

Швейцарский нож тестировщика или JMeter не только для нагрузки

контур

SQA[®]
DAYS#25

Махетов Сергей

Обо мне

- ▶ В области разработки ПО > 10 лет, тестирование > 5 лет.
- ▶ Тестирование как бизнес-приложения(Web, API, desktop), так и инфраструктурные решения(Интеграции через HTTP, JMS, JDBC, Kafka...)
- ▶ Основной стек - Java(Scala)/JMeter, Gatling, Postman



О чем поговорим

- ▶ Базовое описание Apache JMeter
- ▶ Создание функционального теста
- ▶ Дополнительные возможности
- ▶ Описание других сценариев использования
- ▶ Работа в команде

Что такое Apache JMeter



- ▶ **Open Source application designed to load test applications and measure performance. By The Apache Software Foundation**



Что такое Apache JMeter



TM

- ▶ **Open Source application designed to load test applications and measure performance. By The Apache Software Foundation**



Внешний вид

The screenshot displays the Apache JMeter 4.0 user interface. The title bar shows the file name "JMTest.jmx (D:\Преза\jmeter Пермь\JMTest.jmx) - Apache JMeter (4.0 r1823414)". The menu bar includes "File", "Edit", "Search", "Run", "Options", and "Help". The toolbar contains various icons for file operations, navigation, and execution.

The left pane shows a test plan tree structure:

- Тест-план
 - Пользовательские переменные
 - setUp Thread Group
 - Получение токена
 - Извлечение токена из ответа
 - Запись токена в переменную
 - Thread Group
 - HTTP Header Manager
 - Публикация записи
 - Формирование записи
 - Извлечение идентификатора из ответа** (highlighted)
 - Получение записи по идентификатору
 - JSON Assertion
 - tearDown Thread Group
 - JDBC Connection Configuration
 - Чистка таблицы после тестов
 - View Results Tree

The right pane shows the configuration for the selected "Извлечение идентификатора из ответа" element:

JSON Extractor

Name: Извлечение идентификатора из ответа

Comments:

Apply to:
 Main sample and sub-samples Main sample only Sub-samples

Names of created variables: recId

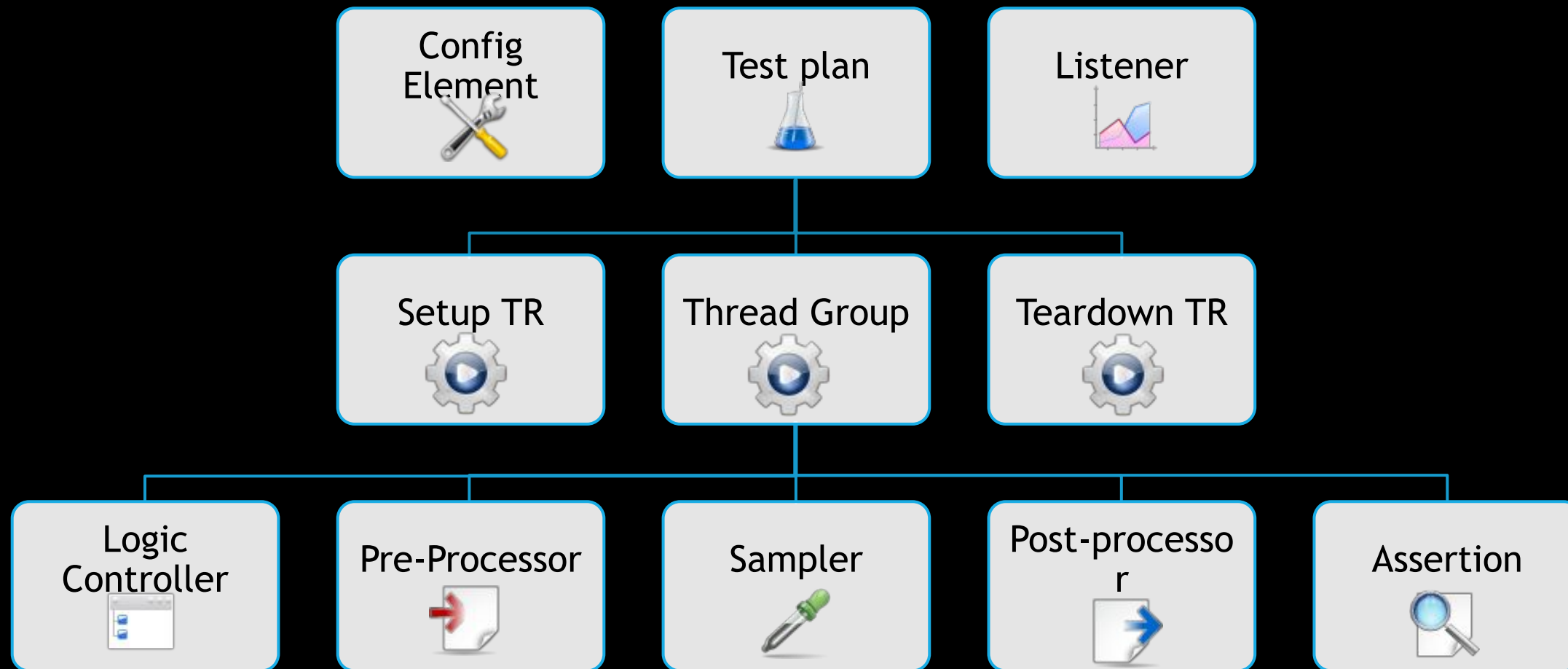
JSON Path expressions: \$.id

Match No. (0 for Random):

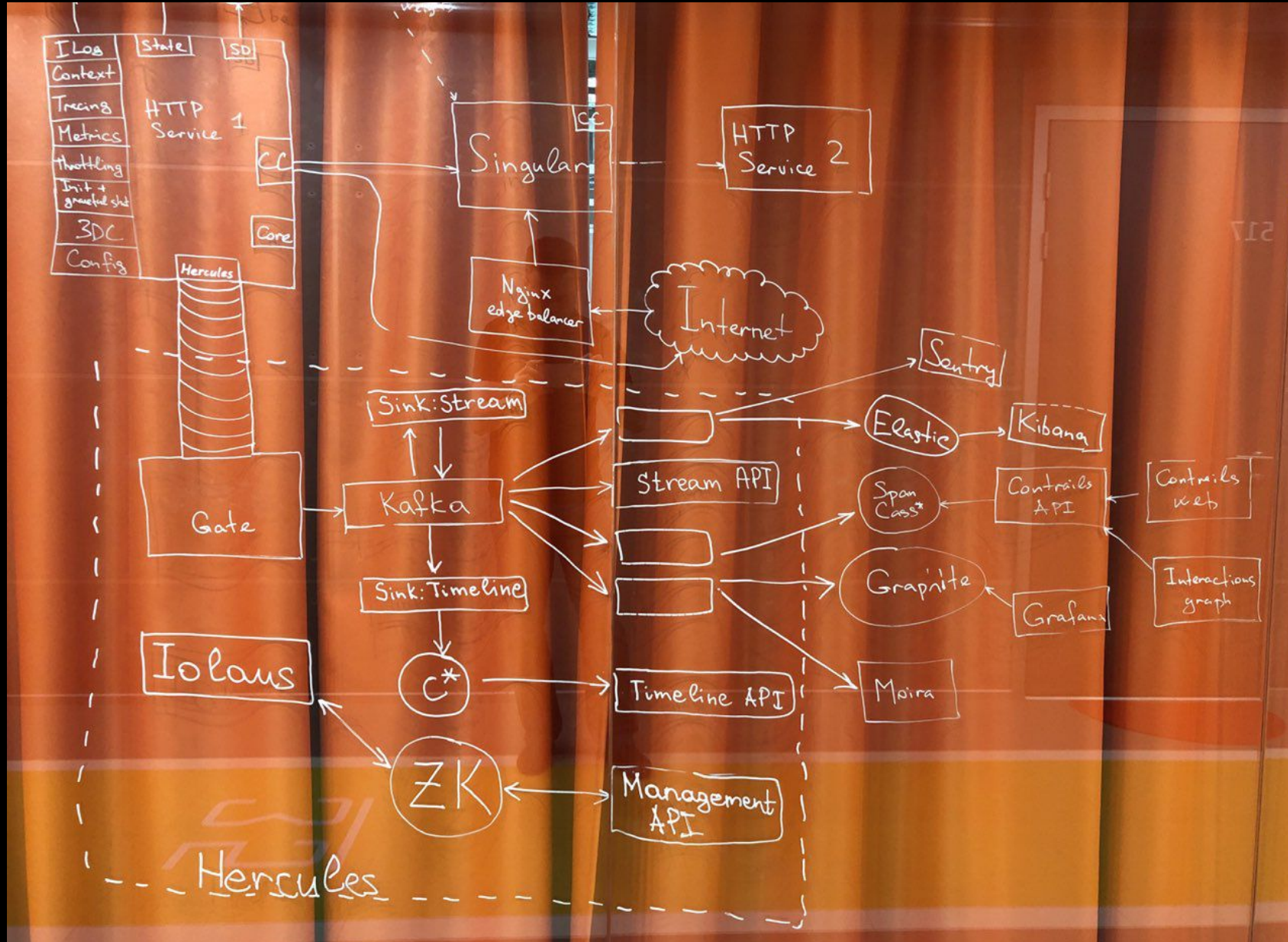
Compute concatenation var (suffix _ALL):

Default Values:

Компоненты JMeter



Тест цепочки отправки логов



Тест цепочки отправки логов



Thread group

The screenshot displays the Apache JMeter 4.0 r1823414 interface. The main window is titled "Test Plan" and contains a sub-element named "Тест отправки логов". This sub-element is highlighted with a green rounded rectangle. The right-hand pane shows the configuration for the selected "Thread Group". The "Name" field is set to "Тест отправки логов". The "Thread Properties" section is also highlighted with a green rounded rectangle and contains the following settings:

- Number of Threads (users): 1
- Ramp-Up Period (in seconds): 1
- Loop Count: Forever 1
- Delay Thread creation until needed
- Scheduler

Other visible settings in the Thread Group configuration include "Comments", "Action to be taken after a Sampler error" (set to "Continue"), and "Action to be taken after a Thread error" (set to "Continue").

Sampler

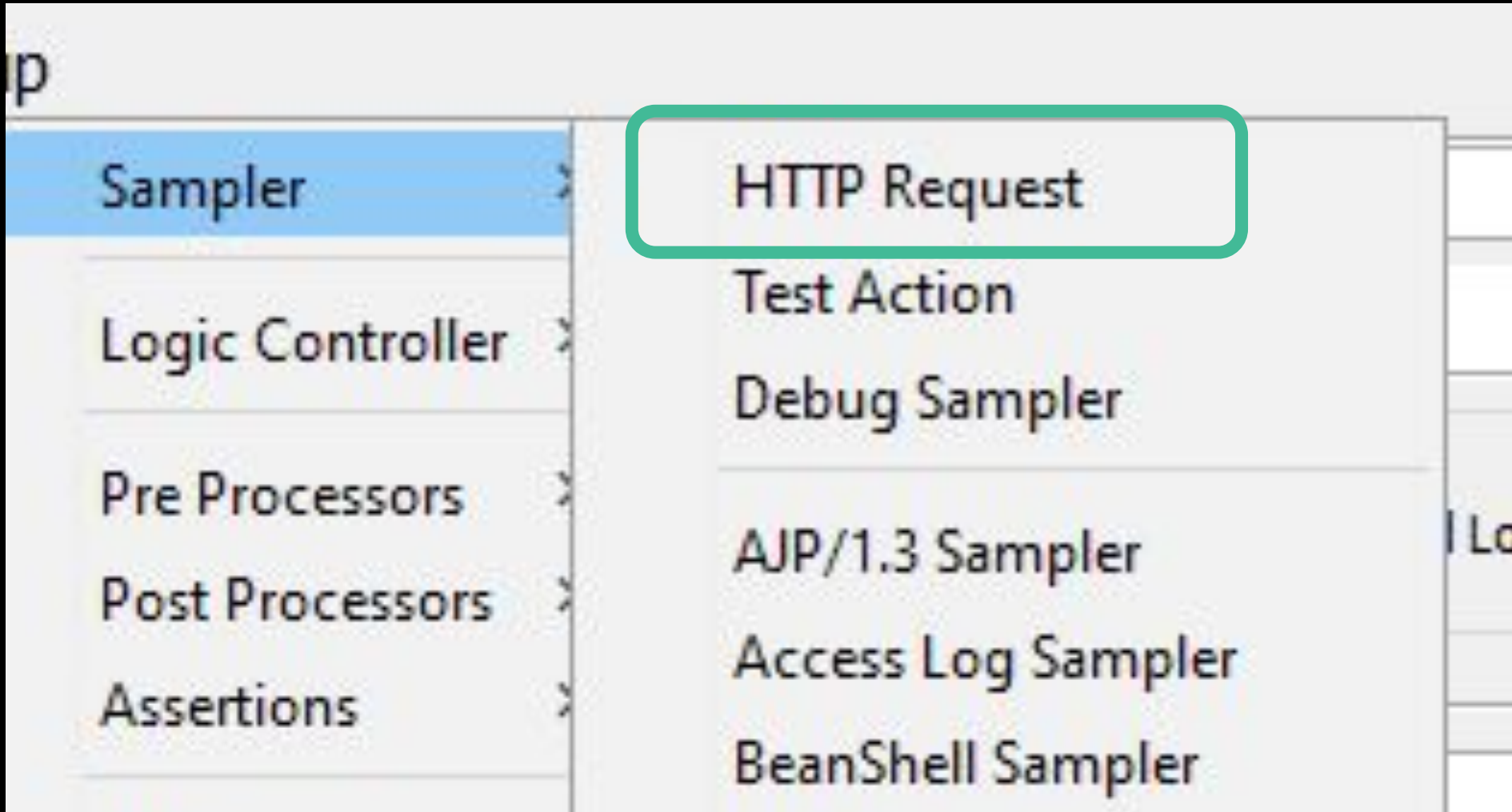
The screenshot shows the Apache JMeter 4.0 interface. The title bar reads "Apache JMeter (4.0 r1823414)". The menu bar includes "File", "Edit", "Search", "Run", "Options", and "Help". The toolbar contains various icons for file operations and execution. The main window displays a "Test Plan" tree on the left with a selected node "Тест отправки логов". The "Thread Group" panel is active, and the "Add" menu is open, showing a list of sampler options under the "Sampler" category.

