SELENUM Programming Cookbook

Hot Recipes for Selenium Development



PETR ARSENTEV





Selenium Programming Cookbook

Contents

1	Sele	nium Installation Example	1
	1.1	Introduction	1
	1.2	Installation	1
	1.3	Conclusion	6
2	Sele	nium IDE Tutorial	7
	2.1	Introduction	7
	2.2	Installation	7
	2.3	Testing	11
	2.4	Conclusion	16
3	Sele	nium Automation Testing Tutorial	17
	3.1	Introduction	17
	3.2	Create a tests plan	17
	3.3	Install the Selenium IDE	18
	3.4	Record the user activities in Selenium IDE	21
	3.5	Refactoring exported code	26
	3.6	Download the Maven project	28
4	Sele	nium Interview Questions and Answers	29
	4.1	Introduction	29
	4.2	Interview Questions and Answers	29
	4.3	Conclusion	38
5	Sele	nium Standalone Server Example	39
	5.1	Introduction	39
	5.2	High level architecture	39
	5.3	Configuration	40
	5.4	Run tests	42
	5.5	Download the Code Project	46

6	Sele	nium JUnit Example	47
	6.1	Introduction	47
	6.2	Record test cases	47
	6.3	Integrate to JUnit	51
	6.4	Conclusion	56
	6.5	Download the source code	57
7	Sele	nium Grid Example	58
7	Sele 7.1	nium Grid Example	58 58
7	Sele 7.1 7.2	nium Grid Example Introduction	58 58 59
7	Sele 7.1 7.2 7.3	nium Grid Example Introduction	58 58 59 60
7	Seler 7.1 7.2 7.3 7.4	nium Grid Example Introduction	58 58 59 60 65

Copyright (c) Exelixis Media P.C., 2016

All rights reserved. Without limiting the rights under copyright reserved above, no part of this publication may be reproduced, stored or introduced into a retrieval system, or transmitted, in any form or by any means (electronic, mechanical, photocopying, recording or otherwise), without the prior written permission of the copyright owner.

Preface

Selenium is a portable software testing framework for web applications. Selenium provides a record/playback tool for authoring tests without learning a test scripting language (Selenium IDE). It also provides a test domain-specific language (Selenese) to write tests in a number of popular programming languages, including Java, C#, Groovy, Perl, PHP, Python and Ruby.

The tests can then be run against most modern web browsers. Selenium deploys on Windows, Linux, and Macintosh platforms. It is open-source software, released under the Apache 2.0 license, and can be downloaded and used without charge. (Source: https://bit.ly/2bNZYaa)

In this ebook, we provide a compilation of Selenium programming examples that will help you kick-start your own projects. We cover a wide range of topics, from Installation and JUnit integration, to Interview Questions and Standalone Server functionality. With our straightforward tutorials, you will be able to get your own projects up and running in minimum time.

About the Author

Petr Arsentev has over 8 years of experience in java development. He participated in the development a few startup projects, which run successfully. He finished Moscow Power Engineering Institute (National Research University) at 2009.

After he started to work in a local company as java developer and still keeps improving the knowledge about software developments. He focused on JVM languages like Java, Scala and related technologies and frameworks. He has developed the few courses about Java in Russian. He teaches students Java language too. His personal website is https://parsentev.ru/.

Chapter 1

Selenium Installation Example

1.1 Introduction

In this tutorial, we are going to show how you can install the Selenium IDE, Selenium Server and Selenium WebDriver.

Selenium is the tool for automation testing web apps. Selenium consists from IDE, WebDrivers and Server.

The general scenery of testing in Selenium is to record the user activities by Selenium IDE and after that to run this tests cases automatically. It uses two different approaches to execute the tests. One of them uses the native browser API, another uses the injecting JavaScript codes to browsers. The main benefit about Selenium IDE is the ability to execute tests on varied browsers. You can find the list of supported Selenium browsers below:

- Firefox
- IE
- Safari
- Opera
- Google Chrome

Another great thing about Selenium is that you can export the tests cases to your favorite programming languages: Java, Ruby, Python, C# and so on.

1.2 Installation

The installation process is very simple. Firstly, You should go to official web site seleniumhq.org.



Figure 1.1: Selenium main page

There you can find the download tab, if you click on the tab you will move to download page, as show below. First, you will need Selenium IDE. You need to scroll the page and find the Selenium IDE paragraph. Selenium IDE is the Firefox plugins, so you need to click on this link and follow the next steps. The main important thing is that you can use only Firefox on this process. Selenium IDE addon is not supported by another browsers.

