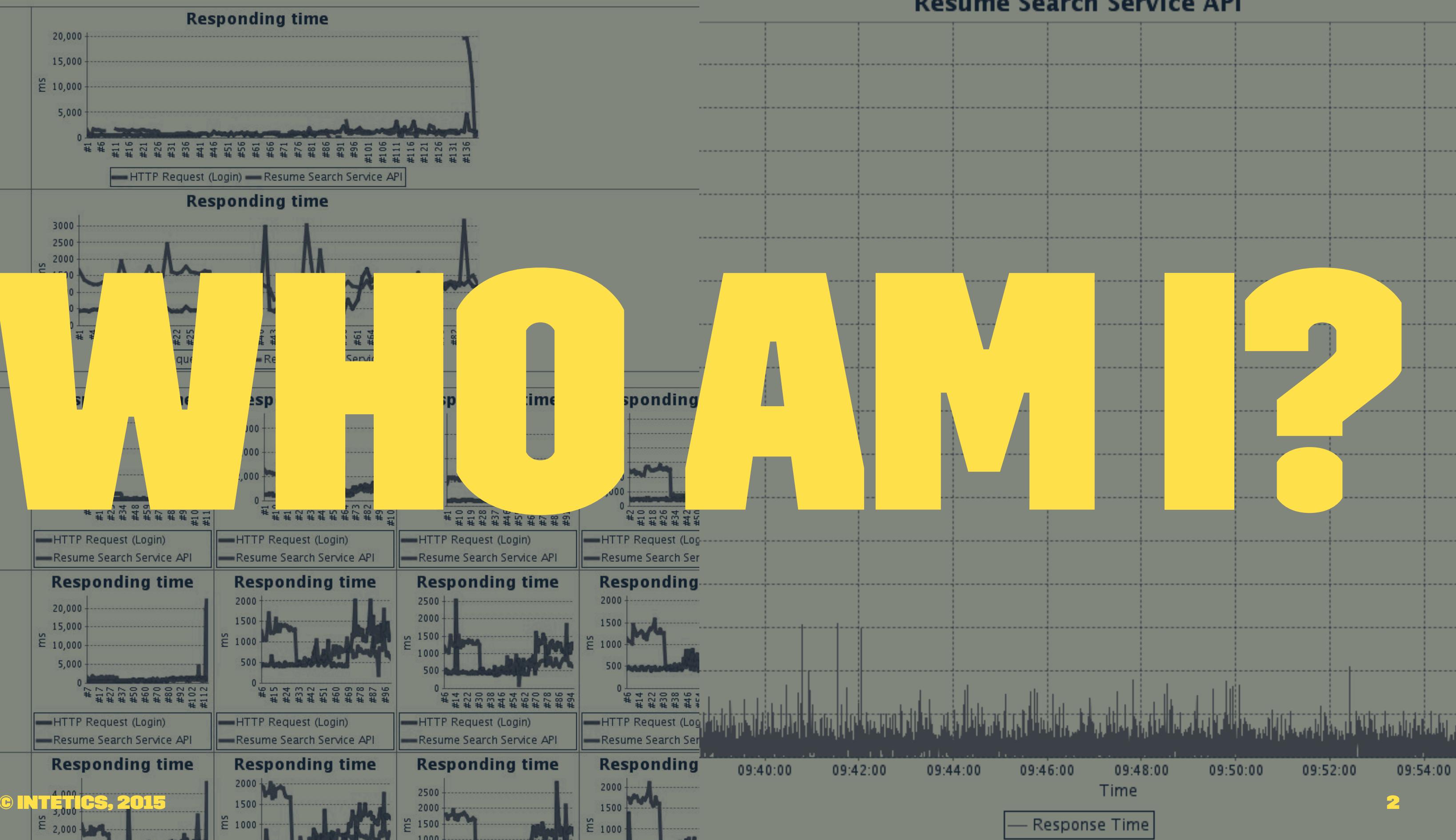


PERFORMANCE TESTING FOR EVERYONE



Resume Search Service API

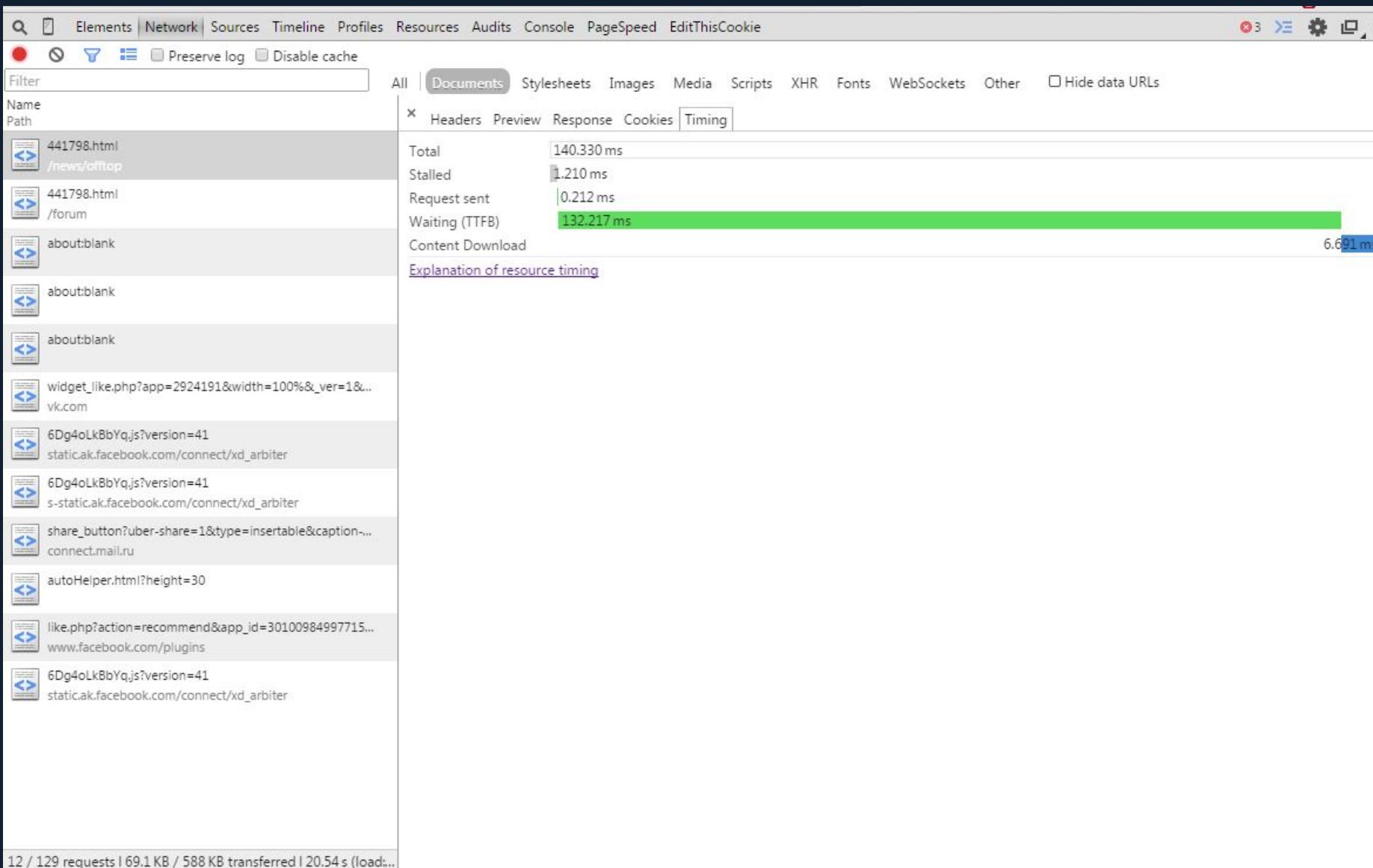


NAIVE APPROACH





A BIT SMARTER APPROACH CHROME

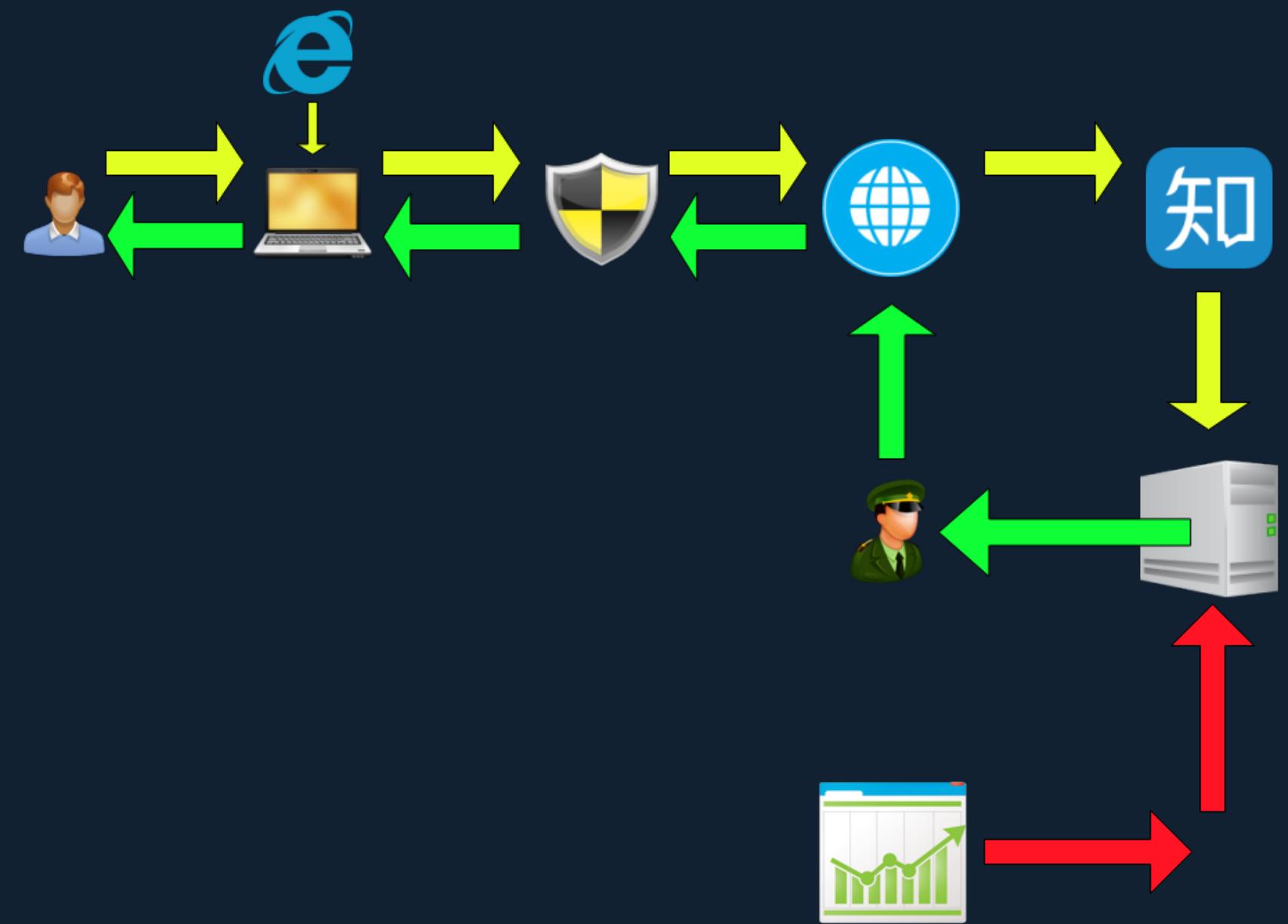


A BIT SMARTER APPROACH FIREFOX / FIREBUG

The screenshot shows the Firefox Firebug extension's Net panel. The panel has tabs for XHR, Clear, Persist, All, HTML, CSS, JavaScript, XHR, Images, Plugins, Media, and Fonts. The All tab is selected. The main area displays a table of network requests with columns for URL, Status, Domain, Size, Remote IP, and Timeline. A tooltip is open over the Timeline column, providing a detailed breakdown of the request phases and their durations relative to the request start.

URL	Status	Domain	Size	Remote IP	Timeline
+ GET 441798.html	200 OK	auto.tut.by	22.5 KB	178.124.133.66:80	845ms
+ GET style~modal~article~r1791	200 OK	news.tut.by	52.0 KB	178.124.133.65:80	0ms Request start time since the beginning
+ GET auto~r17979~r9583~r1616	200 OK	auto.tut.by	6.0 KB	178.124.133.166:80	Request phases start and elapsed time relative to the request start:
+ GET jquery-1.8.3.min~moderni..	200 OK	tut.by	125.7 KB	178.124.133.66:80	DNS Lookup 0ms 2ms
+ GET core~r17977~r9584~r1472	200 OK	news.tut.by	12.4 KB	178.124.133.65:80	Connecting +2ms 18ms
+ GET tut~tellnews~r17977~r958	304 Not Modified	auto.tut.by	5.0 KB	178.124.133.166:80	Sending +20ms 0ms