Menu Item	Shortcut
Add	
Add Think Times to children	
Start	
Start no pauses	
Validate	
Cut	Ctrl+X
Copy	Ctrl+C
Paste	Ctrl+V
Duplicate	Ctrl+Shift+C
Remove	Delete
Open...	
Merge	
Save Selection As...	
Save Node As Image	Ctrl+G
Save Screen As Image	Ctrl+Shift+G

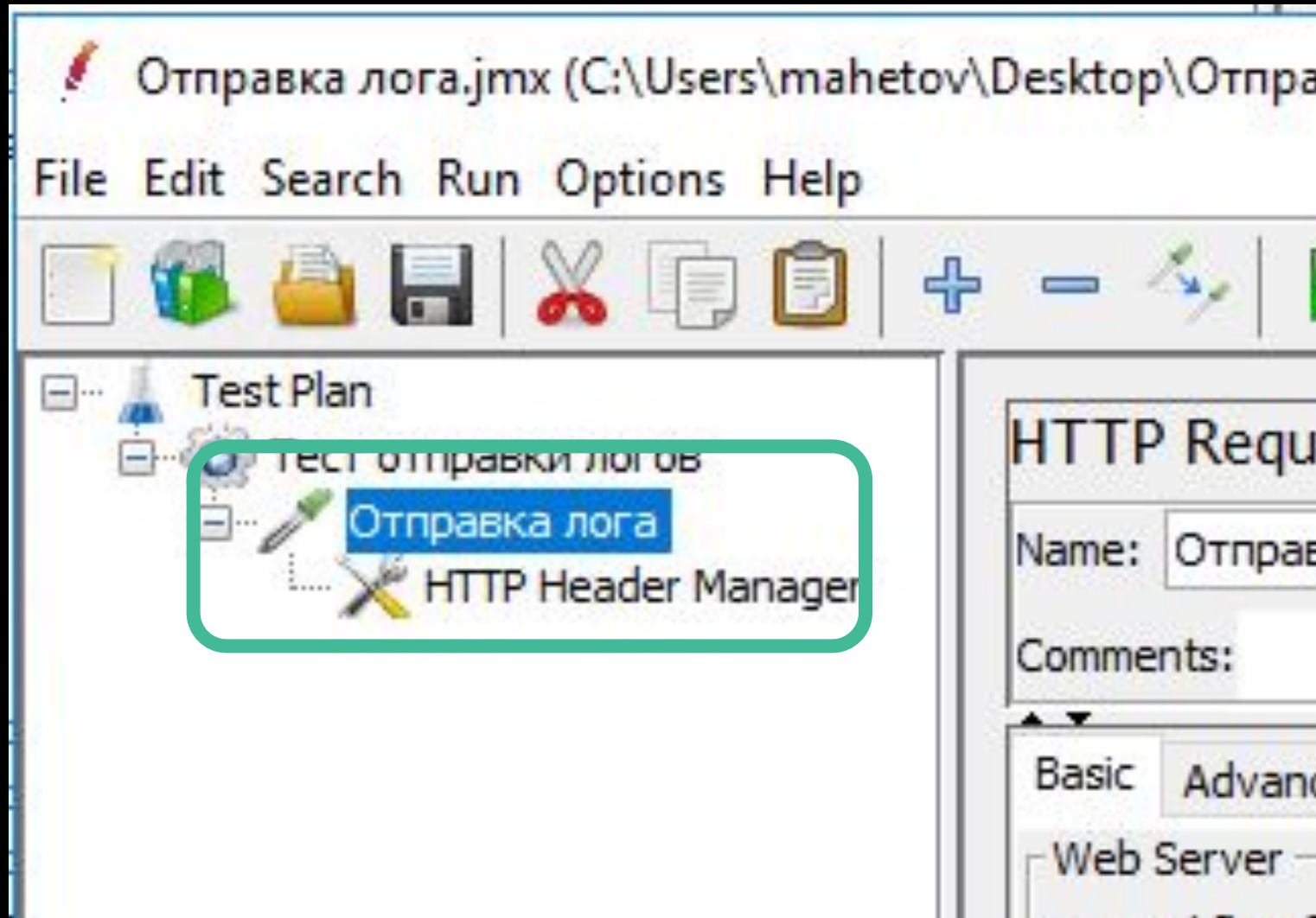
The "Sampler" menu options are:

- HTTP Request
- Test Action
- Debug Sampler
- AJP/1.3 Sampler
- Access Log Sampler
- BeanShell Sampler
- FTP Request
- JDBC Request
- JSR223 Sampler
- JUnit Request
- Java Request
- OS Process Sampler
- SSH Command
- SSH SFTP
- TCP Sampler
- jp@gc - Dummy Sampler

Sampler



Отправка лог-записи



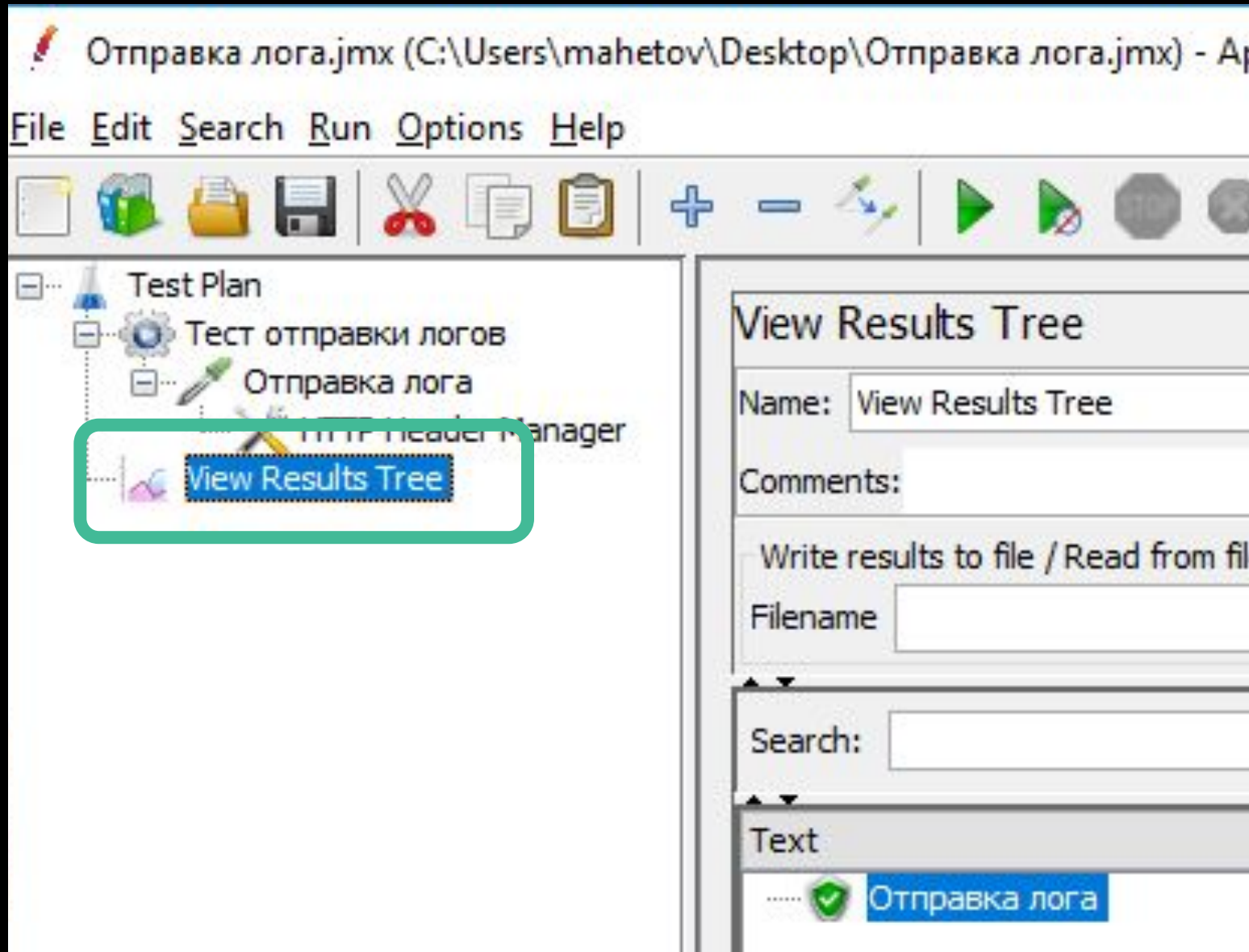
Отправка лог-записи

The screenshot shows a web client interface with the following details:

- Web Server:** Protocol [http], Server Name or IP: `vostok14`, Port Number: `6308`
- HTTP Request:** Method: `GET`, Path: `/async-logs/hercules-test-elk-0-2018.12.18`, Content encoding: [empty]
- Options:** Redirect Automatically, Follow Redirects, Use KeepAlive, Use multipart/form-data for POST, Browser-compatible headers
- Parameters:** [empty]
- Body Data:**

```
{ "level": "ERROR", "logger": "Gatling logger", "@timestamp": "2018-12-18T12:14:53.693Z", "message": "123_SomeGatlingLogs_7dd44be7-9059-4b6f-b625-481e3097d8a4", "thread": "main", "From": "Gatling" }
```

Просмотр результата отправки



Просмотр результата отправки

Text

..... Отправка лога

Sampler result Request Response data

Thread Name: Тест отправки логов 1-1
Sample Start: 2018-12-18 15:19:08 MSK
Load time: 8
Connect Time: 2
Latency: 8
Size in bytes: 99
Sent bytes:453
Headers size in bytes: 99
Body size in bytes: 0
Sample Count: 1
Error Count: 0
Data type ("text"|"bin"|""):
Response code: 200
Response message: OK

Response headers:
HTTP/1.1 200 OK
Connection: keep-alive
Content-Length: 0
Date: Tue, 18 Dec 2018 12:19:08 GMT

Scroll automatically?

Raw Parsed

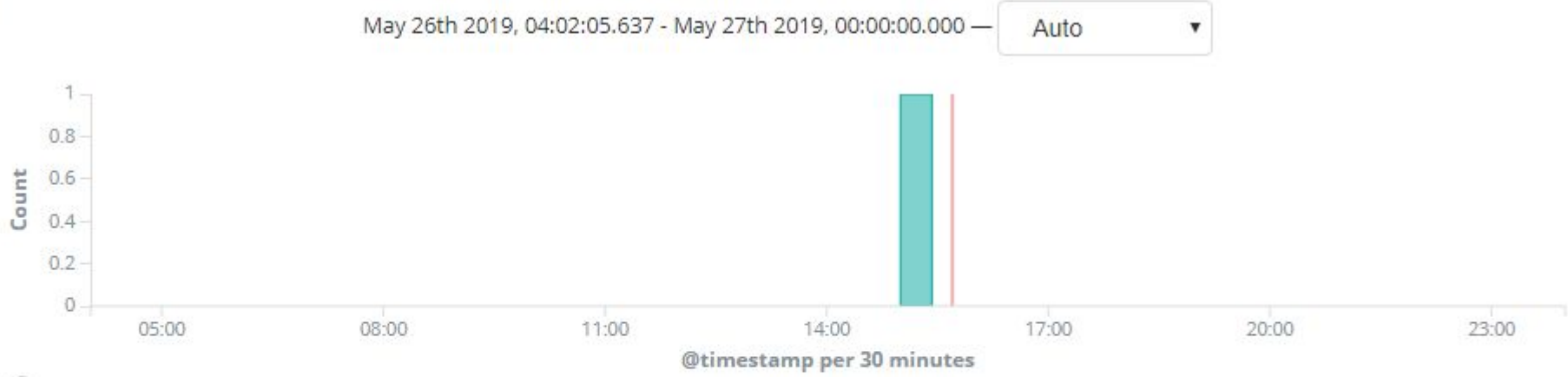
Просмотр результата отправки

1 hit New Save Open Share Auto-refresh May 26th 2019, 04:02:05.637 to May 27th 2019, 00:00:00.000

Search... (e.g. status:200 AND extension:PHP) Uses lucene query syntax

Add a filter +

hercules-test-elk* May 26th 2019, 04:02:05.637 - May 27th 2019, 00:00:00.000 Auto



Count

@timestamp per 30 minutes

Time Time _source

May 26th 2019, 15:13:56.757

@timestamp: May 26th 2019, 15:13:56.757 elk-index: hercules-test-elk-0

Dkj;tyyjcnm.dfsgds: fdsfds level: ERROR logger: Gatling logger чиселка: 33 thread: main

From: Gatling message: 123_SomeGatlingLogs_280c8626-6ae7-4506-8423-7a91b4ec29a3 Array: a

1, a2, a3 host: vm-hercules05 ПолеНаРусском: Значение на русском stand: testing _id: ADd

h70V0iFD40I nrf54R67z1f5vDi8AA type: LogEvent index: hercules-test-elk-0-2019.05.26

Table JSON View surrounding documents View single document

```
1 {
2   "_index": "hercules-test-elk-0-2019.05.26",
```

Чтение лог-записи из ES

The screenshot shows the JMeter GUI configuration for an HTTP Request. The test plan tree on the left includes a 'Test Plan' containing 'Тест отправки логов', 'Отправка лога', 'Чтение лога из Elasticsearch' (highlighted with a green box), and 'JSON Assertion'. The main configuration area is titled 'HTTP Request' and includes the following fields:

- Name: Чтение лога из Elasticsearch
- Comments:
- Basic tab selected, with 'Advanced' also visible.
- Web Server section: Protocol [http], Server Name or IP: elk-hercules01
- HTTP Request section: Method: GET, Path: 8/_search?q=\${__urlencode(@timestamp:"2018-12-18T12:14:53.693Z")} (highlighted with a green box)
- Options: Redirect Automatically, Follow Redirects, Use KeepAlive, Use multipart/form-data for POST, Browser-o
- Parameters, Body Data, Files Upload tabs are visible at the bottom.