Selenium IDE
Selenium IDE is a Firefox plugin which records and plays back user interactions with the browser. Use this to either create simple scripts or assist in exploratory testing. It can also export Remote Control or WebDriver scripts, though they tend to be somewhat brittle and should be overhauled into some sort of Page Object-y structure for any kind of resiliency.
Download latest released version from addons.mozilla.org or view the <u>Release Notes</u> and then <u>install some plugins</u> .
Selenium IDE Plugins
Selenium IDE can be extended through its own plugin system. Here are a number of plugins that have been created using it. For more information on how to create your
own plugin or have it listed, see the plugin tutorial page.

This link redirects you to Firefox addons page. The page offers you to add the Selenium IDE addons. Then you need to click this button.

	Register or Log in Other A	pplications "
	ADD-ONS	rch for add
» Extens	ons » Selenium IDE	
	Selenium IDF and	*
60	by Jason Huggins, Adam Goucher, Shinya Kasatani, Dave Hunt, Samit Badle	10
36	Selenium IDE is an integrated development environment for Selenium tests. It is	12
	implemented as a Firefox extension, and allows you to record, edit, and debug tests.	8
	 implemented as a Firefox extension, and allows you to record, edit, and debug tests. Add to Firefox 	4
	implemented as a Firefox extension, and allows you to record, edit, and debug tests. + Add to Firefox This add-on has been preliminarily reviewed by Mozilla. Learn more	

Figure 1.3: Plugins page

After that, Firefox browser offers you to install this addon



Figure 1.4: Install Addons

and restart the browser.



Figure 1.5: Restart

Then you restart to browser, you can find the Selenium IDE button on the right-top corner.

Selenium IDE 2.9.1		
<u>File Edit Actions Options</u>	lelp	
Base URL http://examples.javao	odegeeks.com/	
Fast Slow	9 0	() -
Test Case	Table Source	
Untitled	Command Target	t Value
Runs: Failures:	Command Target Value	▼ Select Find

Figure 1.6: Selenium IDE

Now you can record your first test.

Selenium has the special server, which offers to scale your tests. Let's back to Selenium download page and find there the Selenium Server.

🕞 🎯 🛛 www.seleniumhq.org/	(download/ 🖾 C Search ☆ 🖻 🛡 🖡 🎓 🗭 🕷 🛩 🖛
Seleniu Browser Autom	ation Projects Download Documentation Support About
Solonium Downloads	Downloads
Latest Releases	Below is where you can find the latest releases of all the Selenium components. You can also find a list of previous releases, source code, and additional information for
Previous Releases	<u>Maven users</u> (Maven is a popular Java build tool).
Source Code	Selenium Standalone Server
Maven Information	The Selenium Server is needed in order to run either Selenium RC style scripts or Remote Selenium WebDriver ones. The 2.x server is a drop-in replacement for the old Selenium RC server and is designed to be backwards compatible with your existing infrastructure.
with PayPal	Download version <u>2.48.2</u> To use the Selenium Server in a Grid configuration <u>see the wiki page</u> .
through sponsorship You can <u>sponsor the</u>	reatest features of the

Figure 1.7: Selenium Server

Selenium Server is the JAR file and it does not require the installing. So after you download this JAR, you can run it directly.

How we said before Selenium offers you to use varied browsers for execution the tests. For this reason, we should use the WebDriver. You should back to the download page again.

Selenium Client & WebDriver Language Bindings

In order to create scripts that interact with the Selenium Server (Selenium RC, Selenium Remote WebDriver) or create local Selenium WebDriver scripts, you need to make use of language-specific client drivers. These languages include both 1.x and 2.x style clients.

While language bindings for <u>other languages exist</u>, these are the core ones that are supported by the main project hosted on google code.

ru				
Language	Client Ver	sion Release Dat	e	
Java	2.48.2	2015-10-09	Download	Change log Javadoc
C#	2.48.0	2015-10-07	Download	Change log API docs
Ruby	2.48.0	2015-10-07	Download	Change log API docs
Python	2.48.0	2015-10-07	Download	Change log API docs va Conto Cooks
Javascript (No	de) 2.47.0	2015-09-15	Download	Change log API docs JAVA DEVELOPERS RESOURCE CENTER

Figure 1.8: WebDriver

For example, if you use Java, you need to download the WebDriver for Java programming language and so on. This library has the necessary API for execution tests on varied browsers.

1.3 Conclusion

In this tutorial, we've shown how to install the Selenium tools. As you saw this process is kindly simple. If you want to improve your knowledge about Selenium, you can visit the official web site.

Chapter 2

Selenium IDE Tutorial

2.1 Introduction

In this tutorial, we are going to show how you can test your web app by Selenium IDE. Selenium IDE is the Firefox plugin, which can record the user browser action and run it automatically further. We are going to install the Selenium IDE, explain most useful controlls and panels, record the simple use cases and run it automatically by Selenium IDE, add the schedule for tests, export this test case to programming languages.