+ GET catalog_suggests~r17977~	304 Not Modified	catalog.tut.by	9.6 KB	178.124.133.3:80	Waiting +20ms 160ms
+ GET xgemius.js	200 OK	tut.by	1.7 KB	178.124.133.66:80	Receiving +180ms 665ms
+ GET flashDetect.js?ver=2.5	304 Not Modified	reklama.tut.by	1.5 KB	178.124.133.67:80	Event timing relative to the request start:
+ GET 29634_728x90_A.png	200 OK	bugaga.tut.by	16.0 KB	178.124.133.67:80	+3.94s 'DOMContentLoaded' (event)
+ GET show_ads.js	304 Not Modified	pagead2.googlesyndication.com	7.6 KB	216.58.208.66:80	+11.09s 'load' (event)
+ GET navs-ads~r17979.css	200 OK	tut.by	165 B	178.124.133.65:80	1.0ms
+ GET ga~scrolldepth~r17977.js	200 OK	tut.by	3.9 KB	178.124.133.65:80	
+ GET ba.png	304 Not Modified	img.tvt.by	2.6 KB	178.124.133.31:80	

FUNNY NETWORK DIAGRAM



RIGHT SOULTION

JMETER

AGENDA

- » elements of a performance test
- » what is a load
- » testing RESTful services
- » testing 'like a web browser'
- » passing variables to performance tests
- » configuring Jenkins to run performance tests

AGENDA (EXPLAINED)

...

- testing RESTful services
 - authentication
 - sampling
 - variables VS properties
 - custom errors
 - logging

...

AGENDA (EXPLAINED)

...

- testing 'like a web browser'
 - authentication
 - session support
 - sampling

...

PREREQUISITS

- » java 1.7
- » maven 3
- » git-scm or cygwin for Windows users
- » jenkins

LET'S START¹

petapp (auth and search RESTful services)