Проверка текста в ответе

```
{ "took": 2,  
  ...  
  "hits": {  
    ...  
    "hits": [{  
      "_index": "hercules-test-elk-0-2019.05.26",  
      ...  
      "_source": {  
        "@timestamp": "2019-05-26T10:13:56.757000000Z",  
        "elk-index": "hercules-test-elk-0",  
        ...  
        "level": "ERROR",  
        ...  
        "stand": "testing"}  
    ]  
  }  
}
```

Проверка текста в ответе

The screenshot shows the SoapUI interface for configuring a JSON Assertion. The left sidebar contains a list of tools, with 'JSON Assertion' highlighted in blue. The main panel is titled 'JSON Assertion' and contains the following fields and options:

- Name: JSON Assertion
- Comments: (empty)
- Assert JSON Path exists: `$.hits.hits[0]._source.level` (highlighted with a green box)
- Additionally assert value
- Match as regular expression
- Expected Value: **ERROR** (highlighted with a green box)

Дублирование запросов

Отправка лога.jmx (C:\Users\mahetov\Desktop\Отправка лога.jmx) - Apache JMeter (4.0 r1823414)

File Edit Search Run Options Help

00:00:01 0 0/1

Test Plan

- Тест отправки логов
 - Отправка лога
 - HTTP Header Manager
 - Чтение лога из Elasticsearch
 - JSON Assertion
 - View Results 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.

Text

- Отправка лога
- Чтение лога из Elasticsearch**

Sampler result Request Response data

```
{ "took": 2, "timed_out": false, "_shards": { "total": 5, "successful": 5, "skipped": 0, "failed": 0 }, "hits": { "total": 3, "max_score": 1.0, "hits": [ { "_index": "hercules-test-elk-0-2018.12.18", "_type": "LogEvent", "_id": "869c1ad0-02be-11e9-8311-e1ae469d6000", "score": 1.0, "source": { "@timestamp": "2018-12-18T12:14:53.693000000Z", "level": "ERROR", "logger": "Gatling logger", "thread": "main", "From": "Gatling", "message": "123_SomeGatlingLogs_7dd44be7-9059-4b6f-b625-481e3097d8a4" } }, { "_index": "hercules-test-elk-0-2018.12.18", "_type": "LogEvent", "_id": "869c1ad0-02be-11e9-9c30-4bc113d46000", "score": 1.0, "source": { "@timestamp": "2018-12-18T12:14:53.693000000Z", "level": "ERROR", "logger": "Gatling logger", "thread": "main", "From": "Gatling", "message": "123_SomeGatlingLogs_7dd44be7-9059-4b6f-b625-481e3097d8a4" } }, { "_index": "hercules-test-elk-0-2018.12.18", "_type": "LogEvent", "_id": "869c1ad0-02be-11e9-93c9-dc386792c000", "score": 1.0, "source": { "@timestamp": "2018-12-18T12:14:53.693000000Z", "level": "ERROR", "logger": "Gatling logger", "thread": "main", "From": "Gatling", "message": "123_SomeGatlingLogs_7dd44be7-9059-4b6f-b625-481e3097d8a4" } } ] } }
```

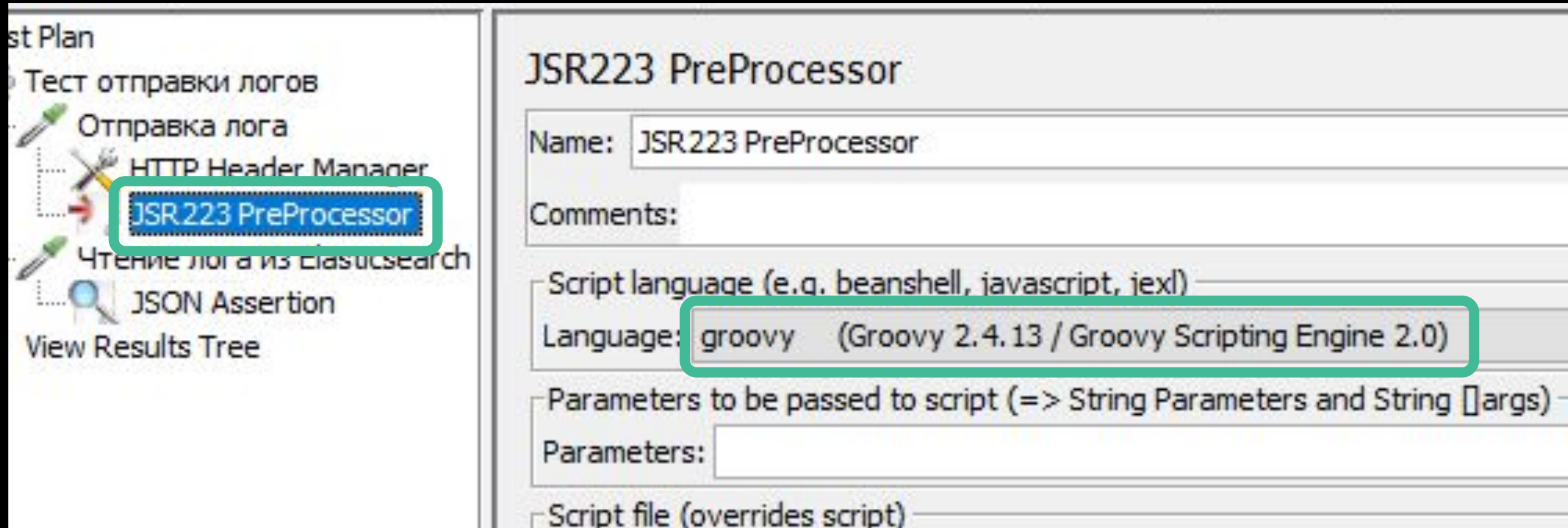
Search: Find Case sensitive Regular exp.

Scroll automatically?

Дублирование запросов

Sampler result	Request	Response data
		<pre>{\"took\":2,\"timed_out\":false,\"_shards\":{\"total\":5,\"successful\":5,\"skipped\":0,\"failed\":0},\"hits\":{\"total\":3,\"max_score\":1.0,\"hits\":[{\"_index\":\"hercules-test-elk-0-2018.12.18\",\"_type\":\"LogEvent\",\"_id\":\"869c1ad0-02be-11e9-8311-e1ae469d6000\",\"_score\":1.0,\"_source\":{\"@timestamp\":\"2018-12-18T12:14:53.693000000Z\",\"level\":\"ERROR\",\"logger\":\"Gatling logger\",\"thread\":\"main\",\"From\":\"Gatling\",\"message\":\"123_SomeGatlingLogs_7dd44be7-9059-4b6f-b625-481e3097d8a4\"}},{\"_index\":\"hercules-test-elk-0-2018.12.18\",\"_type\":\"LogEvent\",\"_id\":\"869c1ad0-02be-11e9-9c30-4bc113d46000\",\"_score\":1.0,\"_source\":{\"@timestamp\":\"2018-12-18T12:14:53.693000000Z\",\"level\":\"ERROR\",\"logger\":\"Gatling logger\",\"thread\":\"main\",\"From\":\"Gatling\",\"message\":\"123_SomeGatlingLogs_7dd44be7-9059-4b6f-b625-481e3097d8a4\"}},{\"_index\":\"hercules-test-elk-0-2018.12.18\",\"_type\":\"LogEvent\",\"_id\":\"869c1ad0-02be-11e9-93c9-dc386792c000\",\"_score\":1.0,\"_source\":{\"@timestamp\":\"2018-12-18T12:14:53.693000000Z\",\"level\":\"ERROR\",\"logger\":\"Gatling logger\",\"thread\":\"main\",\"From\":\"Gatling\",\"message\":\"123_SomeGatlingLogs_7dd44be7-9059-4b6f-b625-481e3097d8a4\"}}]}}}</pre>

Pre-processor для подстановки времени



The image shows a screenshot of the Apache JMeter configuration interface. On the left, a tree view shows the test plan structure with 'JSR223 PreProcessor' highlighted in a blue box. On the right, the configuration panel for 'JSR223 PreProcessor' is shown. The 'Name' field is 'JSR223 PreProcessor'. The 'Language' dropdown is set to 'groovy (Groovy 2.4.13 / Groovy Scripting Engine 2.0)', which is also highlighted in a blue box. Other fields include 'Comments', 'Script language (e.g. beanshell, javascript, jexl)', 'Parameters to be passed to script (= > String Parameters and String []args)', 'Parameters', and 'Script file (overrides script)'.

st Plan

- Тест отправки логов
 - Отправка лога
 - HTTP Header Manager
 - JSR223 PreProcessor**
 - Чтение лога из Elasticsearch
 - JSON Assertion
- View Results Tree

JSR223 PreProcessor

Name: JSR223 PreProcessor

Comments:

Script language (e.g. beanshell, javascript, jexl)

Language: **groovy (Groovy 2.4.13 / Groovy Scripting Engine 2.0)**

Parameters to be passed to script (= > String Parameters and String []args)

Parameters:

Script file (overrides script)

Текущее время в переменной

Script:

```
1 dt = java.time.Instant.now().toString()  
2 //vars.put("tmstamp", "2018-08-02T20:19:52.703Z")  
3 vars.put("tmstamp", dt)  
4
```


Добавление переменной и функции

```
Parameters Body Data Files Upload
1 [{"level": "ERROR", "logger": "Gatling logger", "@timestamp": "${tmstmp}" "message":
  " 123_SomeGatlingLog:_${__UUID}", "thread": "main", "From": "Gatling"}]
```

Переменная в запросе лог-записи

Host or IP: Port:



URL: Content type:

Follow Redirects Use KeepAlive Use multipart/form-data for POST Browse

Подстановка значения переменной и функции

Search: Case sensitive Regular exp.

Text ▼

-  Отправка лога
-  Чтение лога из Elasticsearch

Sampler result Request Response data

```
POST
http://vostok15:6308/async-logs/hercules-test-elk-0-2018.12.18

POST data:
{"level":"ERROR","logger":"Gatling
logger","@timestamp":"2018-12-18T13:35:52.791Z","mes
sage":""
123_SomeGatlingLogs_fee6851c-5082-4cde-9208-15ca740d
0bbd","thread":"main", "From":"Gatling"}

[no cookies]
```

Ошибка при запросе лог-записи

The screenshot displays a test runner interface with two main panels. The left panel, titled 'Text', shows a test execution tree with three steps: 'Отправка лога' (Log sending) with a green checkmark, 'Чтение лога из Elasticsearch' (Reading log from Elasticsearch) with a red 'X' icon, and 'JSON Assertion' with a red 'X' icon and a blue highlight. The right panel, titled 'Assertion result', shows the following text: 'Assertion error: false', 'Assertion failure: true', and 'Assertion failure message: No results for path: \$[hits][hits][0][_source][level]'.

Text

- Отправка лога
- Чтение лога из Elasticsearch
- JSON Assertion

Assertion result

Assertion error: false
Assertion failure: true
Assertion failure message: No results for path: \$[hits][hits][0][_source][level]

Добавление таймаута

The screenshot displays the Apache JMeter application window. The title bar reads "Отправка лога.jmx (C:\Users\mahetov\Desktop\Отправка лога.jmx) - Apache JMeter (4.0 r1823414)". The menu bar includes "File", "Edit", "Search", "Run", "Options", and "Help". The toolbar contains various icons for file operations and execution. The left sidebar shows a test plan tree with the following elements: "Test Plan", "Тест отправки логов", "Отправка лога", "HTTP Header Manager", "JSR223 PreProcessor", "Чтение лога из Elasticsearch", "JSON Assertion", "Constant Timer", and "View Results Tree". The "Constant Timer" element is highlighted with a green box. The right sidebar shows the configuration for the "Constant Timer", with the "Thread Delay (in milliseconds)" field set to "2000" and also highlighted with a green box.

Отправка лога.jmx (C:\Users\mahetov\Desktop\Отправка лога.jmx) - Apache JMeter (4.0 r1823414)

File Edit Search Run Options Help

Test Plan

- Тест отправки логов
 - Отправка лога
 - HTTP Header Manager
 - JSR223 PreProcessor
 - Чтение лога из Elasticsearch
 - JSON Assertion
 - Constant Timer**
 - View Results Tree

Constant Timer

Name: Constant Timer

Comments:

Thread Delay (in milliseconds): 2000

Тест пройден!