Selenium IDE has special commands. Selenium commands emulate user activities. You can emulate any user activities: insert the text, submit the form, navigate in apps, click on link, click on checkbox, select the options in combobox. The main benefic about Selenium IDE is that you don't need to have any experience in programming languages. All you need is install the Selenium plugin, record the use activities and run the tests. It is so user-friendly. Selenium IDE uses the native browser API for testing, so you can use all specters commands, which Firefox browser supports. Let's start to install the Selenium IDE.

2.2 Installation

First of all, we go to the download page and select the last version of Selenium ide there - https://docs.seleniumhq. org/download/.Download Selenium IDE Download Selenium IDE

After we downloaded this plugin, Firefox browser offers to install this addons. You need to allow this process. Selenium IDE is installed and we can see the Selenium IDE button on the right-top corner in Firefox.



Figure 2.1: Selenium IDE Button

Then you need to open the Selenium IDE, you can click on this button or select the Selenium IDE in development tools in Firefox, as show below.

☆ 自 ♥	7 🖡 🏠 🗑 📓	* - E
Cut	WEB DEVELOPER	
-	Toggle Tools	
	Inspector	Ctrl+Shift+C
	Web Console	Ctrl+Shift+K
√indow	Debugger	Ctrl+Shift+S
	Style Editor	Shift+F7
	Performance	Shift+F5
int	Network	Ctrl+Shift+Q
	Developer Toolbar	Shift+F2
P	WebIDE	Shift+F8
nd	Browser Console	Ctrl+Shift+J
	Responsive Design View	Ctrl+Shift+M
6	Eyedropper	
loper	Scratchpad	Shift+F4
	Page Source	Ctrl+U
	Get More Tools	
ו in to S	Selenium IDE	Alt+Ctrl+S
tomize	Work Offline AVA CO	de Geeks Ers resource center

Figure 2.2: Open Selenium IDE plugins

Selenium IDE looks as show below.

<u>File E</u> dit <u>A</u> c	tions <u>O</u> ptions Help			
Base URL htt	p://localhost:8080/TrackStu	dio		
Fast Slow	s de de 11 🦻 🖉	0		0 - 0
Test Case		Table Source		
Untitled	Run single	test	Target	Value Case
	Run multiple tests	Ĵ		
			Test	t case panel
		Command		•
		Target	✓ S	elect
Runs: Failures:	0	Value		
Log Refere	ence Expert UI-Element	Rollup	••• ▼ 1	Info + Clear
		Informational p		ava Code Geek

Figure 2.3: Selenium main controls

Let's describe the main useful controll:

- Record user activities it is the toggle button, when it turns on, it records the user activities in the main browser window.
- Run single test it run the single selected test.
- Run multiple tests it run all tests, which it was opened in Selenium IDE.
- **Base url** this input box defines the webapp url, when we go to this url in main browser window, Selenium IDE startes to record all activities on this window.
- Test case panel this panel contains all user activities, which Selenium IDE records.

Next, we are ready to record our first test case.

2.3 Testing

Let's imagine, that we need to test the searching function in this site - https://examples.javacodegeeks.com. First, we should turn the record button on, put the base URL and go to main browser window. We go to this site, find the search input, type the necessary text, submit the form. After that, we get the searching result and now we need to verify this result. We need to select the necessary element and open the right mouse menu, as show below.



Figure 2.4: User controlls

This menu contains addition Selenium IDE options: list of available commands, assert's commands. We need to select the assert command, if we want to check the result. Now our test case is ready and we can go to Selenium IDE to see how it looks in Selenium code.



Figure 2.5: Manual adding command

How we can see the test case panel is fulled. This table contains the emulations user activities. Sometimes, we need to add the command manually.

You need to click on empty row in the table and fill the input controls below. Let's run our test and see the result. We click on run button and wait until test finishs.

<u>File Edit Actions Options He</u>	p			
Base URL http://examples.javacoo	legeeks.com/			
East Slow	₽ @			<u>(</u>) - (
Test Case	Table Source			
Test JCG *	Command	Target	Value	
	open	/		
	type clickAndWait	id=s css=button.searc	selenium h-button	
Indication of test's result	assertText	link=Seleniur So	elenium elenium Au ommand	stomation Tes
	Command	Ele	ment location	•
	Target		Select	Find
Runs:	1 Value	63		
Failures:	0		N. In	
		Cor	nmand params	
Log Reference Expert UI-E	ement Rollup			Info+ Clear
[info] Executing: open /	I			*
[info] Executing: type id=	s selenium			
[info] Executing: clickAndW	ait css=button.search-	button		
[info] Executing: assertTex Tutorial	t link=Selenium Autom	ation Testing Tutoria	FI Selenium Automa	tion Testing
[info] Test case passed	Test's log	0		

Figure 2.6: Run test

We need to save out test case, that we can use it further.



Figure 2.7: Save Test suite

Sometimes, we need to run the test periodically. Selenium IDE supports this functionality too. Go to Main menu -Options -Schedule tests to run periodically, as shown below. Schedule controll Schedule controll

You can make the configuration in open windows. For example, we chose to run our test case every hour. Then you need to turn test schedule on.