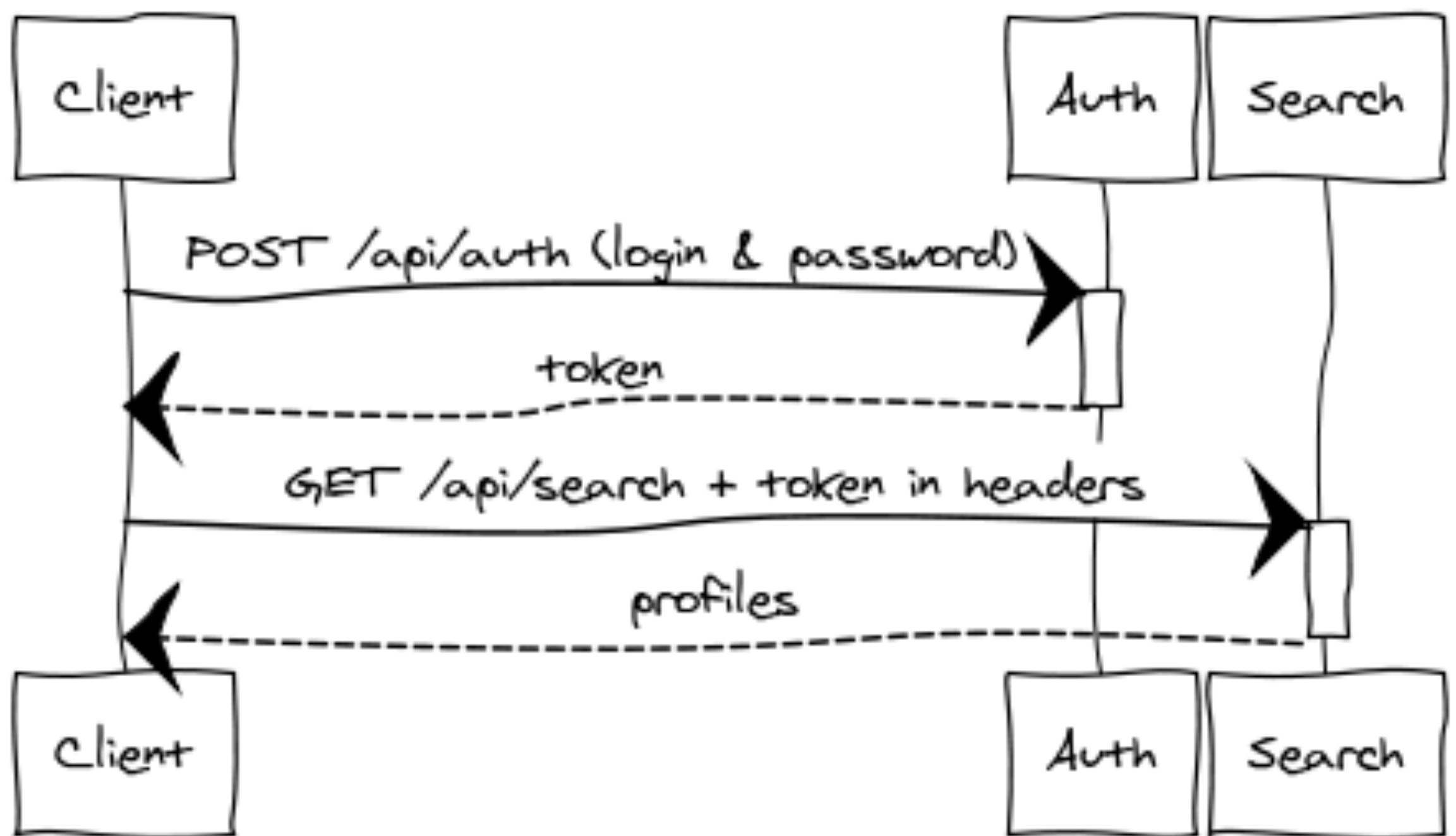
```
petapp$ mvn spring-boot:run
```

petweb (web app with login form)

```
petweb$ mvn spring-boot:run -Dserver.port=9090
```

¹ showtime

authentication & search



www.websequencediagrams.com

AUTHENTICATION

```
$ curl localhost:8080/api/auth -d 'login=user&password=user' -X POST
```

RESPONSE

```
<AuthToken xmlns="">
  <token>pprl2c2u694b9hi2is7cqkalva</token>
  <expirationDate>1427717457176</expirationDate>
</AuthToken>
```

SEARCH REQUEST

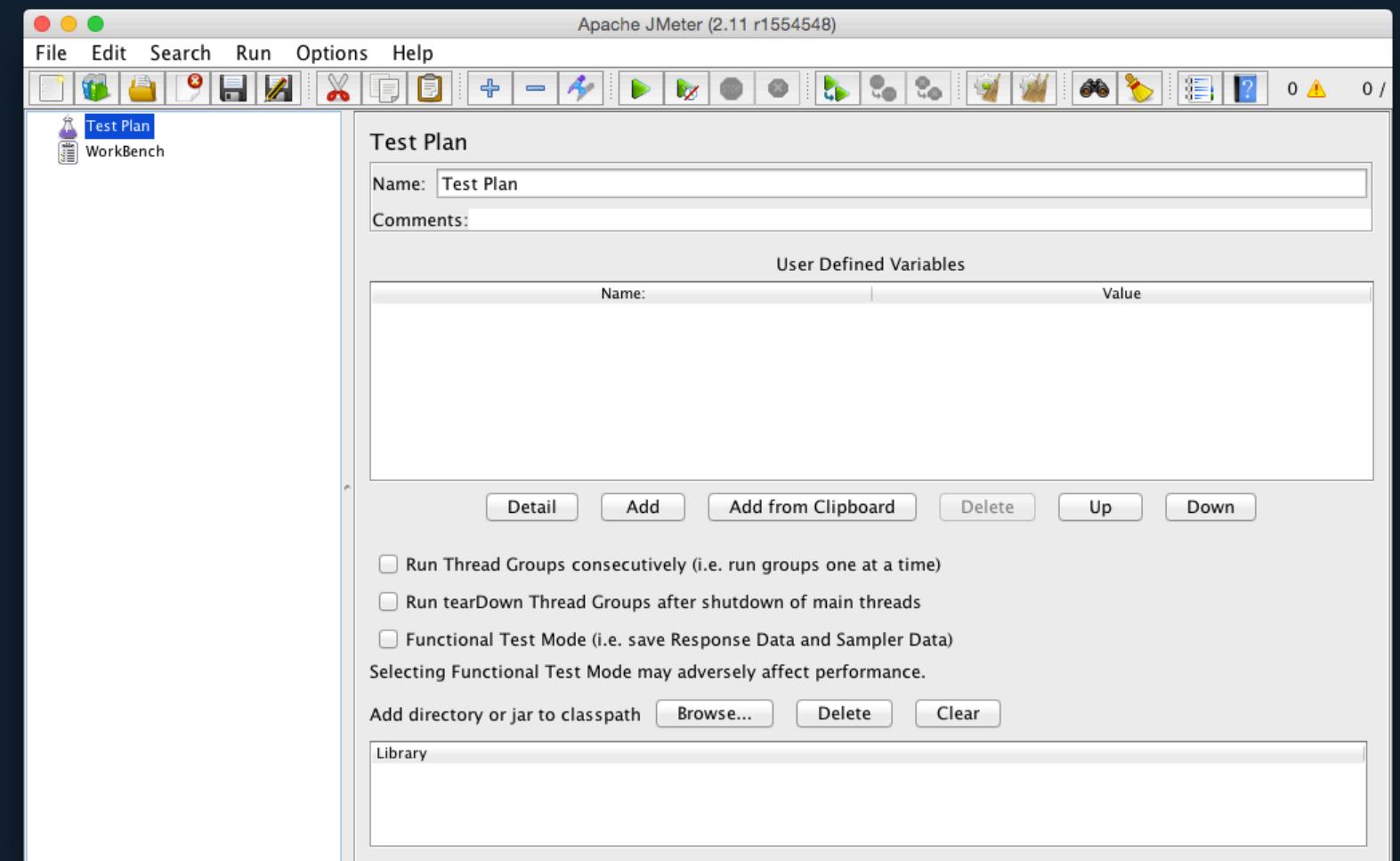
```
$ curl localhost:8080/api/search -H "token: pprl2c2u694b9hi2is7cqkalva"
```

SEARCH RESPONSE

```
{  
  "page": 1,  
  "items": [  
    {  
      "id": 6592,  
      "title": "Oil & Gas Drilling Supervisor",  
      "content": "B",  
      "location": "C"  
    },  
    ...  
  ]  
}
```

JMETER GUI

tests\$ mvn jmeter:gui



AGENDA

» elements of a performance test

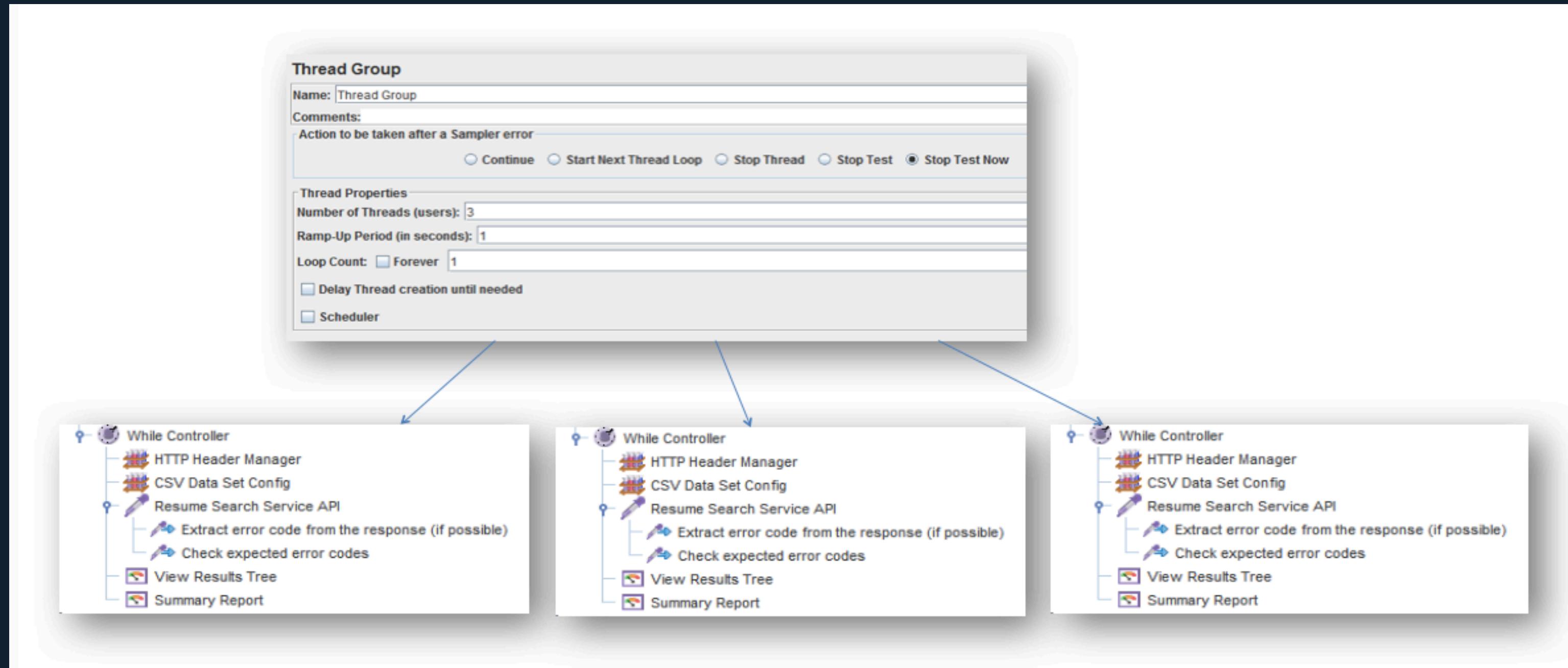
...
...