The screenshot displays a monitoring tool interface with the following components:

- Filename:** A text input field with a "Browse..." button.
- Log/Display Only:** Checkboxes for "Errors" and "Success".
- Search:** A search input field with checkboxes for "Case sensitive" and "Regular exp.", and "Search" and "Reset" buttons.
- Text:** A tree view showing test results:
 - Отправка лога (Success)
 - Чтение лога из Elasticsearch (Failure)
 - Отправка лога (Success)
 - Чтение лога из Elasticsearch (Success) - highlighted with a green box
- Sampler result:** A detailed view of the selected test result:
 - Request
 - Response data
 - Thread Name: Тест отправки логов 1-1
 - Sample Start: 2018-12-18 16:51:29 MSK
 - Load time: 6
 - Connect Time: 3
 - Latency: 6
 - Size in bytes: 543
 - Sent bytes: 278
 - Headers size in bytes: 87
 - Body size in bytes: 456
 - Sample Count: 1
 - Error Count: 0
 - Data type ("text"|"bin"|"): text
 - Response code: 200

Ошибка в цепочке

The screenshot displays a testing tool interface with the following elements:

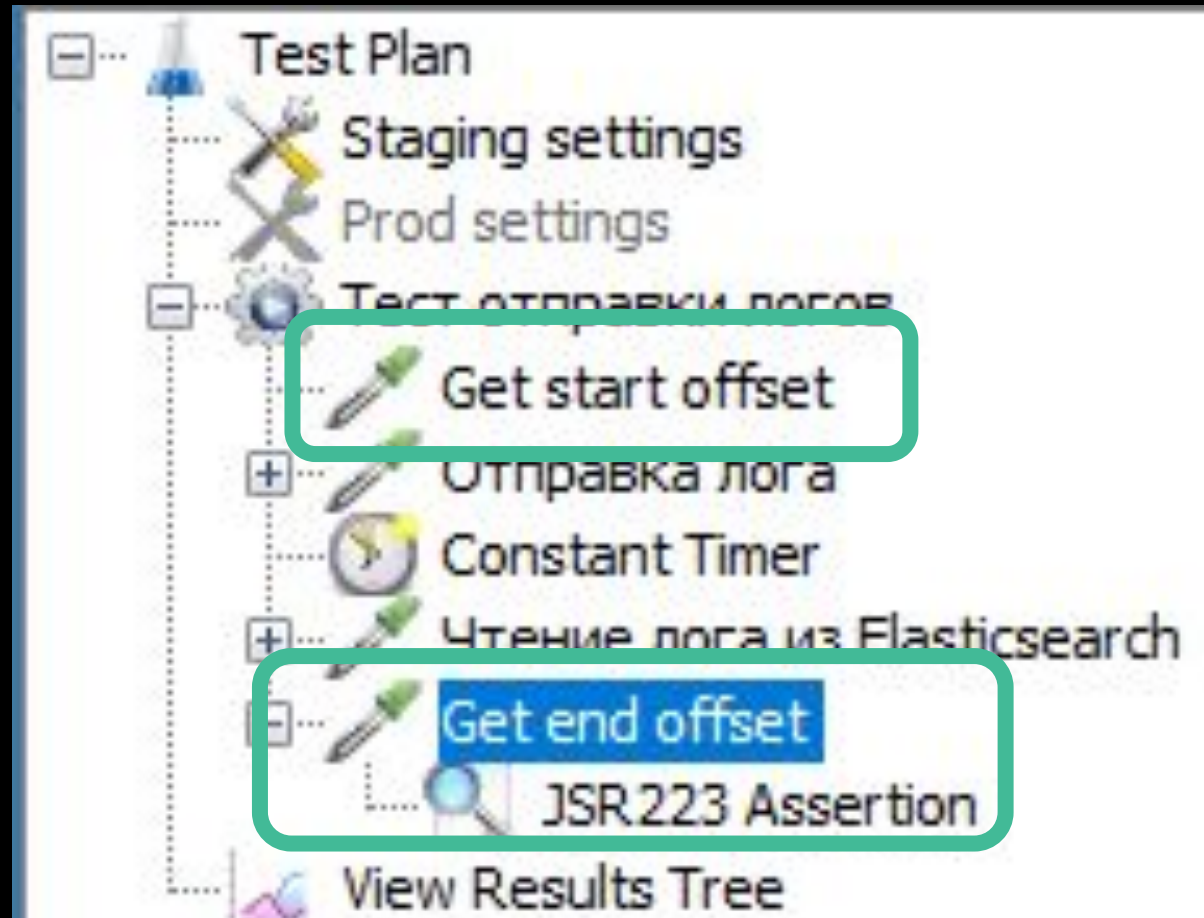
- Top section: "Write results to file / Read from file" with a "Filename" input field, a "Browse..." button, and "Log/Display Only" options for "Errors" and "Successes".
- Search section: A "Search:" input field, checkboxes for "Case sensitive" and "Regular exp.", and "Search" and "Reset" buttons.
- Main content area: A tree view under "Text" showing a sequence of steps:
 - "Отправка лога" (Send log) with a green checkmark.
 - "Чтение лога из Elasticsearch" (Read log from Elasticsearch) with a red 'x' icon.
 - "JSON Assertion" with a red 'x' icon and a blue highlight box.
- Right panel: "Assertion result" showing:
 - Assertion error: false
 - Assertion failure: true
 - Assertion failure message: No results for path: `$_source[level]`

Офсеты Kafka

TOPIC	PARTITION	CURRENT-OFFSET
elk_adapter_test_0	0	17111565
elk_adapter_test_0	2	17105615
elk_adapter_test_0	1	17105427

Проверка увеличения офсета Kafka

TOPIC	PARTITION	CURRENT-OFFSET
elk_adapter_test_0	0	17111565
elk_adapter_test_0	2	17105615
elk_adapter_test_0	1	17105427



Получение офсета

```
1 import java.util.Arrays;
2 import java.util.Properties;
3 import org.apache.kafka.clients.consumer.KafkaConsumer;
4 import org.apache.kafka.common.TopicPartition;
5 import org.apache.kafka.common.serialization.StringDeserializer;
6
7 Properties props = new Properties();
8 props.put("bootstrap.servers", "${kafkaHostPort}");
9 props.put("group.id", "consumer-tutorial1");
10 props.put("key.deserializer", StringDeserializer.class.getName());
11 props.put("value.deserializer", StringDeserializer.class.getName());
12 KafkaConsumer<String, String> consumer = new KafkaConsumer<>(props);
13
14 String topic = "${kafkaTopic}";
15
16 List<TopicPartition> tp = new ArrayList<>();
17 consumer.partitionsFor(topic).each{x -> tp.add(new TopicPartition(topic, x.partition()))};
18 Map<TopicPartition, Long> topicPartitionLongMap = consumer.endOffsets(tp);
19 long offset = 0;
20 topicPartitionLongMap.each{k, v ->
21     offset += v;
22     log.info(v.toString());
23 };
24 consumer.close();
25 vars.put("startOffset", offset.toString())
26 System.out.println(offset.toString());
27 return "Current offset: " + offset.toString()
```

Сохранение текущего офсета

```
topicPartitionLongMap.each{k, v ->
    offset += v;
    log.info(v.toString());
};
consumer.close();
vars.put("startOffset", offset.toString())
System.out.println(offset.toString());
return "Current offset: " + offset.toString()
```

Сравнение офсетов до отправки и после

Script:

```
1 String var1 = vars.get("startOffset");
2 String var2 = vars.get("endOffset");
3
4 fail = var1==var2;
5
6 AssertionResult.setFailure(fail)
7 AssertionResult.setFailureMessage("Offsets are equal")
```

Ошибка осталась

Search: Case sensitive Reg

Text

- ✓ Get start offset
- ✓ Отправка логов
- ✓ Get end offset
- +..... ✗ Чтение логов из Elasticsearch

Sampler result

Thread Name:
Sample Start:
Load time: 5
Connect Time
Latency: 5
Size in bytes:
Sent bytes: 27
Headers size i
Body size in b
Sample Count

Плагины

The screenshot displays the JMeter Plugins Manager window. The left sidebar, titled 'Available Plugins', contains a list of plugins with checkboxes. The 'Selenium/WebDriver Support' plugin is selected and highlighted in blue. The main pane shows the details for this plugin, including its title, vendor, description, documentation link, version information, and a preview of its configuration in the JMeter GUI.

JMeter Plugins Manager

Installed Plugins Available Plugins Upgrades

Search...

- Page Data Extractor
- Parallel Controller & Sampler
- Parameterized Controller & Set Variables Action
- RTE Protocol Support
- Random CSV Data Set
- Redis Data Set
- Rotating JTL Listener
- SSHMon Sample Collector
- Selenium/WebDriver Support**
- Siebel CRM Recorder
- Synthesis Report
- Test Plan Check Tool
- UDP Protocol Support
- Variables from CSV File
- WS Security for SOAP
- WebSocket Sampler by Maciej Zaleski
- WebSocket Samplers by Peter Doornbosch
- Weighted Switch Controller
- XML Plugins
- XMPP Protocol Support
- jmeter - JMS Support
- jmeter - LDAP Protocol Support
- jmeter - Mail/SMTP Support
- jmeter - MongoDB Support

Selenium/WebDriver Support

Vendor: *JMeter-Plugins.org*

This plugin allows testing real browser behavior using Selenium/WebDriver technology

Documentation: <https://github.com/undera/jmeter-plugins-webdriver>

What's new in version 3.0: Upgrade Selenium library to version 3.14.0

Maven groupId: *kg.apc*, artifactId: *jmeter-plugins-webdriver*, version: *3.0*

Test Plan

- Thread Group
 - jp@gc - Firefox Driver Config
 - jp@gc - Web Driver Sampler
- WorkBench

jp@gc - Web Driver Sampler

Name: jp@gc - Web Driver Sampler

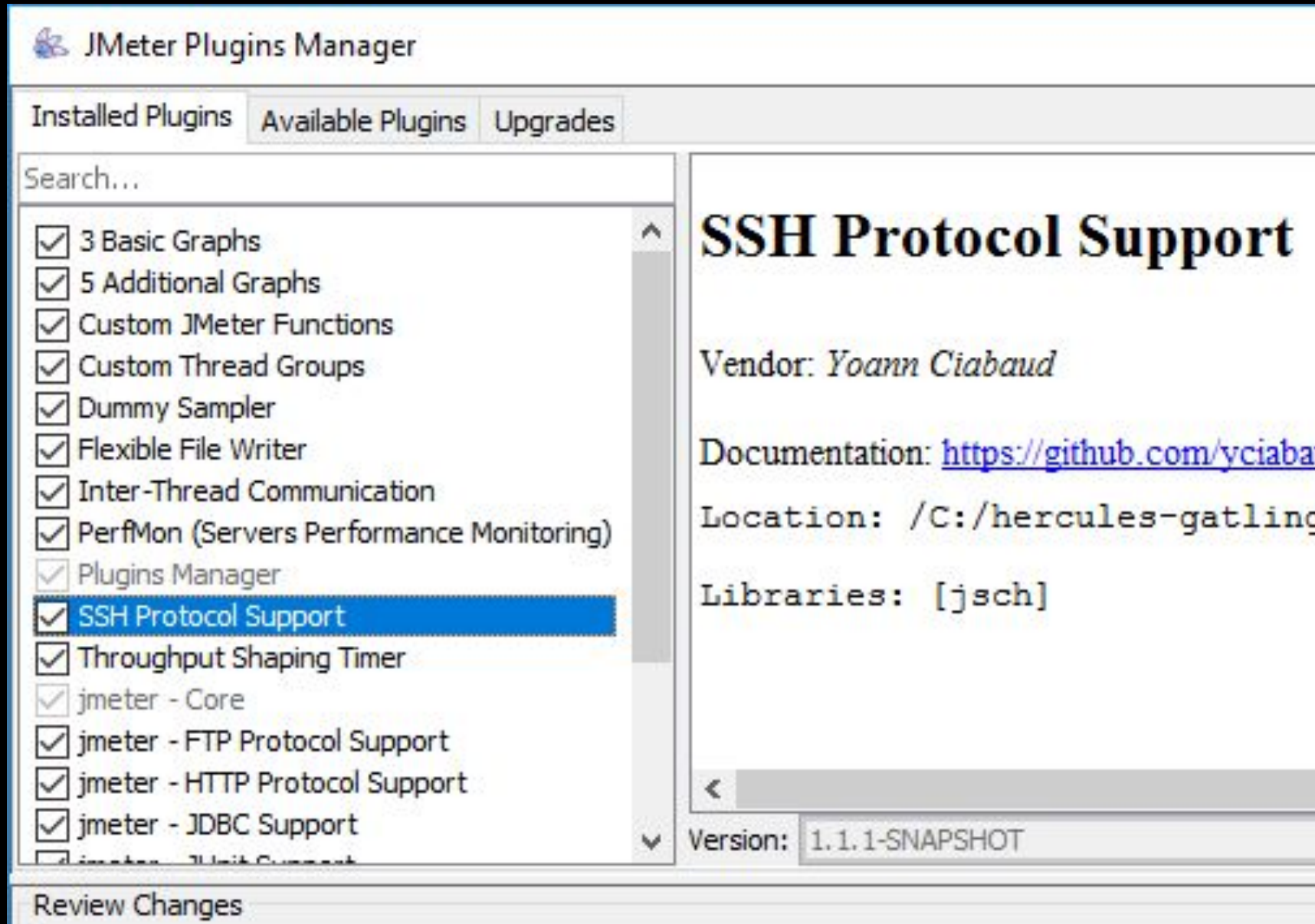
Comments:

[Help on this plugin](#)

Parameters (-) String Parameters and String[] args

Version: 3.0

SSH Protocol Support



The screenshot shows the JMeter Plugins Manager interface. The 'Available Plugins' tab is selected. A search bar is at the top left. A list of plugins is shown on the left, with 'SSH Protocol Support' highlighted. The right pane displays details for the selected plugin.

JMeter Plugins Manager

Installed Plugins | Available Plugins | Upgrades

Search...