Suite C:\Users\parser	ntev\Documents\Te	est JCG.html	Choose
Schedule	6	Test case	
Day	V Hour	V Hour	Minute
Sunday	00	12	00
🔽 Monday	☑ 01	V 13	05
Tuesday	02	V 14	10
Wednesday	03	15	V 15
🔽 Thursday	☑ 04	16	20
V Friday	☑ 05	17	25
📝 Saturday	V 06	V 18	30
	☑ 07	19	35
	08	20	V 40
	09	21	V 45
	1 0	22	50
	▼ 11 Exec	ution period	55
Advanced Every hour			✓ Change
Run Now			23.12.2015, 11:50:00

Figure 2.8: Schedule config

X Selenium IDE 2.9.1 File Edit Actions Options Help Export the test case to ackStudio New Test Case Ctrl+N programm language Ctrl+O 0 Open... Save Test Case Ctrl+S I Source Save Test Case As... TT Value Export Test Case As... C# / NUnit / WebDriver . Recent Test Cases Þ C# / NUnit / Remote Control Java (JUnit 4 / WebDriver Add Test Case ... Ctrl+D Java / TestNG / WebDriver Properties... Java / JUnit 4 / WebDriver Backed New Test Suite Java / JUnit 4 / Remote Control Open Test Suite... Java / JUnit 3 / Remote Control Save Test Suite Java / TestNG / Remote Control Save Test Suite As... Python 2 / unittest / WebDriver Find Select Export Test Suite As... ٠ Python 2 / unittest / Remote Control **Recent Test Suites** ۶ Ruby / RSpec / WebDriver Close (X) Ctrl+W Ruby / Test:: Unit / WebDriver Ruby / RSpec / Remote Control Reference Expert UI-E Infor Clear Log Ruby / Test:: Unit / Remote Control Java Code Geek

Selenium IDE can export your case tests to your favorite programming languages.

Figure 2.9: Export test cases

2.4 Conclusion

In this article, we show you how you can use Selenium IDE to test your web apps. Selenium IDE is the great tool for building automations tests easily.

• You can find more information about this tool on the official website link.

Chapter 3

Selenium Automation Testing Tutorial

3.1 Introduction

In this example, we shall show you how to write the automation tests by Selenium. Selenium is collections of tools to test the web applications. We are going to cover user's cases for TrackStudio system.

Automation tests has few benefits:

- It is easy for supporting.
- It is faster than manual tests.
- It has possibility to repeat tests.
- It has the lower cost with compare to manual tests.
- It has ability to use in Continue Integration.
- It has ability to get exhausted reports.

Selenium consists from three main parts:

- Selenium IDE add-ons for Firefox.
- Selenium WebDriver library, which use native browser API for testing.
- Selenium Server server for execute tests in different environments.

3.2 Create a tests plan

Before we start to write the tests code, we should write the tests plan. The plan should describe what we should do and what we will expect after those actions. Let's tell few words about TrackStudio system. TrackStudio is issues/tasks tracker in the simple words.

It supports: workflow processing, documents processing, access rules control and so on. It fits great for our demonstration purpose, because it is the web app with rich user interfaces and it has many functional features.

Let's imagine that we want to test all cases for creating the library (books store) configuration in TrackStudio.

Table 3.1: Tests cases

Short explanation	Detail explanation	Expected result
Log in as administrator	1 Go to index page 2 Fill the login	Redirect to main page
	form. 3. Submit the form.	Rediffect to main page.
Create the new rule - Library manager.	4. Go to manage user tab 5. Open the	The rule should appear in rules list
	rules tab. 6. Click on the create button	
	7. Fill the form. 8. Submit the form.	
Create the new rule - Reader.	1. Go to manage user tab 2. Open the	The rule should appear in rules list.
	rules tab. 3. Click on the create button	
	4. Fill the form. 5. Submit the form.	
Create the new User with Manager	1. Go to manage user tab 2. Open the	The new user should appear in users
rule - Bread.	users tab. 3. Click on the create button	list.
	4. Fill the form. 5. Submit the form.	
Create the new User with Reader rule	1. Go to manage user tab 2. Open the	The new user should appear in users
- Smith.	users tab. 3. Click on the create button	list.
	4. Fill the form. 5. Submit the form.	
Create the new workflow - Book	1. Go to manage task tab. 2. Open the	The new workflow should appear in
	workflow tab. 3. Click on the create	workflows list.
	button. 4. Fill the form. 5. Submit the	
	form.	
Create the two workflow states - in,	1. Go to manage task tab. 2. Open the	The new status should appear in
out	workflow tab. 3. Open the status tab.	workflows status list.
	4. Click on the create button. 5. Fill	
	the form. 6. Submit the form.	
Create the two workflow transaction -	1. Go to manage task tab. 2. Open the	The new transactions should appear in
out to in, in to out	workflow tab. 3. Open the transaction	the workflow.
	tab. 4. Selected the necessary status	
	and click on the create button.	
Create the new category for Book	1. Go to manage task tab. 2. Open the	The new category should appear in the
workflow.	category tab. 3. Click on the create	categories list.
	button. 4. Fill the form. 5. Submit the	
	form.	
Create the new project for library.	1. Go to manage task tab. 2. Click on	The new project should appear in the
	the new project button. 3. Fill the	task list.
	form. 4. Submit the form.	
Add the access for rule Manager and	1. Go to manage task tab. 2. Open the	The new access rule should appear in
Reader to the library project	access rule tab. 3. Selected the rules.	the access rules list.
	4. Submit the form.	
Login as the Bread user and create the	1. Log out. 2. Log in as Bread. 3. Go	The new book should appear in the
new book.	to the library project. 4. Click the new	book list.
	book button. 5. Fill the form. 6.	
	Submit the form.	
Login as the Smith user and take the	1. Log out. 2. Log in as Smith. 3. Go	The book should change the status to
new dook.	to the library project. 4. Click the new	out.
	book link. 5. Click the out operation	
	button. 6. Fill form and submit it.	