THREAD GROUP

“Thread group elements are the beginning points of any test plan. All controllers and samplers must be under a thread group. Other elements, e.g. Listeners, may be placed directly under the test plan, in which case they will apply to all the thread groups.”

Jmeter documentation

THREAD GROUP



CONTROLLERS

“JMeter has two types of Controllers: Samplers and Logical Controllers. These drive the processing of a test. Samplers tell JMeter to send requests to a server. Logical Controllers let you customize the logic that JMeter uses to decide when to send requests.”

JMeter documentation

NECESSARY CONTROLLERS

Once Only Controller Name: Once Only Controller Comments:	Runtime Controller Name: Runtime Controller Comments: Runtime (seconds) 60	While Controller Name: While Controller Comments: Condition (function or variable)
--	--	--

AGENDA

...

- what is a load

...

1000000
users



THREADS

Sequence				Partially or fully crossed			
Time	Thread #1	Thread #2	Thread #3	Time	Thread #1	Thread #2	Thread #3
0:00:00				0:00:00	work		
0:00:01	work			0:00:01	work		
0:00:02				0:00:02			
0:00:03		work		0:00:03		work	
0:00:04				0:00:04			
0:00:05				0:00:05	work		
0:00:06				0:00:06			
0:00:07			work	0:00:07			work
0:00:08				0:00:08			
0:00:09	work			0:00:09			
0:00:10	work			0:00:10		work	
0:00:11				0:00:11			
0:00:12				0:00:12			
0:00:13		work		0:00:13	work		
0:00:14				0:00:14			work
0:00:15				0:00:15			
0:00:16				0:00:16			
0:00:17			work	0:00:17	work	work	
0:00:18				0:00:18			
0:00:19	work			0:00:19			
0:00:20				0:00:20			work
0:00:21				0:00:21	work		
0:00:22				0:00:22		work	
0:00:23				0:00:23			
0:00:24				0:00:24			
0:00:25				0:00:25			

RAMP-UP PERIOD

“Ramp-up Period - How long JMeter should take to get all the threads started. If there are 10 threads and a ramp-up time of 100 seconds, then each thread will begin 10 seconds after the previous thread started, for a total time of 100 seconds to get the test fully up to speed.”

JMeter documentation

RAMP-UP PERIOD

Thread Group

Name: Thread Group

Comments: Action to be taken after a Sampler error
 Continue Start Next Thread Loop Stop

Thread Properties

Number of Threads (users): 3

Ramp-Up Period (in seconds): 10

Loop Count: Forever 1

Delay Thread creation until needed

Scheduler

Ramp-Up Period

Sequence			
Time	Thread #1	Thread #2	Thread #3
0:00:00			
0:00:01	work		
0:00:02			
0:00:03			
0:00:04		work	
0:00:05			work
0:00:06	work		
0:00:07			
0:00:08			
0:00:09			
0:00:10	work		work
0:00:11		work	
0:00:12			work
0:00:13			
0:00:14			
0:00:15			

AGENDA (EXTENDED)¹

...

- testing RESTful services
 - authentication
 - sampling

...

¹ showtime

AUTHENTICATION

Regular Expression Extractor

Name: Extract token from response body

Comments:

Apply to:

Main sample and sub-samples Main sample only Sub-samples only JMeter Variable

Response Field to check

Body Body (unesaped) Body as a Document Headers URL Response Code Response Message

Reference Name: token

Regular Expression: <token>(.*?)</token>

Template: \$1\$

Match No. (0 for Random):

Default Value: INVALID_TOKEN

AUTHENTICATION

BeanShell PostProcessor

Name: Store token in properties

Comments:

Reset bsh.Interpreter before each call

Reset Interpreter: False

Parameters to be passed to BeanShell (=> String Parameters and String []bsh.args)

Parameters:

Script file (overrides script)

File Name:

Script (variables: ctx vars props prev data log)

Script:

```
1 ${__setProperty(token, ${token})};
```

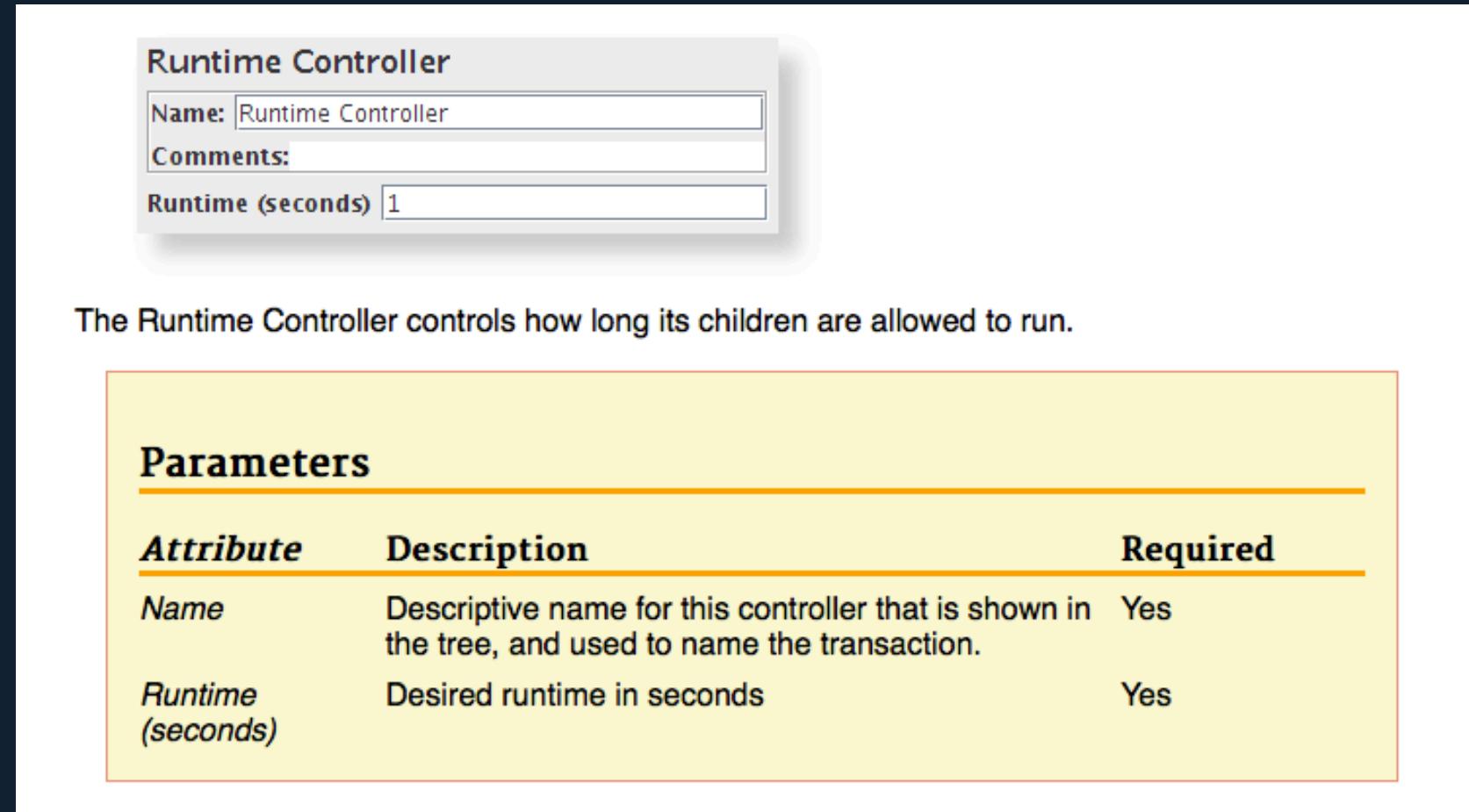
PASSING HEADERS

The screenshot shows the 'HTTP Header Manager' configuration dialog. It has fields for 'Name' (set to 'HTTP Header Manager') and 'Comments'. Below these, a table titled 'Headers Stored in the Header Manager' lists a single header named 'token' with a value of \${__P(token)}.