- 3 Basic Graphs
- 5 Additional Graphs
- Custom JMeter Functions
- Custom Thread Groups
- Dummy Sampler
- Flexible File Writer
- Inter-Thread Communication
- PerfMon (Servers Performance Monitoring)
- Plugins Manager
- SSH Protocol Support**
- Throughput Shaping Timer
- jmeter - Core
- jmeter - FTP Protocol Support
- jmeter - HTTP Protocol Support
- jmeter - JDBC Support
- jmeter - JUnit Support

SSH Protocol Support

Vendor: *Yoann Ciabaud*

Documentation: <https://github.com/yciaba>

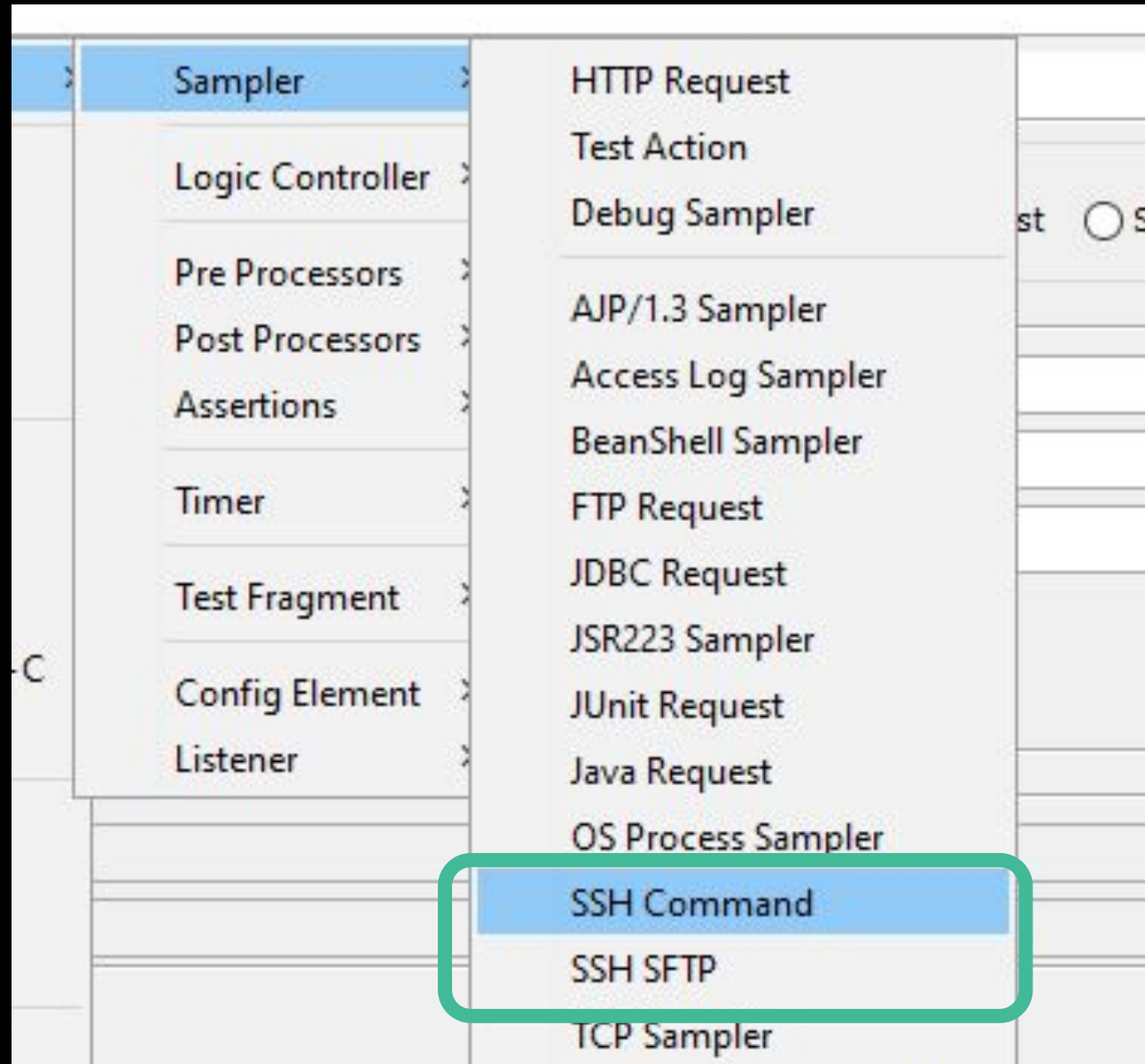
Location: `/C:/hercules-gatling`

Libraries: `[jsch]`

Version: 1.1.1-SNAPSHOT

Review Changes

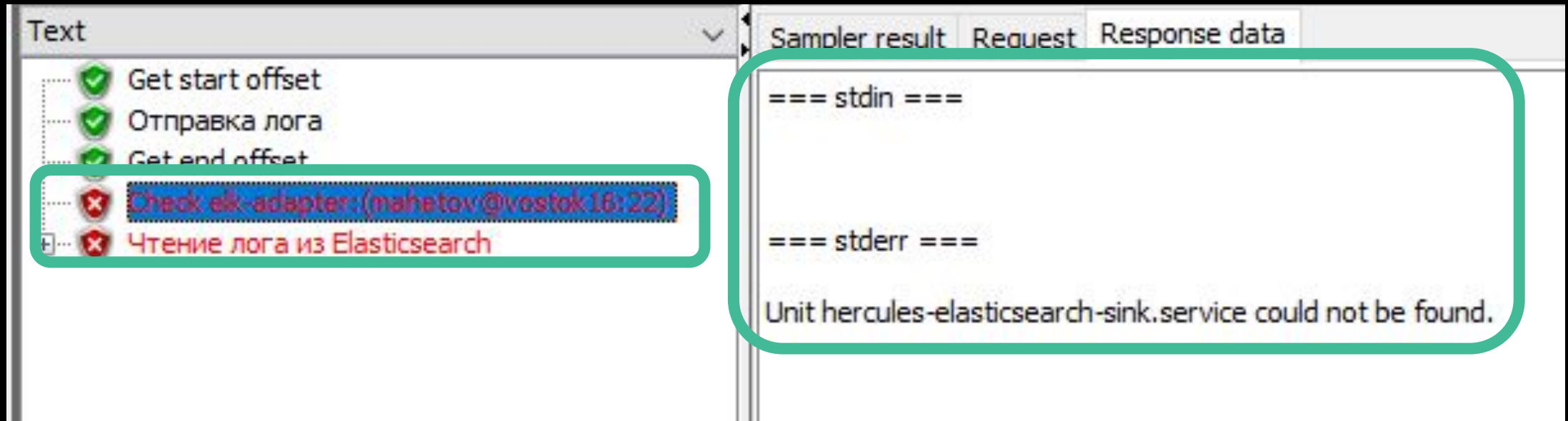
SSH Sampler



Команда проверки статуса

base:	
Command:	<code>systemctl status hercules-elasticsearch-sink</code>
Use return code:	True
Standard Error output:	True

Ошибка приложения



Text

- ✓ Get start offset
- ✓ Отправка лога
- ✓ Get end offset
- ✗ **Check elk-adapter:(mahatov@vostok 16:22)**
- ✗ Чтение лога из Elasticsearch

Sampler result Request Response data

```
=== stdin ===  
  
=== stderr ===  
Unit hercules-elasticsearch-sink.service could not be found.
```

Настройки для разных стендов

The screenshot shows the JMeter GUI. On the left, the 'Test Plan' tree is visible, with 'Staging settings' highlighted in a green box. Below it are 'Prod settings' and a test 'Тест отправки логов' containing several components: 'Отправка лога', 'HTTP Header Manager', 'JSR223 PreProcessor', 'Чтение лога из Elasticsearch', 'JSON Assertion', 'Constant Timer', and 'View Results Tree'. On the right, the 'User Defined Variables' section is shown, with a table of variables highlighted in a green box.

User Defined Variables

Name:

Comments:

Name:	Value
kafkaHostPort	vostok02:9092,vostok03:9092,vostok04:9092
kafkaTopic	legacy_logs_elk_0
elkAdapterHost	vostok15
elkAdapterPort	6308
elasticHost	elk-hercules01
elasticPort	9200

Настройки для разных стендов

Server Name or IP: Port Number:

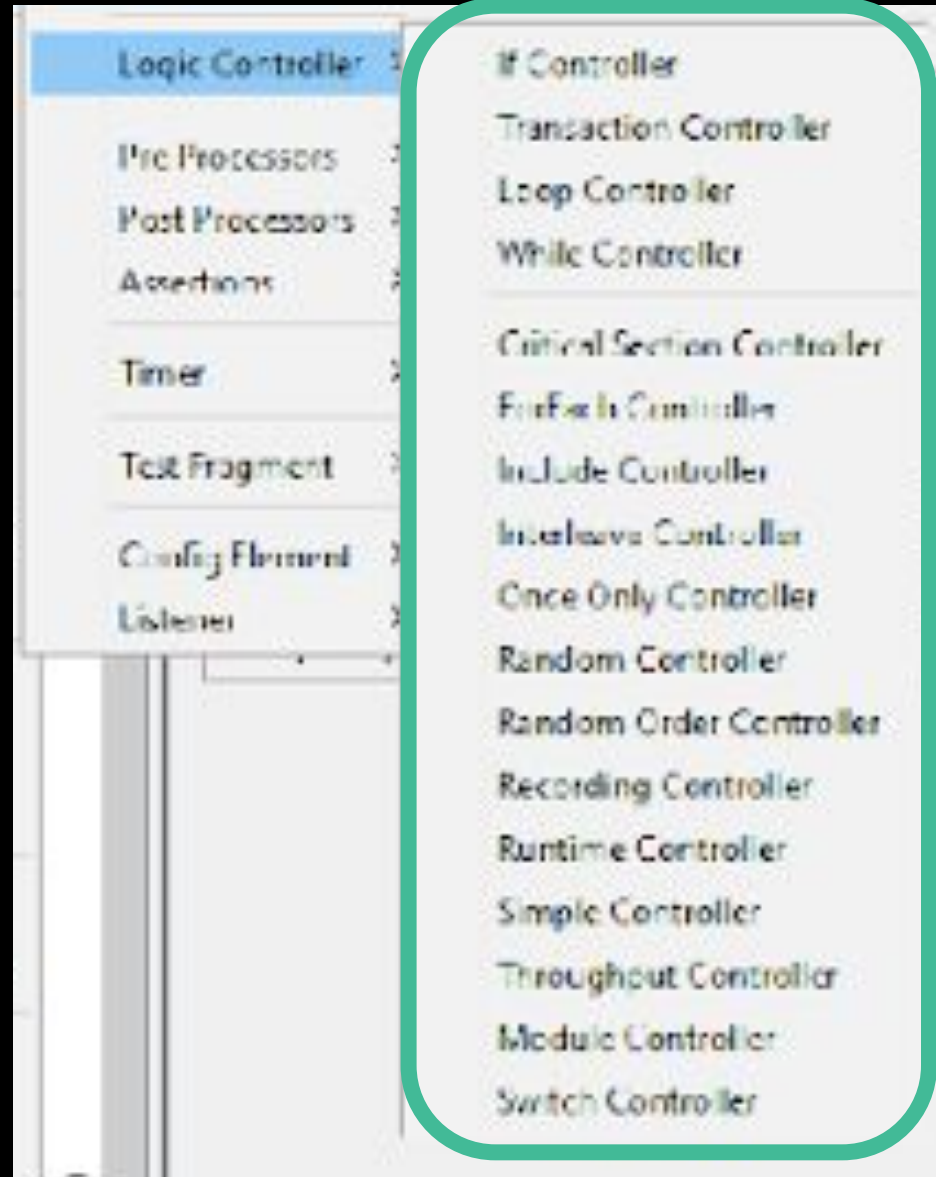
Path: Content encoding:

Follow Redirects Use KeepAlive Use multipart/form-data for POST Browser-compatible headers

Files Upload

```
,"logger":"Gatling logger","@timestamp":"${tmstmp}","message":  
ngLogs_${__UUID}","thread":"main","From":"Gatling"}]
```

Logic Controllers



Проверка нескольких хостов

Отправка лога.jmx (C:\Users\mahetov\Desktop\Отправка лога.jmx) - Apache JMeter (4.0 r1823414)

File Edit Search Run Options Help

00:00:12 0 0/1

Test Plan

- Staging settings
- Prod settings
- Тест отправки логов
 - Get start offset
 - ForEach Controller
 - Отправка лога \${host}
 - HTTP Header Manager
 - JSR223 PreProcessor
 - Constant Timer
 - Get end offset
 - Check elk-adapter
 - Чтение лога из Elasticsearch
 - View Results Tree

User Defined Variables

Name: Staging settings

Comments:

Name:	Value	Description
kafkaHostPort	vostok02:9092,vostok03:9092,vostok04:9092	
kafkaTopic	legacy_logs_elk_0	
elkAdapterHost_1	vostok14	
elkAdapterHost_2	vostok15	
elkAdapterPort	6308	
elasticHost	elk-hercules01	
elasticPort	9200	
elkSinkHost	vostok16	

Detail Add Add from Clipboard Delete Up Down

ForEach контроллер

The image shows a screenshot of the Apache JMeter GUI. On the left, a tree view of a 'Test Plan' is visible. A 'ForEach Controller' is highlighted with a green rounded rectangle. Below it, several steps are listed: 'Отправка лога \${host}', 'HTTP Header Manager', and 'JSR223 PreProcessor'. On the right, the configuration panel for the 'ForEach Controller' is shown. The 'Name' field is 'ForEach Controller'. The 'Input variable prefix' is 'elkAdapterHost'. The 'Output variable name' is 'host'. The 'Add "_" before number?' checkbox is checked.

Test Plan

- Staging settings
- Prod settings
- Тест отправки логов
 - Get start offset
 - ForEach Controller**
 - Отправка лога `${host}`
 - HTTP Header Manager
 - JSR223 PreProcessor
 - Constant timer
 - Get end offset
 - Check elk-adapter
 - Чтение лога из Elasticsearch
- View Results Tree

ForEach Controller

Name: ForEach Controller

Comments:

Input variable prefix: elkAdapterHost

Start index for loop (exclusive):

End index for loop (inclusive):

Output variable name: host

Add "_" before number ?