Table 3.1: (continued)

Now we can go to write your tests.

3.3 Install the Selenium IDE

First of all we should install the Selenium IDE. You should open the Firefox.Then you should go to official Selenium web site link.



Figure 3.1: Download Selenium IDE

After you restart the firefox browser you can see the button in the right top corner. It is the Selenium IDE.

🖉 Downloads	× +						
www.seleniumhq.org/d	ownload 🗊 🔻 C 🔍 Search 😒 🖨 🖾 🕊 🖛 🗩 🗲 🚍						
Seleniu	mHQ edit this page search selenium: Go						
Browser Automa	tion Projects Download Documentation Support About						
Selenium Downloads	Downloads						
Latest Pelasses	Below is where you can find the latest releases of all the Selenium components.						
Latest Releases	Latest Releases You can also find a list of <u>previous releases</u> , <u>source code</u> , and additional information for Mayen users (Mayen is a popular Java build tool).						
Previous Releases							
Source Code	Selenium Standalone Server						
Maven Information	Maven InformationThe Selenium Server is needed in order to run either Selenium RC style scripts or Remote Selenium WebDriver ones. The 2.x server is a drop-in replacement for the						
Donate to Selenium	Donate to Selenium Selenium RC server and is designed to be backwards compatible with your existing infrastructure. existing infrastructure.						
with PayPal	Download version 2.48.2						
Donate	To use the Selenium Server in a Grid configuration see the wiki page.						
	The Internet Explorer Driver Server						
through sponsorship	This is required if you want to make use of the latest and greatest features of the						

Figure 3.2: Selenium IDE Button

Then we should click on this button and Selenium IDE opens.

📵 Selenium IDE 2.9.1					-	
File Edit Actions Optio	ns Help					
Base URL http://localhost	Base URL http://localhost:8080/TrackStudio -					
Fast Slow	0 🕫 🔝					🕓 - 💽
Test Case	Table Source					
Untitled	Command	Target		١	Value	
Ī						
: •						
	Command					
	Target			-	Select	Find
Runs: 0	Value					
		▼	······ ▼I			
Log Reference Expert	UI-Element	Rollup				Info ⁺ Clear
			T			
			<u>y</u>	QG)	Java L	idde Geeks
Selenium Expert by S	Samit Badle. Cli	ick the inspect	button to r	un inspe	ctions.	LOPERS RESOURCE CENTER

Figure 3.3: Selenium IDE

Now we are ready to record out test cases.

3.4 Record the user activities in Selenium IDE

Firstly, we should run the TrackStudio Server. You should open the FireFox and open the URL https://localhost:8080/TrackStudio/It will look similar:

Powered by TrackStudio × +	e Crea							×
♦ Iocalhost:8080/TrackStudio/Login/ ▼	C Q Search	☆自	◙	÷	⋒	ø	»	≡
	TrackStudio	D.						
Используйте администрир TrackStudio.	е аккаунт root с паролем ^с ровать и настраивать ваш	'root' чтобы экземпляр						
Учетная запись	root							
Пароль	••••							
	Войти							
	<u>Забыли пароль?</u>							
TrackStudio 5.5.0.20150922 Evaluation l	License (null-Days Trial) Purch	ase today! ©	2002-:	2015, Java 2	TrackS	itudio, L Code ELOPERS R	.td. GQ Issource	CENTER

Figure 3.4: TrackStudio Login Page

Now we need to record our user's cases. Open the Selenium IDE. You have to check that Base URL is the same as TrackStudio Login page and the button record turns on.