Name:	Value
token	$\${__P(token)}$

“The Header Manager lets you add or override HTTP request headers.”

HOW TO RUN A TEST PLAN FOR A CERTAIN AMOUNT OF TIME



The screenshot shows the 'Runtime Controller' configuration dialog. It includes fields for 'Name' (set to 'Runtime Controller'), 'Comments', and 'Runtime (seconds)' (set to '1'). Below the dialog, a descriptive text states: 'The Runtime Controller controls how long its children are allowed to run.' A table titled 'Parameters' provides detailed information about the controller's attributes:

Attribute	Description	Required
Name	Descriptive name for this controller that is shown in the tree, and used to name the transaction.	Yes
Runtime (seconds)	Desired runtime in seconds	Yes

JMeter documentation

CSV DATA SET CONFIG

CSV Data Set Config

The screenshot shows the 'CSV Data Set Config' dialog box. It includes fields for 'Name' (set to 'CSV Data Set Config') and 'Comments'. Below these, there's a section titled 'Configure the CSV Data Source' with the following settings:

- Filename: csvdata.txt
- File encoding: (empty)
- Variable Names (comma-delimited): a,b,c
- Delimiter (use '\t' for tab): ,
- Allow quoted data?: False
- Recycle on EOF ?: True
- Stop thread on EOF ?: False
- Sharing mode: All threads

Below the dialog, a descriptive text explains the purpose of the CSV Data Set Config:

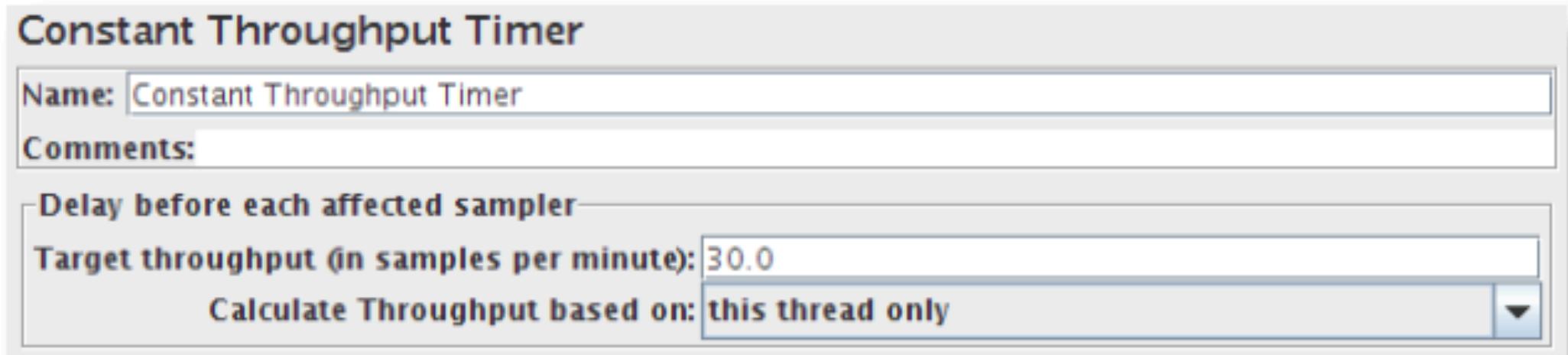
CSV Data Set Config is used to read lines from a file, and split them into variables. It is easier to use than the __CSVRead() and _StringFromFile() functions. It is well suited to handling large numbers of variables, and is also useful for testing with "random" and unique values. Generating unique random values at run-time is expensive in terms of CPU and memory, so just create the data in advance of the test. If necessary, the "random" data from the file can be used in conjunction with a run-time parameter to create different sets of values from each run - e.g. using concatenation - which is much cheaper than generating everything at run-time.

Versions of JMeter after 2.3.1 allow values to be quoted; this allows the value to contain a delimiter. Previously it was necessary to choose a delimiter that was not used in any values. If "allow quoted data" is enabled, a value may be enclosed in double-quotes. These are removed. To include double-quotes within a quoted field, use two double-quotes. For example:

JMeter documentation

CONSTANT THROUGHPUT TIMER

Constant Throughput Timer



This timer introduces variable pauses, calculated to keep the total throughput (in terms of samples per minute) as close as possible to a give figure. Of course the throughput will be lower if the server is not capable of handling it, or if other timers or time-consuming test elements prevent it.

Jmeter documentation

CONSTANT THROUGHPUT TIMER

this thread only - each thread will try to maintain the target throughput. The overall throughput will be proportional to the number of active threads.

all active threads in current thread group - the target throughput is divided amongst all the active threads in the group. Each thread will delay as needed, based on when it last ran.

CONSTANT THROUGHPUT TIMER

all active threads - the target throughput is divided amongst all the active threads in all Thread Groups. Each thread will delay as needed, based on when it last ran. In this case, each other Thread Group will need a Constant Throughput timer with the same settings.

CONSTANT THROUGHPUT TIMER

all active threads in current thread group (shared) - as above, but each thread is delayed based on when any thread in the group last ran.

all active threads (shared) - as above; each thread is delayed based on when any thread last ran.

AGENDA (EXPLAINED)

- ...
- testing RESTful services
- ...
- variables VS properties
- ...

USER DEFINED VARIABLES

Test Plan

Name: Test Plan

Comments:

User Defined Variables	
Name:	Value
duration	<code>__P(duration,30)</code>
throughput	<code>__P(throughput,1000)</code>

Detail Add Add from Clipboard Delete Up Down

Run Thread Groups consecutively (i.e. run groups one at a time)

USER DEFINED VARIABLES

Runtime Controller

Name:	Runtime Controller
Comments:	
Runtime (seconds)	<code> \${duration}</code>

Constant Throughput Timer

Name:	Constant Throughput Timer
Comments:	
Delay before each affected sampler	
Target throughput (in samples per minute):	<code> \${throughput}</code>
Calculate Throughput based on:	<code> all active threads i</code>

PASSING VARIABLES INTO TESTS

POM.XML

```
<!--  
     User-defined variables  
-->  
<propertiesUser>  
    <threads>${threads}</threads>  
    <duration>${duration}</duration>  
    <throughput>${throughput}</throughput>  
</propertiesUser>
```

PASSING VARIABLES INTO TESTS

```
$ mvn verify \
-Dduration=60 \
-Dthroughput=2000 \
-Dthreads=200
etc...
```

AGENDA (EXPLAINED)

...

- testing RESTful services

...

- custom errors
- logging

PROCESSING CUSTOM ERRORS

```
if (prev.getResponseCode().equals("400") &&
    ( vars.get("error").equals("PAGE_IS_NOT_VALID")
    || vars.get("error").equals("LOCATION_ID_IS_NOT_VALID")
    )
)
{
    prev.setResponseOK(); // It's expected error, we have to ignore it.
}
```

LOGGING

```
// logging into log file
log.info("Server returns status code 400 for the URL: "
    + prev.getUrlAsString()
    + ". Root cause: " + vars.get("error"));

// logging into console
System.out.println("Server returns status code 400 for the URL: "
    + prev.getUrlAsString()
    + ". Root cause: " + vars.get("error"));
```

ALL TOGETHER

```
if (prev.getResponseCode().equals("400") &&
    ( vars.get("error").equals("PAGE_IS_NOT_VALID")
    || vars.get("error").equals("LOCATION_ID_IS_NOT_VALID"))
    )
) {
    log.info("Server returns status code 400 for the URL: "
        + prev.getUrlAsString()
        + ". Root cause: " + vars.get("error"));
    System.out.println("Server returns status code 400 for the URL: "
        + prev.getUrlAsString()
        + ". Root cause: " + vars.get("error"));
    prev.setResponseOK(); // It's expected error (business error), we have to ignore it.
} else if (prev.getResponseCode().equals("400") || prev.getResponseCode().equals("500")) {
    log.info("!!! Unknown server error for the URL: "
        + prev.getUrlAsString()
        + ". Root cause: " + vars.get("error"));
    System.out.println("!!! Unknown server error for the URL: "
        + prev.getUrlAsString()
        + ". Root cause: " + vars.get("error"));
}
```

AGENDA (EXPLAINED)

...

