Errors Successes

Search: Case sensitive Regular exp.

Browser

- ✘ ajax.googleapis.com:443
- ✘ cdn.optimizely.com:443
- ✘ cdnjs.cloudflare.com:443
- ✘ cdnjs.cloudflare.com:443
- ✘ ajax.googleapis.com:443
- ✔ 103 /success.bt
- ✘ cdn.mxpnl.com:443
- ✘ www.googletagmanager.com:443
- ✘ cdnjs.cloudflare.com:443
- ✘ cdnjs.cloudflare.com:443
- ✔ 104 /success.bt
- ✘ blazemeter.disqus.com:443
- ✘ clients6.google.com:443
- ✔ 105 /success.bt
- ✔ 106 /blog
- ✘ cdn.optimizely.com:443
- ✘ ajax.googleapis.com:443
- ✔ 113 /success.bt
- ✔ [107 /sites/default/files/css/css_IC8ne_1VSMmxYmTUMDo
- ✔ [109 /sites/default/files/js/js_NoHwYsnFrp1qxmixFmQ3E-
- ✔ [110 /sites/all/themes/blazeng/js/lib/typed.js]
- ✔ [111 /sites/all/themes/blazeng/js/additional_scripts/test_wiz
- ✔ [112 /sites/all/themes/blazeng/js/blogs.js]
- ✔ [108 /sites/default/files/js/js_nXtu-J7naHckvN-ay72sN03ne9
- ✘ cdn.mxpnl.com:443
- ✔ 120 /success.bt
- ✘ www.googletagmanager.com:443
- ✔ [115 /new_images/new_blog_cross.png]
- ✔ 121 /success.bt
- ✔ [118 /new_images/new_blog/bounce.png]
- ✔ 116 /sites/default/files/styles/thumbnail/public/default_imag
- ✔ 119 /sites/default/files/styles/blog_thumb/public/Untitled%2
- ✔ [122 /new_images/new_blog/small_white_blazemeter_log
- ✔ [124 /sites/default/files/alon_3_0.png]
- ✘ www.youtube.com:443
- ✔ [125 /sites/default/files/sage_3_0.png]

100%

FEATURES PRICING RESOURCES BLOG REQUEST A DEMO SIGN IN

JMETER · Jul 11 2017

Running Your Load Tests With Pass Fail Criteria - A Guide

Learn how to easily check if your tests passed or failed. Read this guide to set your thresholds in Taurus and in Jenkins and automatically run your tests.

[Read more...](#)

Guy Salton

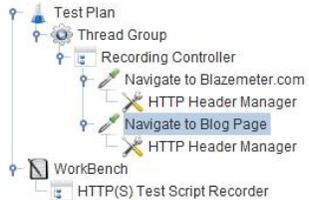
Search: Case sensitive Regular exp.

Scroll automatically?

HTTP Request

HTTP Request позволяет отправлять HTTP / HTTPS-запрос на веб-сервер, имитируя запрос пользователя на страницу с целевого сервера.

Также позволяет контролировать, обрабатывает ли JMeter файлы HTML и отправляет HTTP-запросы для их получения.



The screenshot shows the 'HTTP Request' configuration dialog box in JMeter. The 'Name' field is set to 'Navigate to Blog Page'. The 'Basic' tab is selected, showing the following configuration:

- Web Server:** Protocol [http]: https, Server Name or IP: www.blazemeter.com, Port Number: 443
- HTTP Request:** Method: GET, Path: /blog, Content encoding: (empty)
- Options:** Redirect Automatically, Follow Redirects, Use KeepAlive, Use multipart/form-data for POST, Browser-compatible headers

The 'Parameters' tab is also visible, showing a table for 'Send Parameters With the Request':

| Name: | Value | Encode? | Include Equals? |
|-------|-------|---------|-----------------|
| | | | |

Test plan is ready! Run!

The screenshot displays the Apache JMeter 3.2 user interface. The title bar indicates the file path: [Shutylev_Valerij][Test_0][06-07-2017].jmx (D:\NC EDU\Shutylev_Valerij)[Test_0][06-07-2017].jmx - Apache JMeter (3.2 r1790748). The menu bar includes File, Edit, Search, Run, Options, and Help. The toolbar contains various icons for file operations and execution. The status bar shows a timer at 00:00:12, 0 warnings, and 0/5 tests.

The main window is divided into several sections:

- Test Plan:** A tree view on the left showing the test plan structure:
 - Test Plan
 - View Results Tree
 - Thread Group
 - Recording Controller
 - Navigate to Blazemeter.com
 - Navigate to Blog Page
 - WorkBench
 - HTTP(S) Test Script Recorder
- View Results Tree:** A panel on the right showing the execution results. It includes a Name field (View Results Tree), a Comments field, and a section for writing results to a file. The filename field is empty, and there is a 'Browse...' button. The 'Log/Display Only' section has checkboxes for 'Errors' and 'Successes', and a 'Configure' button. Below this is a search bar with 'Case sensitive' and 'Regular exp.' checkboxes, and 'Search' and 'Reset' buttons.
- HTML:** A tabbed view showing the response data for the selected test element. The 'HTML' tab is active, and the content area is empty. The 'Sampler result' tab is also visible.

* Переменные

В JMeter существует возможность указать параметры запросов через переменные, определяемые самим пользователем. Для этих целей чаще всего применяется элемент «User Defined Variables» (либо в корне «Test Plan», но это плохой путь)

Синтаксис: $\${variable_name}$

Также можно определить значение переменной с помощью регулярного выражения (Regular Expression)

* Чтение из файла и запись в файл

Jmeter позволяет выполнять тест с использованием заранее подготовленного прериквизита в виде csv-файлов

Чтение файла происходит построчно с разбиением их на переменные с помощью **CSV Data Set Config**

Запись в файл производится с помощью BeanShell кода с помощью **BeanShell Sampler**

*советую создавать csv-файл в блокноте, а не в Excel

* Регулярные выражения

Регулярные выражения используются для поиска и работы с текстом, на основе шаблонов. Jmeter интерпретирует формы регулярных выражений или шаблонов и использует их на протяжении тест- плана.

Для того, чтобы использовать регулярные выражения в тест-плане, необходимо использовать постпроцессор Regular Expression Extractor. Разместить регулярные выражения можно в любом компоненте в плане тестирования.

* Example: Регулярные выражения

Необходимо получить ID продукта из ответа от сервера в виде html-страницы

```
name="Product_1" value="100500"
```

A screenshot of the 'Regular Expression Extractor' configuration dialog in JMeter. The dialog has the following fields and options:

- Name: Regular Expression Extractor
- Comments: (empty)
- Apply to: Main sample and sub-samples Main sample only Sub-samples only JMeter Variable
- Field to check: Body Body (unescaped) Body as a Document Response Headers Request Headers URL Response Code Response Message
- Reference Name: variable
- Regular Expression: value="(\\d+)"
- Template: \$1\$
- Match No. (0 for Random): 1
- Default Value: value_not_found Use empty default value

RegExp: value="(\\d+)"

* (\\d+) - числовые значения

Q&A

Thank You



Focused forward
Netcracker