_	🅤 📴 Powered by TrackStudio	× +			
(Contemporaries Contem	Studio/LoginAction.do	?method=loginPage	▼ C Search	☆ 自 ♥
				TrackStudio.	
			Используйта администри TrackStudio	е аккаунт root с паролем 'root' что ровать и настраивать ваш экземпл	обы ляр
			Учетная запись	root	
			Пароль	••••	
				🔲 Запомнить меня	
				Войти	
				Забыли пароль?	_
1	🕘 Selenium IDE 2.9.1				
I	<u>File Edit Actions Options</u>	s Help			2002-2015 TrackStudie Lt
I	Base URL http://localhost:8	080/TrackStudio			▼ 2002-2013, HackStudio, Et
I	Fast Slow	II 🕹 🥝		🕓 - (
4	Test Case	Table Source			
ļ	Untitled	Command	Target	Value	
				G	LAVA CODE GEEKS

Figure 3.5: Start to record the user activities

Then you start to make the actions in browser as general users: click on link, fill the form and so on. Selenium IDE will records all this activities. After you finished your actions, you can see that Selenium IDE filled the table.

🌒 1 (untitled suite) - Seleniur	m IDE 2.9.1			
<u>File Edit Actions Options</u>	Help			
Base URL http://localhost:80	080/TrackStudio			▼.
Fast Slow	II 🕹 🥝			🕓 - 🥥
Test Case	able Source			
1	Command	Target	Value	
4	open type clickAndWait assertLocation	http://localhost:8080/TrackStu id=password css=input.iconized http://localhost:8080/TrackStu	dio/Login root dio/staticfr	
Test Case Property	Connect Con		×	
File			· ·	Select Find
Runs: Failur Title LoginAsAdm	iin			
Log				Info+ Clear
		ОК	Cancel	Java Code Geeks

Figure 3.6: Login As Admin

We do other tests similar actions. After that, we should get the list with our tests cases.

Untitled (untitled suite) - Selenium IDE	2.9.1 *		a de la deserve
<u>File Edit Actions Options Help</u>			
Base URL http://localhost:8080/TrackStu	dio		
East Slow)		
Test Case	Table Source		
TrackStudio, Create the library confi	Command	Target	Value
	type	id=password_confirm	root
	clickAndWait	name=saveButton	
	clickAndWait	css=a.internal	
	assertTable	css=table.general.2.1	Smith
	clickAndWait	id=ts-menu-object-16	
	clickAndWait	id=ts-menu-object-18	
	clickAndWait	link=Create Workflow	
	type	id=name	Book
	clickAndWait	name=SETCATEGORY	
	clickAndWait	css=a.ul	
	assertTable	css=table.general.1.1	Book
	_1:_1.A1\A/_:a	link Deels	
	Command		
	Target		
Runs: 0	Value		
Failures: 0		I 🗶	······································
Log Reference Expert UI-Element	Rollup		~
			Java Code Geeks

Figure 3.7: TrackStudio Selenium IDE Test Cases

When you need to assert the text in the web page, you should select the necessary element and open the popup menu, like this:



Figure 3.8: Assert Link

Selenium IDE inserts there assertions in the tests code. So now we are ready to export the code to our favorite programming language from Selenium IDE. You should go to File->Export->Select language

🕘 ເ	Untitled (untitled suite) - Selen	ium IDE 2.9.1 *					
<u>F</u> ile	<u>E</u> dit <u>A</u> ctions <u>O</u> ptions He	elp					
	New Test Case Ctrl+N ackStudio						
	Open Ctrl+O	Ð (O					
	Save Test Case Ctrl+S	Table Source					
	Save Test Case <u>A</u> s						
	Export Test Case As 🕨	C# / NUnit / WebDriver	Value				
	Recent Test Cases	C# / NUnit / Remote Control	root				
	Add Test Case Ctrl+D	Java / JUnit 4 / WebDriver Jutton					
	Properties	Java / TestNG / WebDriver al	Dooleo				
		Java / JUnit 4 / WebDriver Backed neral.2.1	Smith				
	New Test Suite	Java / JUnit 4 / Remote Control object-16	ESOURCE CENTER				
	Open Test Suite	chiect-18					

Figure 3.9: Export

3.5 Refactoring exported code

Now we have the source code on your tests cases. Then we should create the maven project with selenium dependency and put there your test code.

pom.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="https://maven.apache.org/POM/4.0.0"</pre>
         xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
         xsi:schemaLocation="https://maven.apache.org/POM/4.0.0 https://maven.apache.org/ ↔
            xsd/maven-4.0.0.xsd">
    <modelVersion>4.0.0</modelVersion>
    <groupId>ru</groupId>
    <artifactId>parsentev</artifactId>
    <version>1.0-SNAPSHOT</version>
    <dependencies>
        <dependency>
            <groupId>junit</groupId>
            <artifactId>junit</artifactId>
            <version>4.12</version>
        </dependency>
        <dependency>
            <proupId>org.seleniumhq.selenium</proupId>
            <artifactId>selenium-java</artifactId>
            <version>2.48.2</version>
        </dependency>
    </dependencies>
</project>
```

Now we can open our new project in Eclipse and make the refactoring. How you can see below the exported code does look good. It has duplicates codes.



Figure 3.10: Eclipse

First of all, we should extract all use data to specific file. It is the design approach - Map UI.

TrackStudioData.java