- testing 'like a web browser'
 - authentication
 - session support
 - sampling

...

“JMeter is not a browser.”

Jmeter documentation

As far as web-services and remote services are concerned, JMeter looks like a browser (or rather, multiple browsers); however JMeter does not perform all the actions supported by browsers. In particular, JMeter does not execute the Javascript found in HTML pages. Nor does it render the HTML pages as a browser does (it's possible to view the response as HTML etc, but the timings are not included in any samples, and only one sample in one thread is ever viewed at a time).

WEB APP (LOGIN FORM)

Thymeleaf - Plain [Home](#)

Login with Username and Password

Username Password

WEB APP (HOME)

A screenshot of a web application interface. At the top, there is a dark header bar with the text "Thymeleaf - Plain" on the left, and "Home" and "Logout" on the right. Below the header, the main content area has a white background. It features a large, bold, dark blue header "Hello Home". Underneath it, the text "Hello World" is displayed in a smaller, dark blue font. Below that, the date and time "March 30, 2015 5:02:43 PM FET" are shown in a smaller, dark blue font.

CSRF

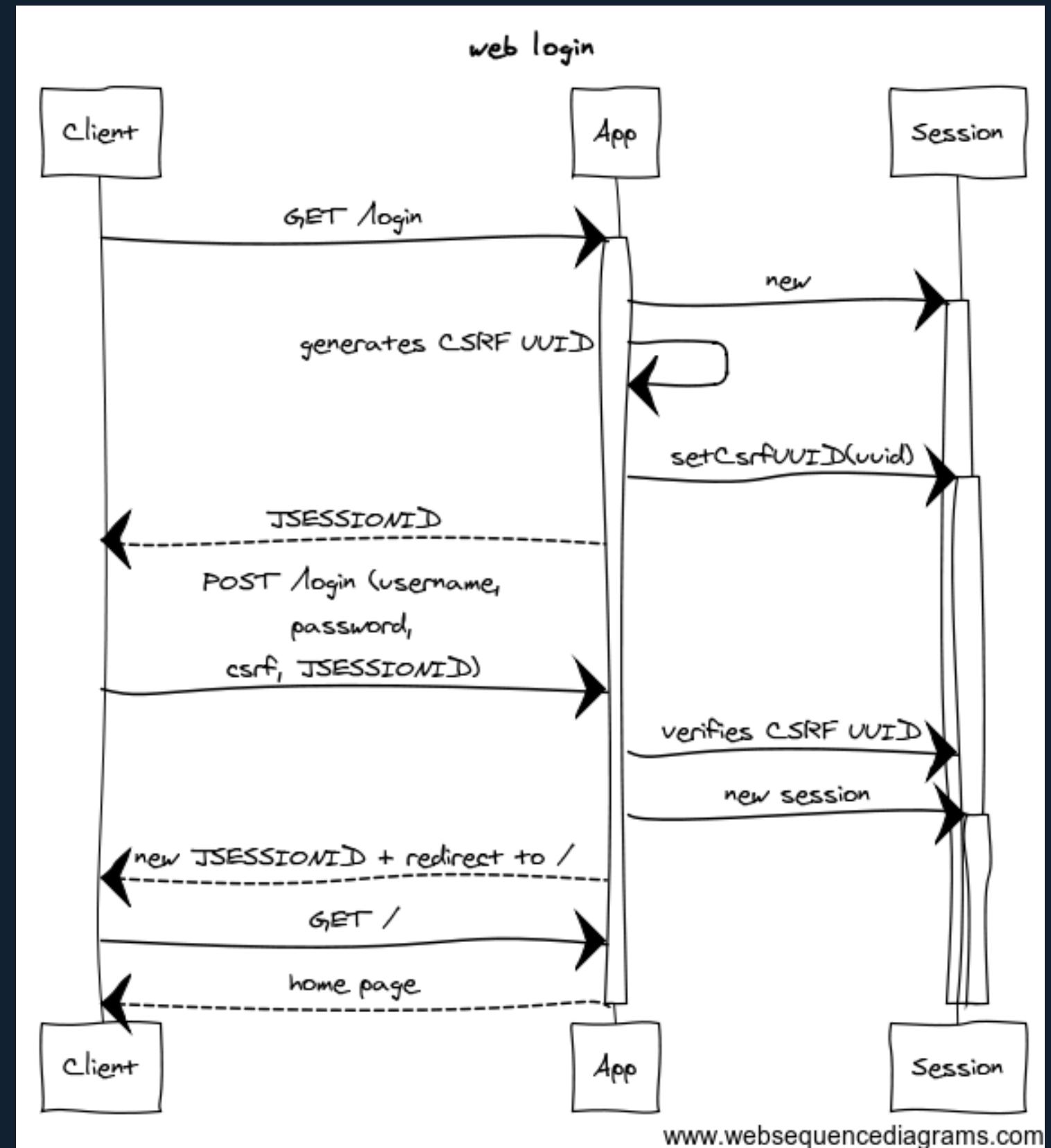
“Cross-Site Request Forgery (CSRF) is a type of attack that occurs when a malicious Web site, email, blog, instant message, or program causes a user's Web browser to perform an unwanted action on a trusted site for which the user is currently authenticated.”

OWASP.ORG

SESSION FIXATION

“Session Fixation is an attack that permits an attacker to hijack a valid user session. The attack explores a limitation in the way the web application manages the session ID, more specifically the vulnerable web application.”

OWASP.ORG



www.websequencediagrams.com

GETTING WEB APP LOGIN FORM

```
$ curl localhost:9090/login -v
```

RESPONSE

```
> Set-Cookie: JSESSIONID=22EF4454BF1268649B200A70A60F13B3; Path=/; HttpOnly

<form name="form" action="/login" method="POST">
  <fieldset>
    <input type="text" name="username" value="" placeholder="Username" />
    <input type="password" name="password" placeholder="Password" />
  </fieldset>
  <input type="submit" id="login" value="Login" class="btn btn-primary" />
  <input type="hidden" name="_csrf" value="2773bb75-a090-4a6a-8ad3-966c94fb7092" />
</form>
```

LOGIN

```
$ curl -v -X POST localhost:9090/login \
-d 'username=user&password=user&_csrf=69035629-af6d-4a49-8396-c3034704ec73' \
-H 'Cookie: JSESSIONID=22EF4454BF1268649B200A70A60F13B3'
```

RESPONSE

```
> Set-Cookie: JSESSIONID=C9778DFFBD1A6CD5DDF3BC77B86CA87B; Path=/; HttpOnly
> Location: http://localhost:9090/
```

CHECKING SECURED PAGE

```
$ curl -v localhost:9090 -H 'Cookie: JSESSIONID=C9778DFFBD1A6CD5DDF3BC77B86CA87B'
```

RESPONSE

```
<h1>Hello Home</h1>
<div>Hello World</div>
```

JMETER EXTRACTING JSESSIONID

Regular Expression Extractor

Name: Regular Expression Extractor

Comments:

Apply to:

Main sample and sub-samples Main sample only Sub-samples only JUnit

Response Field to check

Body Body (unescape) Body as a Document Headers URL

Reference Name: jsessionid

Regular Expression: Set-Cookie: JSESSIONID=(.+?);

Template: \$1\$

Match No. (0 for Random):

Default Value: INVALID_JSESSIONID

JMETER EXTRACTING CSRF TOKEN

Regular Expression Extractor

Name: Regular Expression Extractor

Comments:

Apply to:

Main sample and sub-samples Main sample only Sub-samples only JMeter Variable []

Response Field to check

Body Body (unescape) Body as a Document Headers URL Response Code Response M

Reference Name: csrf

Regular Expression: name="_csrf" value="(.)?"