ForEach контроллер

HTTP Request

Name: Отправка логга `${host}`

Comments:

Basic Advanced

Web Server

Protocol [http]: Server Name or IP: `${host}` Port Number: `{AdapterPort}`

HTTP Request

Method: POST Path: `/async-logs/hercules-test-elk-0-${_time(yyyy.MM.dd)}` Content encoding:

Redirect Automatically Follow Redirects Use KeepAlive Use multipart/form-data for POST Browser-compatible headers

Parameters Body Data Files Upload

```
1 {"level":"ERROR","logger":"Gatling logger","@timestamp":"${tmstmp}","message":  
" 123_SomeGatlingLogs_${_UUID}|","thread":"main", "From":"Gatling"}
```

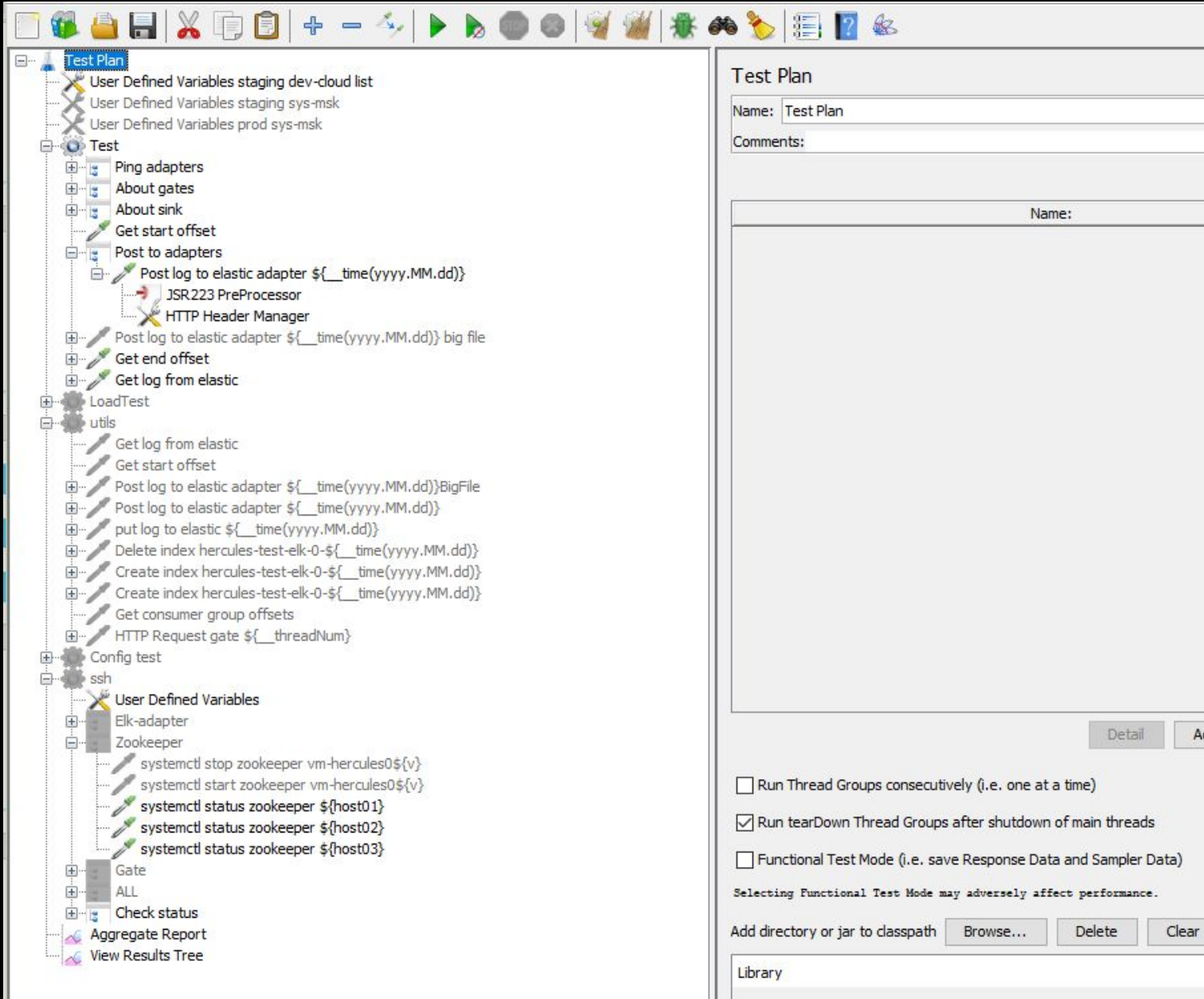

ForEach контроллер

The screenshot displays the Gatling test runner interface. On the left, a 'Text' panel shows a list of test steps: 'Get start offset' (green checkmark), 'Отправка лога vostok14' (green checkmark), 'Отправка лога vostok15' (green checkmark and highlighted in blue), 'Get end offset' (green checkmark), 'Check elk-adapter:(mahetov@vostok16:22)' (red X), and 'Чтение лога из Elasticsearch' (red X). On the right, the 'Request' tab is active, showing a 'POST' request to 'http://vostok15:6308/async-logs/hercules-test-e1k-0-2018.12.20'. Below the URL, the 'POST data' is shown as a JSON object:

```
{ "level": "ERROR", "logger": "Gatling logger", "@timestamp": "2018-12-20T09:19:38.603Z", "message": "123_SomeGatlingLogs_9a9fe9de-79a9-4830-99f7-08e749049458", "thread": "main", "From": "Gatling" }
```

 At the bottom of the request view, it says '[no cookies]'.

Реальный пример теста



The screenshot displays the JMeter Test Plan configuration interface. The left pane shows a hierarchical tree of test elements:

- User Defined Variables staging dev-cloud list
- User Defined Variables staging sys-msk
- User Defined Variables prod sys-msk
- Test
 - Ping adapters
 - About gates
 - About sink
 - Get start offset
 - Post to adapters
 - Post log to elastic adapter \${__time(yyyy.MM.dd)}
 - JSR223 PreProcessor
 - HTTP Header Manager
 - Post log to elastic adapter \${__time(yyyy.MM.dd)} big file
 - Get end offset
 - Get log from elastic
 - LoadTest
 - utils
 - Get log from elastic
 - Get start offset
 - Post log to elastic adapter \${__time(yyyy.MM.dd)}BigFile
 - Post log to elastic adapter \${__time(yyyy.MM.dd)}
 - put log to elastic \${__time(yyyy.MM.dd)}
 - Delete index hercules-test-elk-0-\${__time(yyyy.MM.dd)}
 - Create index hercules-test-elk-0-\${__time(yyyy.MM.dd)}
 - Create index hercules-test-elk-0-\${__time(yyyy.MM.dd)}
 - Get consumer group offsets
 - HTTP Request gate \${__threadNum}
 - Config test
 - ssh
 - User Defined Variables
 - Elk-adapter
 - Zookeeper
 - systemctl stop zookeeper vm-hercules0\${v}
 - systemctl start zookeeper vm-hercules0\${v}
 - systemctl status zookeeper \${host01}
 - systemctl status zookeeper \${host02}
 - systemctl status zookeeper \${host03}
 - Gate
 - ALL
 - Check status
 - Aggregate Report
 - View Results Tree

The right pane shows the 'Test Plan' configuration details:

- Name: Test Plan
- Comments:
- Name: (empty field)
- Run Thread Groups consecutively (i.e. one at a time)
- Run tearDown Thread Groups after shutdown of main threads
- Functional Test Mode (i.e. save Response Data and Sampler Data)
- Selecting Functional Test Mode may adversely affect performance.
- Add directory or jar to classpath: Browse... Delete Clear
- Library

Несколько экземпляров

The image displays two side-by-side screenshots of the Apache JMeter 4.0 interface, showing the results of a test plan. Both windows are titled 'ELKTest.jmx (D:\GitLab\vostok\hercules-gatling-jmeter-tests\jmeter\ELKTest.jmx) - Apache JMeter (4.0 r1823414)'. The left window shows a 'View Results Tree' for a 'Ping elasticsearch' test, with a list of results including 'Ping elasticsearch' (failed) and 'Ping gate' (successful). The right window shows a 'View Results Tree' for a 'status all' test, with results for 'status all vm-hercules04' and 'status all vm-hercules05' (both successful).

Left Window: 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.

Text	Sampler result	Request	Response data
✓ Ping elasticsearch			
✓ Ping gate			
✓ Get start offset			
✓ Post log to elastic adapter 2018.11.22			
✓ Get end offset			
✓ Get log from elastic			
✓ Ping elasticsearch			
✗ Ping elasticsearch			
✓ Ping gate			
✓ Get start offset			
✓ Post log to elastic adapter 2018.11.22			
✓ Get end offset			
✓ Get log from elastic			
✓ Ping elasticsearch			
✓ Ping gate			
✓ Ping gate			
✓ Get start offset			
✓ Post log to elastic adapter 2018.11.22			
✓ Get end offset			
✓ Get log from elastic			
✓ Ping elasticsearch			
✓ Ping gate			
✓ Ping gate			
✓ Get start offset			
✓ Post log to elastic adapter 2018.12.01			
✓ Get end offset			
✓ Get log from elastic			

Scroll automatically?

Search: Case sensitive Regular exp.

Right Window: 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.

Text	Sampler result	Request	Response data
✓ status all vm-hercules04:(mahetov@vm-hercules04)			=== stderr ===
✓ status all vm-hercules05:(mahetov@vm-hercules05)			Loaded: loaded (/etc/systemd/system/hercules-elasticsearch-sink-1.service; preset: disabled) Active: active (running) since Thu 2018-11-22 16:09:08 MSK; 27s ago
✓ restart all vm-hercules04:(mahetov@vm-hercules04)			--
✓ restart all vm-hercules05:(mahetov@vm-hercules05)			Loaded: loaded (/etc/systemd/system/hercules-gate-1.service; enabled; sabled) Active: active (running) since Thu 2018-11-22 16:09:13 MSK; 22s ago
✓ status all vm-hercules04:(mahetov@vm-hercules04)			=== stderr ===
✓ status all vm-hercules05:(mahetov@vm-hercules05)			=== stderr ===

Scroll automatically?

Search: Case sensitive Regular exp.

Что еще делаем JMeter

- ▶ Mock сервисов
- ▶ Мониторинг
- ▶ Запуск Java кода
- ▶ Запуск в CI
- ▶ Тесты из Swagger/Openapi

Mock-сервер с WireMock

```
1 import com.github.tomakehurst.wiremock.WireMockServer;
2 import com.github.tomakehurst.wiremock.client.WireMock;
3 import static com.github.tomakehurst.wiremock.core.WireMockConfiguration.wireMockConfig;
4
5 wireMockServer = new WireMockServer(wireMockConfig().bindAddress("0.0.0.0").port(4321));
6 wireMockServer.start();
7
8 wireMockServer.stubFor(WireMock.get("/test")
9     .willReturn(WireMock.aResponse()
10        .withBody("test response").withStatus(200)));
11
12 wireMockServer.stubFor(WireMock.get("/new")
13     .willReturn(WireMock.aResponse()
14        .withBody("new response").withStatus(203)));
```

Mock-сервер с WireMock

The image shows a screenshot of Apache JMeter (5.1 r1853635) running a test. The JMeter interface displays a 'R223 Sampler' configuration with the following details:

- Name: Start server
- Script language: groovy (Groovy 2.4.16 / Groovy Scripting Engine 2.0)
- Script file (overrides script): File Name: [empty] Browse...
- Script compilation caching: cache compiled script if available:
- Script (variables: ctx vars props SampleResult sampler log Label Filename Parameters args[] OUT):

```
1 import com.github.tomakehurst.wiremock.WireMockServer
2 import com.github.tomakehurst.wiremock.client.WireMock
```

Two browser windows are overlaid on the JMeter interface:

- The top browser window shows the URL `localhost:4321/test` and the response `test response`.
- The bottom browser window shows the URL `localhost:4321/new` and the response `new response`.

The JMeter log output is visible at the bottom, showing the following messages:

```
2019-05-05 15:10:54,283 INFO o.a.j.t.JMeterThread: Thread is done: Thread Group 1-1
2019-05-05 15:10:54,283 INFO o.a.j.t.JMeterThread: Thread finished: Thread Group 1-1
2019-05-05 15:10:54,283 INFO o.a.j.e.StandardJMeterEngine: Notifying test listeners of end of test
2019-05-05 15:10:54,284 INFO o.a.j.g.u.JMeterMenuBar: setRunning(false, *local*)
2019-05-05 15:11:01,341 INFO /: RequestHandlerClass from context returned com.github.tomakehurst.wiremock.http.StubRequestHandl
2019-05-05 15:11:58,267 ERROR WireMock:
Request was not matched
=====
-----
44 | Closest stub | Request |
45 | | |
46 | | |
47 | | |
48 | | |
49 | | |
50 | | |
51 | GET | GET |
52 | /new | /favicon.ico | <<<<< URL doe
53 | | |
```

Мониторинг с Zabbix

The screenshot shows the Apache JMeter 2.11 interface. The window title is "IntPlatformTestBARForPrez.jmx (F:\IntPlatformJMTTest\IntPlatformTestBARForPrez.jmx) - Apache JMeter (2.11 r1554548)". The menu bar includes File, Edit, Search, Run, Options, and Help. The toolbar contains various icons for file operations and execution. The left sidebar shows a tree view of the test plan:

- Тест для проверки интеграционной платформы
 - Общие настройки
 - Thread Group
 - Отправка сообщения
 - Получение сообщения
 - Проверка выполнения
 - Расчет времени выполнения
 - Отправка времени выполнения на сервер**
 - Результаты транзакции
- WorkBench

The right pane displays the configuration for the selected "OS Process Sampler":

- Name: Отправка времени выполнения на сервер
- Comments:
- Command to Execute
 - Command: zabbix_sender
 - Working directory:
- Command parameters
 - Value
 - Z
 - 192.186.16.53
 - s
- Buttons: Detail, Add, Add from Clipboard, Delete, Up
- Environment Variables
 - Name:
 - Value:

Мониторинг с Zabbix

Command to Execute

Command: Working directory:

Command parameters

	Value
-z	
	192.186.16.53
-s	
	WMQ03
-k	
	mq.test.time
-o	
	\${durationTime}

C:\> Командная строка

```
C:\>zabbix_sender -z 192.168.16.53 -s WMQ03 -k mq.test.time -o 30000_
```


CI

```
./jmeter.sh -n -t ELKTest.jmx \  
-l reports/$(date -d "today" +"%%Y%%m%%d_%%H%%M%%S").log \  
-e -o reports/$(date -d "today" +"%%Y%%m%%d_%%H%%M%%S")
```

```
2/2: Run jmeter chain test (Command Line) (8s)  
Step 2/2] Starting: /buildAgent/temp/agentTmp/custom_script4993191551230247798  
Step 2/2] in directory: /buildAgent/work/530983db73e292e0/jmeter/apache-jmeter-4.0/bin  
Step 2/2] Creating summariser <summary>  
Step 2/2] Created the tree successfully using ../../ELKTest_staging_sysmsk.jmx  
Step 2/2] Starting the test @ Thu Oct 11 19:11:31 YEKT 2018 (1539267091335)  
Step 2/2] Waiting for possible Shutdown/StopTestNow/Heapdump message on port 4445  
Step 2/2] summary +      1 in 00:00:00 =   4.3/s Avg:   94 Min:   94 Max:   94 Err:    0 (0.00%) Active: 1 Started: 1 Finished: 0  
Step 2/2] 722536772  
Step 2/2] 722536772  
Step 2/2] summary +      5 in 00:00:05 =   1.0/s Avg:  193 Min:    2 Max:  847 Err:    4 (80.00%) Active: 0 Started: 1 Finished: 1  
Step 2/2] summary =      6 in 00:00:05 =   1.1/s Avg:  177 Min:    2 Max:  847 Err:    4 (66.67%)  
Step 2/2] Tidying up ... @ Thu Oct 11 19:11:37 YEKT 2018 (1539267097491)  
Step 2/2] ... end of run  
Step 2/2] Process exited with code 0
```

```
KT 2018 (1539267091335)
```

```
Heapdump message on port 4445
```

```
g: 94 Min: 94 Max: 94 Err: 0 (0.00%) Active: 1
```

```
g: 193 Min: 2 Max: 847 Err: 4 (80.00%) Active: 0
```