```
package ru.parsentev;
/**
* File contains the trackstudio user data.
 * @author parsentev
 * @since 14.12.2015
 */
public interface TrackStudioData {
        String ROOT_LOGIN = "root";
        String ROOT_PWD = "root";
        String BREAD_LOGIN = "root";
        String BREAD_PWD = "root";
        String SMITH_LOGIN = "root";
        String SMITH_PWD = "root";
        String MANAGER_ROLE = "manager";
        String READER_ROLE = "reader";
        String CATEGORY_NAME = "book";
```

```
String WORKFLOW_NAME = "book";
String TASK_NAME = "the catcher in the rye";
String TR_IN = "in";
String TR_OUT = "out";
```

The next step is to split out main on the few little as we described in table tests ceases.

TrackStudioCreateLibraryConfiguration.java

```
@Test
public void cases() throws Exception {
    this.loginAsAdmin();
    this.createManagerRole();
    this.createWarAleque();
    this.createUserManager();
    this.createUserReader();
    this.createWorkflow();
    this.createCategory();
    this.takeTaskByReader();
}
```

We have the automation tests, which you can run the follow command mvn clean test. You can integrate it to CI easily.

3.6 Download the Maven project

Download

}

You can download the full source code of this example here: SeleniumAutomationTests.zip

Chapter 4

Selenium Interview Questions and Answers

4.1 Introduction

In this example we shall show the most popular interviews questions about Selenium tools and give you exhaustive answers. Questions cover all topics:

- Selenium 1
- Selenium 2
- Selenium IDE
- Selenium Standalone Server

In this article is shown theoretical questions and best practices, which is used by Selenium communities.

4.2 Interview Questions and Answers

What is Selenium?

Selenium is the complex of tools for automation testing the web applications. This tools is developed in Java. You can use different approaches for create the tests: using Selenium IDE, using one of popular programming languages (Java, C#, Perl, Python). Because right now web applications become very interactive it is to difficult to write automation tests for its. Selenium uses two kind of approaches for doing it: directly using browser api (new version), injecting JavaScript (old version).

What modules does Selenium consist?

Selenium has few different modules:

- Selenium IDE firefox plugin, The plugin has two modes : records user activities, run the tests by user activities.
- Selenium 1 (Selenium RC) library, which converts the code to JavaScript and injects it to the browser.
- Selenium 2 (Selenium WebDriver) library, which uses the browser api for testing.
- Selenium Server (Selenium Grid) server. It is used to run tests in different environments.

What the technology is used for searching elements in Selenium?

Selenium is used DOM for searching elements. The DOM can look like the Tree Structure.

How to find different type of elements in Selenium, explain few different ways?

Let's consider this question by simple HTML example

index.html
```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
   <title>Example page for Selenium Test</title>
</head>
<bodv>
<iframe name="tree" src="tree.html"></iframe>
    <div class="form">
        <form about="/" method="POST">
           User name : <input name="username" type="text">
            <a href="/agreements.html">User agreements</a>
        </form>
</div>
</body>
</html>
```

You can search the elements by different criterias:

- By Element ID WebElement element =driver.findElement(By.id("general"));
- By Class name WebElement element =driver.findElement(By.className("form"));
- By Tag name WebElement element =driver.findElement(By.tagName("iframe"));
- By Name WebElement element =driver.findElement(By.name("username"));
- By Link Text WebElement element =driver.findElement(By.linkText("User agreements"));
- By Partial Link Text WebElement element =driver.findElement(By.partialLinkText("agreements "));
- By XPATH List inputs =driver.findElements(By.xpath("//input"));
- Using JavaScript (WebElement) ((JavascriptExecutor)driver).executeScript("return \$('.gene ral')[0]");

How to fill different type of input elements?

Let's consired another HTML page:

create.html

```
<!DOCTYPE html>
<html lang="en">
<head>
   <meta charset="UTF-8">
   <title>Create new user</title>
</head>
<body>
<form action="/create" method="post">
   User name :<input type="text" name="username">
   Agreement : <input type="checkbox" name="agreement">
   Sex :
   <input type="radio" name="sex" value="male">Male
   <input type="radio" name="sex" value="female">Female
   Country :
   <select name="country">
        <option value="1">Country 1</option>
        <option value="2">Country 1</option>
   </select>
```

The main approach looks like : get necessary elements, fill the data.