Template: \$1\$

Match No. (0 for Random):

Default Value: INVALID_CSRF

FINALLY, ALL TESTS¹

tests\$ mvn jmeter:verify

it will execute all performance tests

¹ showtime

JENKINS START

tomcat\$./bin/catalina.sh start

The screenshot shows the Jenkins web interface. At the top, there is a navigation bar with a Jenkins logo, a search bar, and a 'ENABLE AUTO REFRESH' link. Below the navigation bar, on the left, is a sidebar with links: 'New Item', 'People', 'Build History', 'Manage Jenkins', 'Credentials', and 'Jenkins 100K'. In the center, the main content area displays the 'Welcome to Jenkins!' message with a call to action: 'Please [create new jobs](#) to get started.' Below this message, there are two expandable sections: 'Build Queue' (which shows 'No builds in the queue.') and 'Build Executor Status' (which shows '1 Idle' and '2 Idle').

CONFIGURING JENKINS PERFORMANCE PLUGIN

Performance Plugin



This plugin allows you to capture reports from [JMeter](#) and [JUnit](#). Jenkins will generate graphic charts with the trend report of performance and robustness. It includes the feature of setting the final build status as good, unstable or failed, based on the reported error percentage.

1.13

CONFIGURING JENKINS NEW JOB

Item name

Freestyle project
This is the central feature of Jenkins. Jenkins will build your project, combining any SCM with any build system, and this can be even used for something other than software build.

Maven project
Build a maven project. Jenkins takes advantage of your POM files and drastically reduces the configuration.

External Job
This type of job allows you to record the execution of a process run outside Jenkins, even on a remote machine. This is designed so that you can use Jenkins as a dashboard of your existing automation system. See [the documentation for more details](#).

Multi-configuration project
Suitable for projects that need a large number of different configurations, such as testing on multiple environments, platform-specific builds, etc.

CONFIGURING JENKINS REPOSITORY

Source Code Management

None
 CVS
 CVS Projectset
 Git

Repositories	Repository URL
	<input type="text" value="file:///Users/user/projects/jmeter_workshop"/> (?)
Credentials	- none - Add (?)
	Advanced... (?)
	Add Repository Delete Repository (?)
Branches to build	Branch Specifier (blank for 'any')
	<input type="text" value="*/master"/> (?)
	Add Branch Delete Branch

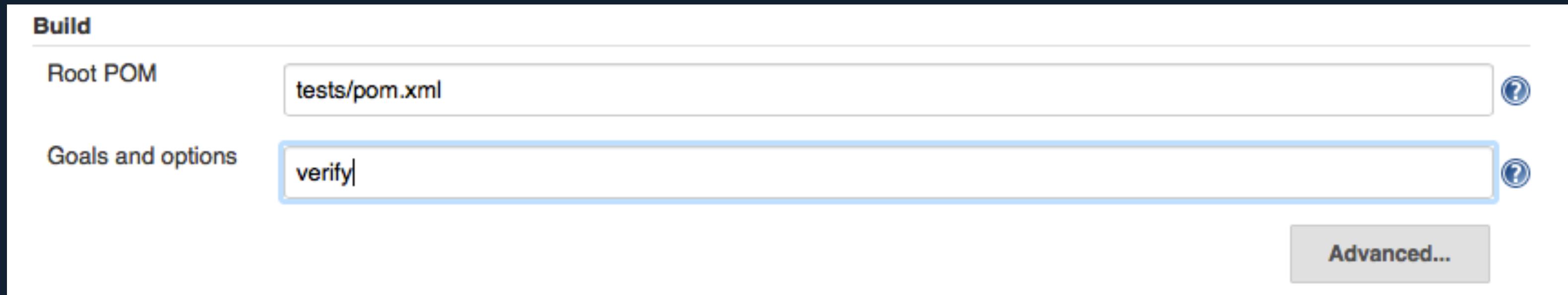
CONFIGURING JENKINS BUILD

Build

Root POM ?

Goals and options ?

Advanced...



CONFIGURING JENKINS PUBLISH REPORT

Post-build Actions

 Publish Performance test result report 

Performance report

 **JMeter** 

Report files 

Delete

Add a new report ▾

Select mode: Relative Threshold Error Threshold

CONFIGURING JENKINS PUBLISH REPORT

The screenshot shows a Jenkins configuration page for a job. On the left, there's a sidebar with the text "Add post-build action" followed by a dropdown arrow. The main area has a title "Performance display" and two checked checkboxes: "Performance Per Test Case Mode" and "Show Throughput Chart". In the bottom right corner of this section, there is a red "Delete" button.