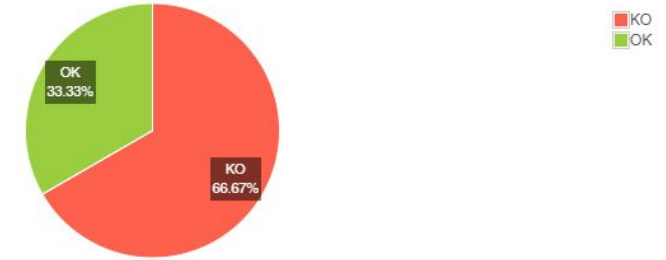
```
g: 177 Min: 2 Max: 847 Err: 4 (66.67%)
```

```
KT 2018 (1539267097491)
```

APDEX (Application Performance Index)

Apdex	T (Toleration threshold)	F (Frustration threshold)	Label
0.250	500 ms	1 sec 500 ms	Total
0.000	500 ms	1 sec 500 ms	Post log to elastic adapter 2018.10.11
0.000	500 ms	1 sec 500 ms	Ping gate
0.000	500 ms	1 sec 500 ms	Get end offset
0.000	500 ms	1 sec 500 ms	Get log from elastic
0.500	500 ms	1 sec 500 ms	Get start offset
1.000	500 ms	1 sec 500 ms	Ping elk-adapter

Requests Summary



Statistics

Requests Label	Executions			Response Times (ms)						Network (KB/sec)		
	#Samples	KO	Error %	Average	Min	Max	90th pct	95th pct	99th pct	Throughput	Received	Sent
Total	6	4	66.67%	177.00	2	847	847.00	847.00	847.00	1.17	0.49	0.15
Get end offset	1	1	100.00%	73.00	73	73	73.00	73.00	73.00	13.70	0.33	0.00
Get log from elastic	1	1	100.00%	23.00	23	23	23.00	23.00	23.00	43.48	9.38	9.04
Get start offset	1	0	0.00%	847.00	847	847	847.00	847.00	847.00	1.18	0.03	0.00
Ping elk-adapter	1	0	0.00%	94.00	94	94	94.00	94.00	94.00	10.64	1.03	1.26
Ping gate	1	1	100.00%	2.00	2	2	2.00	2.00	2.00	500.00	1004.88	0.00
Post log to elastic adapter 2018.10.11	1	1	100.00%	23.00	23	23	23.00	23.00	23.00	43.48	5.01	19.28

Parameters	Description
------------	-------------

Fail build if text matching	<code>Err:\s*[1-9][0-9]*</code> regex appears in build log
-----------------------------	--

Additional Failure Conditions

In this section you can configure build failure depending on various metrics. ⓘ

+ Add failure condition

Type	Parameters Description
Fail build on specific text in build log	Fail build if text matching <code>Err:\s*[1-9][0-9]*</code> regexp appears in build log with message: Some error find

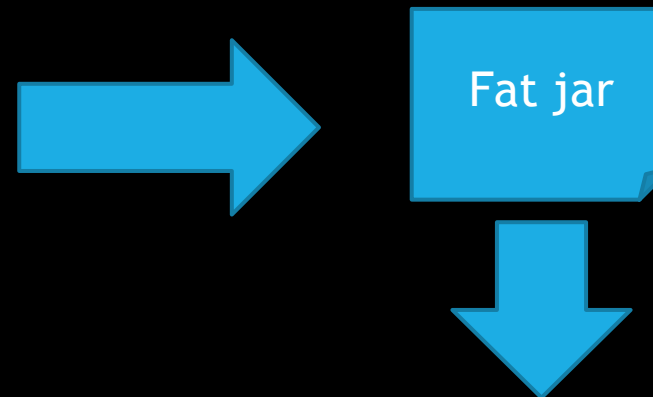
Build number	Status	Changes
#19	❗ Some error find (new) ▾	No changes
#18	✅ Success ▾	No changes
#17	❗ Some error find ▾	No changes
#16	❗ Some error find ▾	Changes (1) ▾
#15	❗ Some error find (new) ▾	Changes (4) ▾
#14	✅ Success ▾	No changes

Запуск кода

```
import ru.kontur.vostok.hercules.protocol.{Event, Variant}
import ru.kontur.vostok.hercules.protocol.encoder.{Encoder, EventWriter}
import ru.kontur.vostok.hercules.protocol.util.EventBuilder
import ru.kontur.vostok.hercules.util.time.TimeUtil
import scala.util.Random

object EventsGenerator {
  def generateEventGroup(count: Int): Array[Byte] = {
    val events = for (_ <- 1 to count) yield generateEvent // generateEvent
    val eventWriter = new EventWriter()
    val stream = new ByteArrayOutputStream
    val encoder = new Encoder(stream)
    encoder.writeInteger(events.length)
    events.foreach(x => eventWriter.write(encoder, x))
    stream.toByteArray
  }

  // Событие размером в 1 КБ (1024 байт)
  def generateEvent: Event = {
    new EventBuilder()
      .setVersion(1)
      .setTimestamp(TimeUtil.millisToTicks(System.currentTimeMillis()))
      .setRandom(UUID.randomUUID())
      .setTag("someRandomString", Variant.ofString(Random.alphanumeric.take(702).mkString))
      .setTag("someString", Variant.ofString("12345"))
      .setTag("someInt", Variant.ofInteger(12345))
  }
}
```



s-gatling-jmeter-tests > jmeter > apache-jmeter-4.0 > lib > ext

Имени

- hercules-protocol-0.20.1-SNAPSHOT.jar
- hercules-util-0.10.1-SNAPSHOT.jar
- scala-hercules-event-builder-assembly-0.1.jar
- ApacheMeter_2.11-1.1.1-SNAPSHOT.jar
- jmeter-plugins-casutg-2.6.jar
- jmeter-plugins-fifo-0.2.jar
- jmeter-plugins-tst-2.5.jar
- jmeter-plugins-ffw-2.0.jar
- jmeter-plugins-functions-2.1.jar
- jmeter-plugins-perfmon-2.1.jar
- jmeter-plugins-graphs-additional-2.0.jar

Запуск кода

```
1 s = EventGenerator.generateEventGroup(2000)
2
3 new File('bd.dat').withOutputStream
4 {
5     it.write(s)
6 }
```

Swagger/Openapi

The screenshot shows the Swagger Editor interface at <https://editor.swagger.io>. The interface is divided into two main sections: a code editor on the left and a client library selection grid on the right.

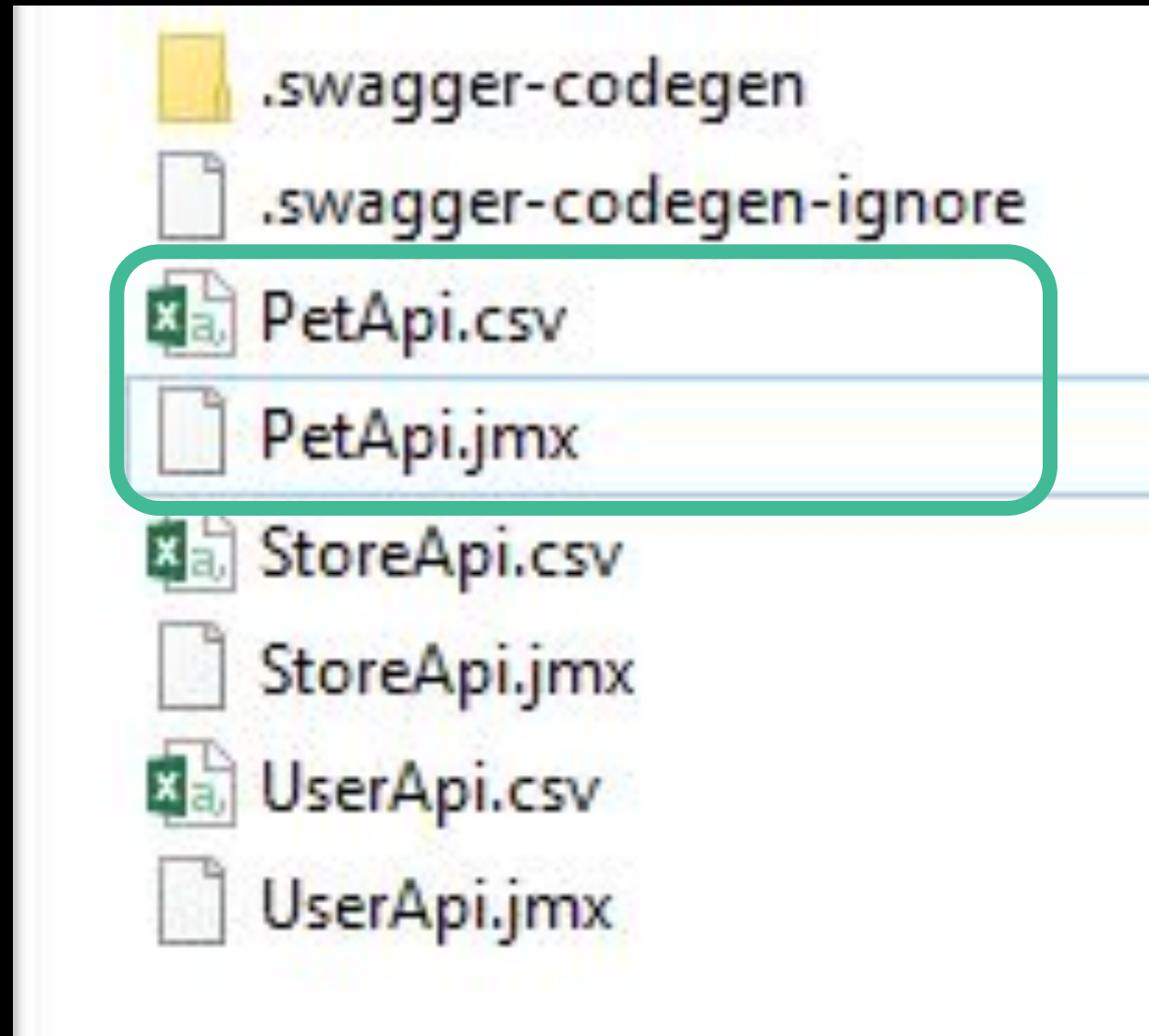
Code Editor (Left): Displays OpenAPI schema for a `deletePet` endpoint. The path is `/pet/{petId}`. The `parameters` section includes a required integer parameter `petId` with format `int64`. The `responses` section includes a `200` response with description "pet response" and a schema reference to `#/definitions/Pet`, and a `default` response with description "unexpected error" and a schema reference to `#/definitions/ModelError`. The `operationId` is `deletePet`.

```
in: "path"
description: "ID of pet to fetch"
required: true
type: "integer"
format: "int64"
responses:
  "200":
    description: "pet response"
    schema:
      $ref: "#/definitions/Pet"
  default:
    description: "unexpected error"
    schema:
      $ref: "#/definitions/ModelError"
description: "deletes a single pet based on ID supplied"
operationId: "deletePet"
parameters:
```

Client Library Selection Grid (Right): A grid of client libraries for various languages. The `jmeter` client is highlighted with a red box.

ada	elixir	jmeter	scalaz
akka-scala	elm	kotlin	swagger
android	erlang-client	lua	swagger-yaml
apex	flash	objc	swift
bash	go	perl	swift3
clojure	groovy	php	swift4
cpprest	haskell-http-client	powershell	tizen
csharp	html	python	typescript-angular
csharp-dotnet2	html2	qt5cpp	typescript-angularjs

Swagger/Openapi



Swagger/Openapi

PetApi.jmx (D:\Преза\SqaDays май 2019\jmeter-client-generated\jmeter-client\PetApi.jmx) - Apache JMeter (4.0 r1823414)

File Edit Search Run Options Help

00:00:00 0 0/0

PetApi Test Plan

- User Defined Variables
- HTTP Request Defaults
- Thread Group - addPet
 - HTTP Header Manager
 - addPet - \${testCase}
 - Load CSV Test Data - addPet
 - HTTP Status Assertion
- Thread Group - deletePet
- Thread Group - findPetsByStatus
- Thread Group - findPetsByTags
- Thread Group - getPetById
 - HTTP Header Manager
 - getPetById - \${testCase}
 - HTTP Status Assertion
- Thread Group - updatePet
- Thread Group - updatePetWithForm
- Thread Group - uploadFile
- View Results Tree

User Defined Variables

Name: User Defined Variables

Comments:

Name:	Value	Description
threads	\${__P(threads, 1)}	
rampup	\${__P(rampup, 1)}	
duration	\${__P(duration, 1)}	
testCases	\${__P(testCases, 10)}	
host	\${__P(host, localhost)}	
port	\${__P(port, 8080)}	
testData.addPetFile	\${__P(testData.addPetFile, PetApi.csv)}	
testData.deletePetFile	\${__P(testData.deletePetFile, PetApi.csv)}	
testData.findPetsByStatusFile	\${__P(testData.findPetsByStatusFile, PetApi.csv)}	
testData.findPetsByTagsFile	\${__P(testData.findPetsByTagsFile, PetApi.csv)}	
testData.getPetByIdFile	\${__P(testData.getPetByIdFile, PetApi.csv)}	
testData.updatePetFile	\${__P(testData.updatePetFile, PetApi.csv)}	
testData.updatePetWithFormFile	\${__P(testData.updatePetWithFormFile, PetApi.c...}	
testData.uploadFileFile	\${__P(testData.uploadFileFile, PetApi.csv)}	