Performance display

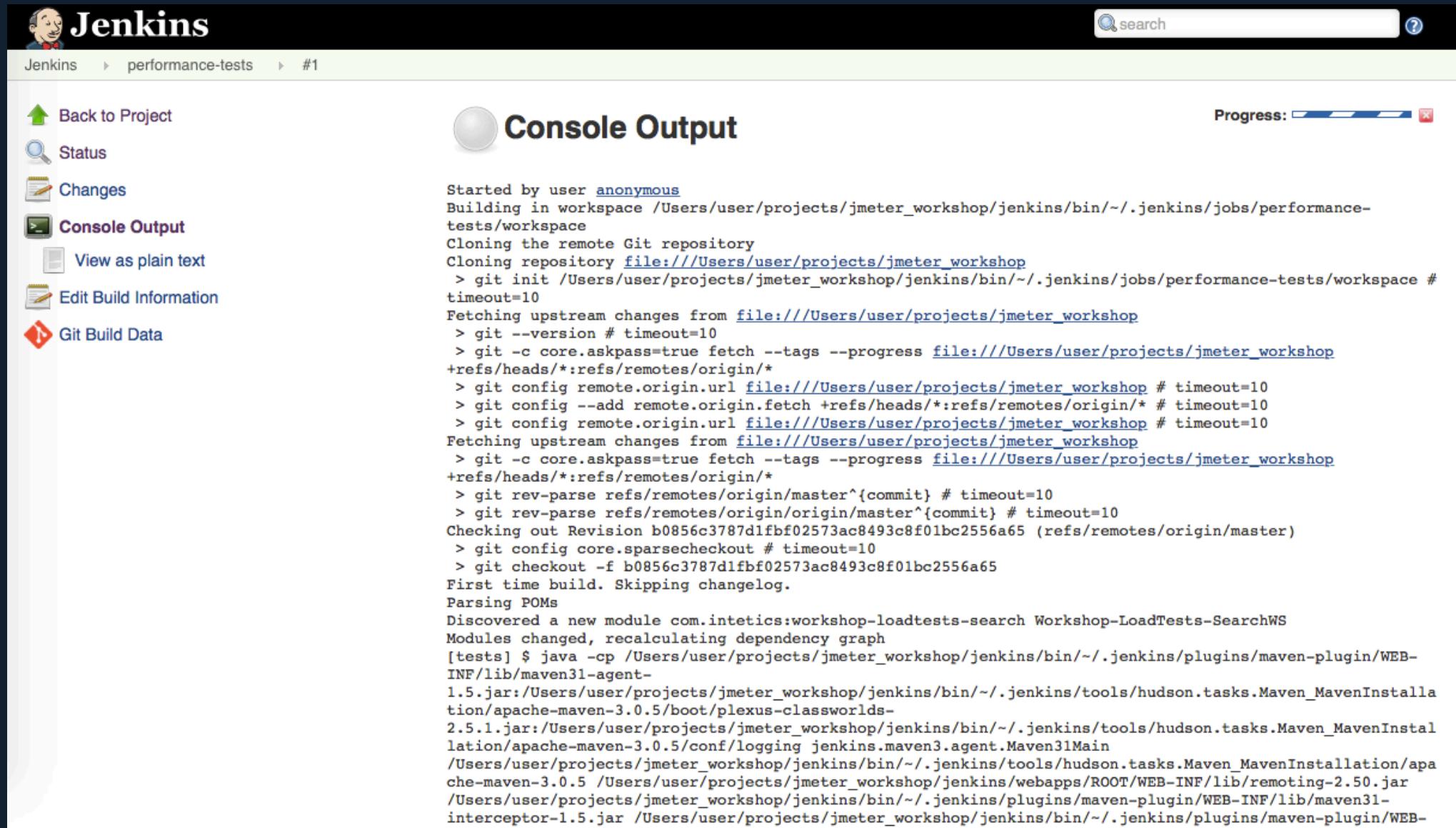
Performance Per Test Case Mode

Show Throughput Chart

Delete

Add post-build action ▾

JENKINS RUN ALL PERFORMANCE TESTS



The screenshot shows the Jenkins interface for a build named "performance-tests #1". The left sidebar includes links for "Back to Project", "Status", "Changes", "Console Output" (which is selected), "View as plain text", "Edit Build Information", and "Git Build Data". The main area is titled "Console Output" and displays the following build log:

```
Started by user anonymous
Building in workspace /Users/user/projects/jmeter_workshop/jenkins/bin/~/jenkins/jobs/performance-
tests/workspace
Cloning the remote Git repository
Cloning repository file:///Users/user/projects/jmeter_workshop
> git init /Users/user/projects/jmeter_workshop/jenkins/bin/~/jenkins/jobs/performance-tests/workspace #
timeout=10
Fetching upstream changes from file:///Users/user/projects/jmeter_workshop
> git --version # timeout=10
> git -c core.askpass=true fetch --tags --progress file:///Users/user/projects/jmeter_workshop
+refs/heads/*:refs/remotes/origin/*
> git config remote.origin.url file:///Users/user/projects/jmeter_workshop # timeout=10
> git config --add remote.origin.fetch +refs/heads/*:refs/remotes/origin/* # timeout=10
> git config remote.origin.url file:///Users/user/projects/jmeter_workshop # timeout=10
Fetching upstream changes from file:///Users/user/projects/jmeter_workshop
> git -c core.askpass=true fetch --tags --progress file:///Users/user/projects/jmeter_workshop
+refs/heads/*:refs/remotes/origin/*
> git rev-parse refs/remotes/origin/master^{commit} # timeout=10
> git rev-parse refs/remotes/origin/origin/master^{commit} # timeout=10
Checking out Revision b0856c3787d1fbf02573ac8493c8f01bc2556a65 (refs/remotes/origin/master)
> git config core.sparsecheckout # timeout=10
> git checkout -f b0856c3787d1fbf02573ac8493c8f01bc2556a65
First time build. Skipping changelog.
Parsing POMs
Discovered a new module com.intetics:workshop-loadtests-search Workshop-LoadTests-SearchWS
Modules changed, recalculating dependency graph
[tests] $ java -cp /Users/user/projects/jmeter_workshop/jenkins/bin/~/jenkins/plugins/maven-plugin/WEB-
INF/lib/maven31-agent-
1.5.jar:/Users/user/projects/jmeter_workshop/jenkins/bin/~/jenkins/tools/hudson.tasks.Maven_MavenInstalla-
tion/apache-maven-3.0.5/boot/plexus-classworlds-
2.5.1.jar:/Users/user/projects/jmeter_workshop/jenkins/bin/~/jenkins/tools/hudson.tasks.Maven_MavenInstal-
lation/apache-maven-3.0.5/conf/logging jenkins.maven3.agent.Maven31Main
/Users/user/projects/jmeter_workshop/jenkins/bin/~/jenkins/tools/hudson.tasks.Maven_MavenInstallation/apa-
che-maven-3.0.5 /Users/user/projects/jmeter_workshop/jenkins/webapps/ROOT/WEB-INF/lib/remoting-2.50.jar
/Users/user/projects/jmeter_workshop/jenkins/bin/~/jenkins/plugins/maven-plugin/WEB-INF/lib/maven31-
interceptor-1.5.jar /Users/user/projects/jmeter_workshop/jenkins/bin/~/jenkins/plugins/maven-plugin/WEB-
```

JENKINS REPORTING

Jenkins

performance-tests

ENABLE AUTO REFRESH

search

Back to Dashboard

Status

Changes

Workspace

Build Now

Delete Maven project

Configure

Modules

Performance Trend

Build History

trend

#3 Mar 30, 2015 9:56 PM

#2 Mar 30, 2015 9:55 PM

#1 Mar 30, 2015 9:53 PM

RSS for all RSS for failures

Maven project performance-tests

add description

Disable Project

Performance Trend

Responding time

ms

2000

1500

1000

500

#1 #2 #3

90% line — average — median

Percentage of errors

%

100

80

60

40

20

0

#1 #2 #3

errors

Workspace

Last Successful Artifacts

dashBoard_.xml

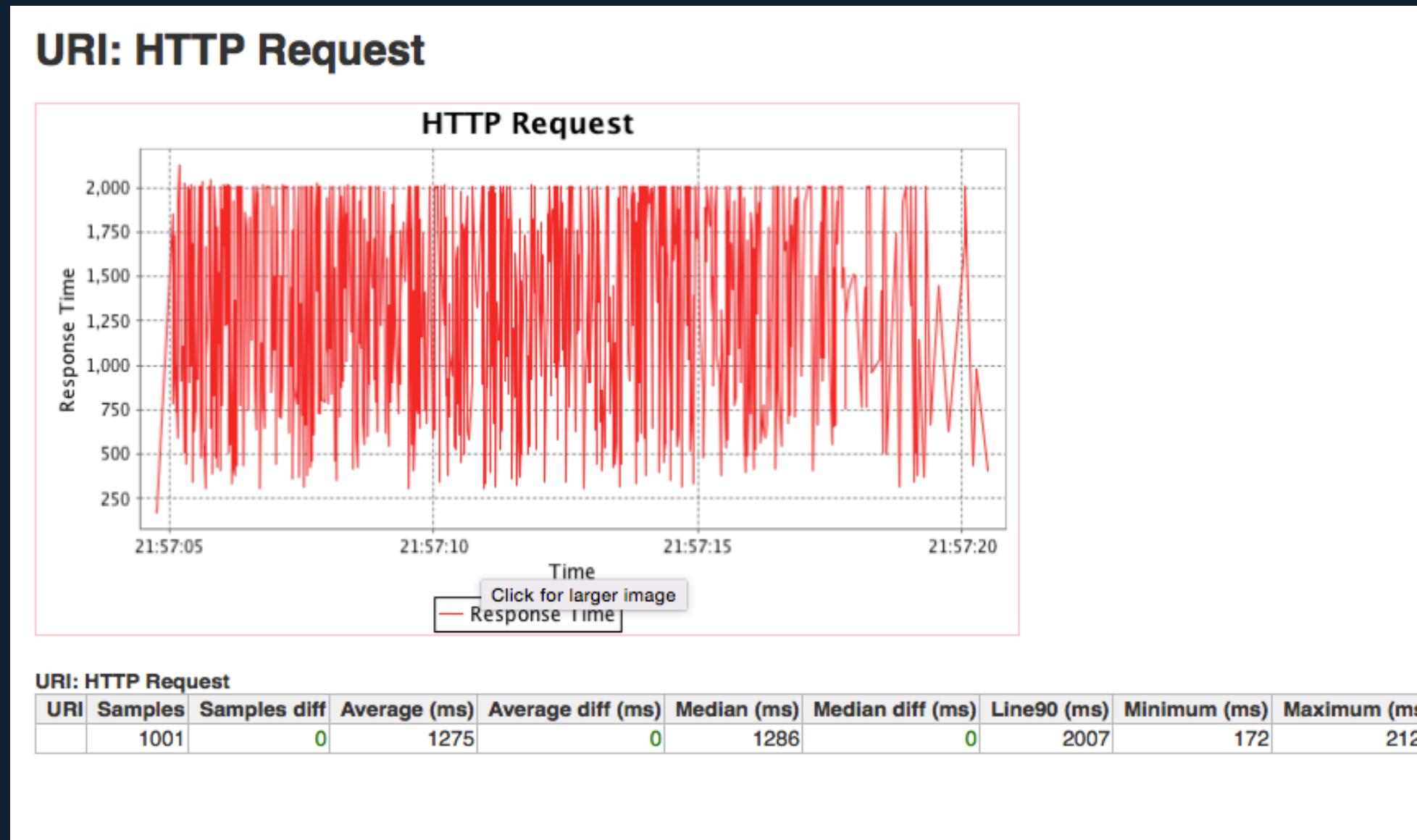
457 B view

Recent Changes

Permalinks

- Last build (#3), 1 min 13 sec ago
- Last stable build (#3), 1 min 13 sec ago
- Last successful build (#3), 1 min 13 sec ago

JENKINS REPORTING



LINKS

- » Automating JMeter tests with Maven and Jenkins
- » JMeter Maven Plugin
- » JMeter Maven Plugin online documentation
- » About Constant Throughput Timer on Stackoverflow
- » How to run a test plan for a certain amount of time (very strange approach)
- » JMeter functions (\$__P(...))

LINKS

- » Passing global parameters to a jmx file
- » How to use a CSV file with JMeter
- » Difference between sharing modes “All threads” and “Current thread group” in Jmeter
- » How to use BeanShell
- » JMeter how to NOT fail 500 Internal Server Errors
- » Howto debug JMeter scripts



QUESTIONS?