Detail Add Add from Clipboard Delete Up Down

Swagger/Openapi

Name:	Value
threads	<code>\${_P(threads,1)}</code>
rampup	<code>\${_P(rampup,1)}</code>
duration	<code>\${_P(duration,1)}</code>
testCases	<code>\${_P(testCases,10)}</code>
host	<code>\${_P(host,localhost)}</code>
port	<code>\${_P(port,8080)}</code>
testData.addPetFile	<code>\${_P(testData.addPetFile,PetApi.csv)}</code>
testData.deletePetFile	<code>\${_P(testData.deletePetFile,PetApi.csv)}</code>
testData.findPetsByStatusFile	<code>\${_P(testData.findPetsByStatusFile,PetApi.csv)}</code>
testData.findPetsByTagsFile	<code>\${_P(testData.findPetsByTagsFile,PetApi.csv)}</code>
testData.getPetByIdFile	<code>\${_P(testData.getPetByIdFile,PetApi.csv)}</code>
testData.updatePetFile	<code>\${_P(testData.updatePetFile,PetApi.csv)}</code>
testData.updatePetWithFormFile	<code>\${_P(testData.updatePetWithFormFile,PetApi.c...}</code>
testData.uploadFileFile	<code>\${_P(testData.uploadFileFile,PetApi.csv)}</code>

Swagger/Openapi

```
PetApi.csv x
```

1	testCase, httpStatusCode, body, petId, apiKey, status, tags, petId,
2	Success, 200, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0

Swagger/Openapi

HTTP Request

Name:

Comments: Find pet by ID Returns a single pet

Basic

Web Server

Protocol [http]: Server Name or IP:

HTTP Request

Method: Path:

Redirect Automatically Follow Redirects Use KeepAlive

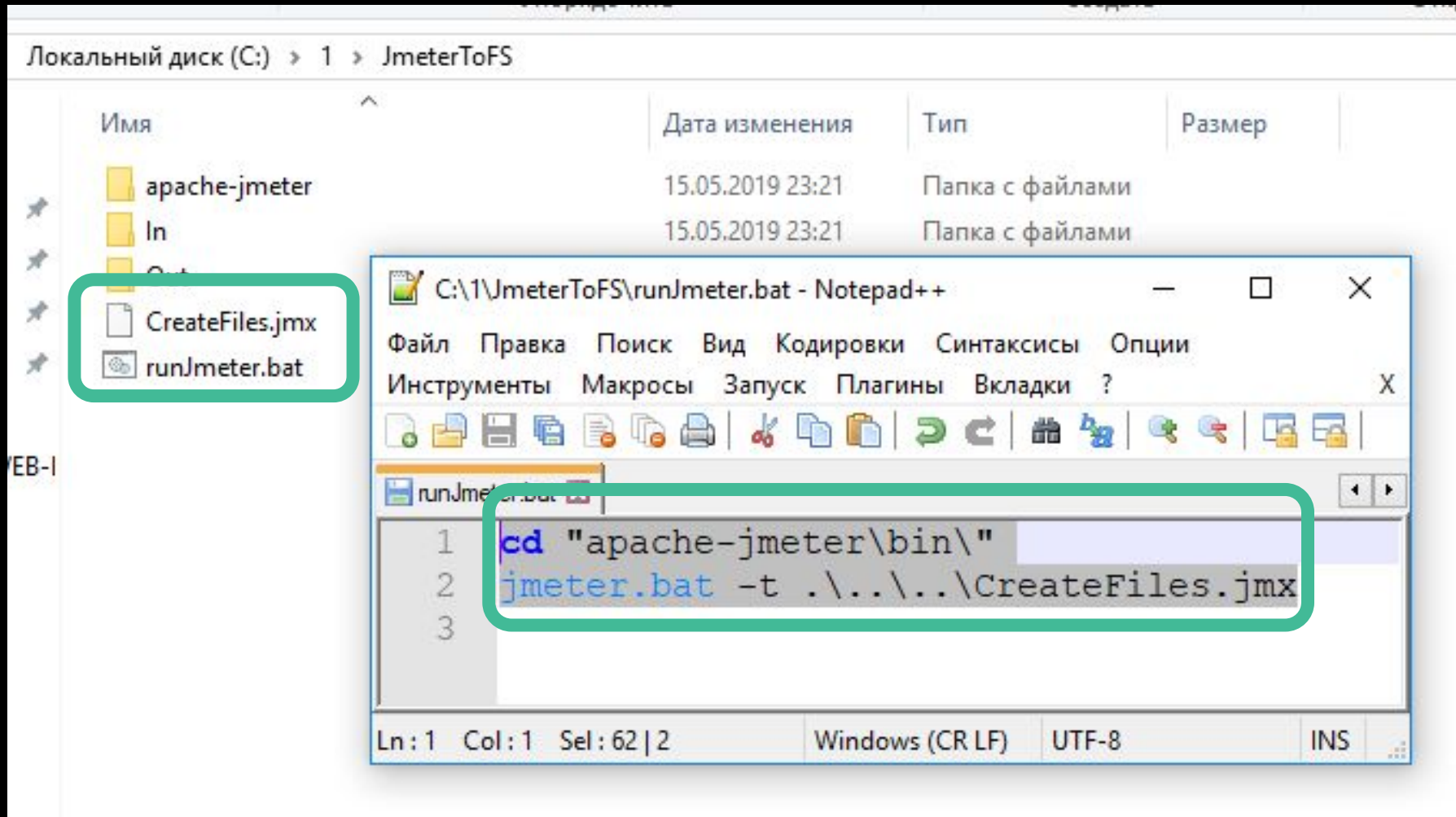
Еще используем для работы с

- ▶ Чтением-записью файлов
- ▶ JDBC
- ▶ JMS / AMQP

Еще классные особенности

- ▶ Кроссплатформенность
- ▶ Установка копированием
- ▶ Быстрый старт

Запуск конкретного сценария



Запуск конкретного сценария

The screenshot shows the Apache JMeter 2.11.1 interface. The title bar indicates the file path: `(C:\1\JmeterToFS\apache-jmeter\bin\..\..\CreateFiles.jmx) - Apache JMeter (2.11.1)`. The menu bar includes **File**, **Edit**, **Search**, **Run**, **Options**, and **Help**. The toolbar contains icons for file operations (New, Open, Save, Print, Copy, Paste, Undo, Redo) and execution (Play, Stop, Pause).

The left sidebar shows a tree view of the test plan:

- Генерация проезда
- Переменные
- Thread Group
 - Counter
 - Генерация переменных (highlighted)
 - Генерация переменных
 - Запись сообщения \${file}
- View Results Tree
- Temp
- WorkBench

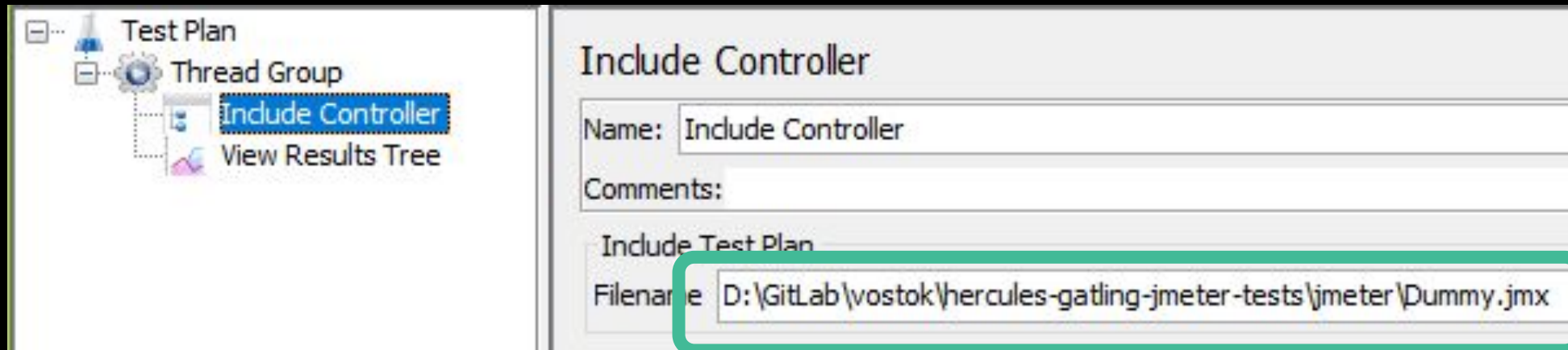
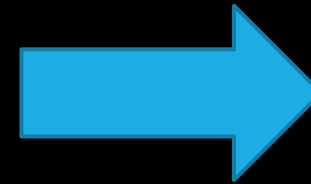
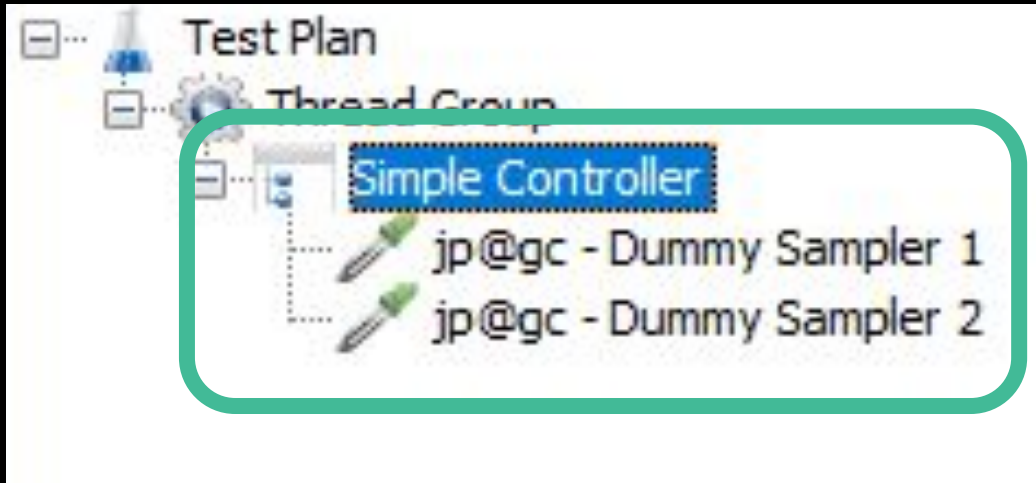
The right pane displays the configuration for the selected element, **User Parameters**:

Name: Генерация переменных (Динамика)
Comments:
 Update Once Per Iteration

Parameters

Name:	
IDBetamount	\${__BeanShell(System.currentTim
ExcessFactDate	\${__BeanShell(new java.text.Simp
fileName	\${__BeanShell(new java.text.Simp

Повторное использование блоков



Повторное использование блоков

The screenshot displays the JMeter GUI. On the left, the Test Plan tree structure is visible, showing a Thread Group containing an Include Controller and a View Results Tree element. The View Results Tree element is highlighted with a blue dashed border. On the right, the configuration panel for the View Results Tree is shown. The Name field is set to "View Results Tree". The "Write results to file / Read from file" section includes a "Filename" field and a "Browse..." button. Below this is a "Search:" field with a search icon. The "Text" section shows a list of results, with two entries highlighted by a green rounded rectangle:

- jp@gc - Dummy Sampler 1
- jp@gc - Dummy Sampler 2

Повторное использование блоков

The image displays the JMeter GUI interface, illustrating the reuse of blocks. On the left, a Test Plan contains a Thread Group with two Module Controller elements and a View Results Tree element. The top Module Controller is highlighted with a green rounded rectangle. On the right, the configuration for a Module Controller is shown. The 'Name' field is set to 'Module Controller'. Below the 'Find target element' button, the 'Module To Run' section contains a Test Plan with a Thread Group containing an Include Controller element, which is also highlighted with a green rounded rectangle.

Повторное использование блоков

The image displays the JMeter GUI interface, illustrating the reuse of blocks. On the left, a Test Plan contains a Thread Group with four elements: Include Controller, Module Controller, Module Controller, and View Results Tree. The second Module Controller is highlighted with a green rounded rectangle. On the right, the configuration panel for a Module Controller is shown. It includes fields for Name (Module Controller) and Comments, a Find target element button, and a Module To Run section. The Module To Run section contains a Test Plan with a Thread Group, and the Include Controller element within that Thread Group is highlighted with a green rounded rectangle.

Повторное использование блоков

The screenshot displays the JMeter GUI. On the left, the Test Plan tree structure is visible, with the 'View Results Tree' element selected and highlighted in blue. The right pane shows the configuration for the 'View Results Tree' element, including a search field and a list of results. The results list is highlighted with a green rounded rectangle and contains the following entries:

- ✓ jp@gc - Dummy Sampler 1
- ✓ jp@gc - Dummy Sampler 2
- ✓ jp@gc - Dummy Sampler 1
- ✓ jp@gc - Dummy Sampler 2
- ✓ jp@gc - Dummy Sampler 1
- ✓ jp@gc - Dummy Sampler 2

Командная работа

- ▶ ~ 262 jmx
- ▶ ~ 5337 Test cases (xml)
- ▶ Описанная структура тестов
- ▶ Регламент создания новых и изменения текущих тестов
- ▶ Инструкции для запуска



Минусы и проблемы

- ▶ Сценарий - xml файл
- ▶ При работе с GIT сложно мержить изменения
- ▶ Переменные - строки
- ▶ Неудобно работать с бинарными http протоколами
- ▶ Иногда требуется понимание JVM
- ▶ Нельзя отменить изменения ctrl-z
- ▶ Scala в JSR223 работает плохо
- ▶ ...

Границы применимости и выбор инструмента

- ▶ Слишком громоздкий для некоторых задач
- ▶ Вклад в изучение
- ▶ Может быть хуже специализированных инструментов
- ▶ Гибкость в некоторых моментах хуже, чем у кода



Спасибо за внимание! Остались вопросы?

- ▶ Махетов Сергей
- ▶ E-mail: Profitfx@mail.ru
- ▶ Telegram: @Mahetovs
- ▶ <https://tech.kontur.ru/>