```
/**
 * Test filling the form with different input elements.
 */
public void fillForm() {
        //fill the user name
        driver.findElement(By.name("username")).sendKeys("Petr");
        //checked agreement
        driver.findElement(By.name("agreement")).click();
        //choose sex
        driver.findElements(By.name("sex")).get(0).click();
        //select country
        WebElement select = driver.findElement(By.tagName("select"));
        select.findElements(By.tagName("option")).get(0).click();
        //fill description
        driver.findElement(By.name("desc")).sendKeys("Petr");
        //push on button
        driver.findElement(By.id("submit")).click();
}
```

How to switch between pupops, frames, windows?

Let's imagine that user opens the two windows index.html and create.html and we need to switch between them. You can do it by pointing the title of necessary windows. For example : driver.switchTo().window("Create new user"); We can use this approach to switch frames by its name. For example driver.switchTo().frame("tree");

What is Selenium IDE? When is it used?

Selenium IDE is firefox plugin. It has frendly user interface. It is used for developing the automation tests. Selenium community recommend to start to learn Selenium with this tools. The use cases look like to record user activites. IDE will gerenete the test code and after that you can run this tests. First of all you need to install the firefox plugin. It looks like below:



Figure 4.1: Download Selenium IDE

After you get your test cases code you can export it to favotire programming languages : Java, Ruby, Python, C#. It looks like below:

- -					
• •					
-					
1 117					
-					
Log Reference UI-Element Pollun					
Log Reference UI-Element Rollup					
ko					
K2					
NTER					
Cle					

Figure 4.2: Selenium Test IDE

🕘 ປ	ntitled (untitled suite) - Seleni	um IDE 2.9.0 *		
<u>F</u> ile	<u>E</u> dit <u>A</u> ctions <u>O</u> ptions He	lp		
	New Test Case Ctrl+N Open Ctrl+O	eks.com/	• () •	
	Save Test Case Ctrl+S	Source		
	Save Test Case <u>A</u> s			
	Export Test Case As	Ruby / RSpec / WebDriver	Value	
	Recent Test Cases 🔹 🕨	Ruby / Test::Unit / WebDriver		
	A <u>d</u> d Test Case Ctrl+D Properties	Ruby / RSpec / Remote Control Ruby / Test::Unit / Remote Control	n label=regexp:\s+-\s+-\sJava Tutorials	
	New Test Suite Open Test Suite Save Test Suite Save Test Suite As Export Test Suite As Recent Test Suites	Python 2 / unittest / webbiver Python 2 / unittest / Remote Control		
		Java / JUnit 4 / WebDriver Java / JUnit 4 / WebDriver Backed Java / JUnit 4 / Remote Control Java / JUnit 3 / Remote Control	▼	
Kuns	Close (X) Ctrl+W	C# / NUnit / WebDriver C# / NUnit / Remote Control	▼ Select Find	
Fallu				
Lo	g Reference UI-Element	Rollup	Info- Clear	
			Java Code Geeks	

Figure 4.3: Export IDE

After that you get the source code file then you can open in your favorite IDE. The code should look similar :

JavaCodeGeeksTests.java

```
package com.example.tests;
import java.util.regex.Pattern;
import java.util.concurrent.TimeUnit;
import org.junit.*;
import static org.junit.Assert.*;
import static org.hamcrest.CoreMatchers.*;
import org.openqa.selenium.*;
import org.openqa.selenium.firefox.FirefoxDriver;
import org.openqa.selenium.support.ui.Select;
public class JavaCodeGeeksTests {
 private WebDriver driver;
  private String baseUrl;
 private boolean acceptNextAlert = true;
 private StringBuffer verificationErrors = new StringBuffer();
  @Before
 public void setUp() throws Exception {
   driver = new FirefoxDriver();
   baseUrl = "https://www.javacodegeeks.com/";
  driver.manage().timeouts().implicitlyWait(30, TimeUnit.SECONDS);
```

```
}
@Test
public void testJavaCodeGeeksTests() throws Exception {
 driver.get(baseUrl + "/");
 driver.findElement(By.linkText("I reject the FREE eBooks")).click();
 driver.findElement(By.cssSelector("span.foobar-close-button")).click();
 new Select(driver.findElement(By.id("top-menu-mob"))).selectByVisibleText("regexp:\\s ↔
     +-\\s+-\\sJava Tutorials");
}
@After
public void tearDown() throws Exception {
 driver.quit();
  String verificationErrorString = verificationErrors.toString();
 if (!"".equals(verificationErrorString)) {
    fail(verificationErrorString);
  }
}
private boolean isElementPresent(By by) {
 trv {
   driver.findElement(by);
   return true;
  } catch (NoSuchElementException e) {
   return false;
  }
}
private boolean isAlertPresent() {
 trv {
   driver.switchTo().alert();
   return true;
 } catch (NoAlertPresentException e) {
    return false;
  }
}
private String closeAlertAndGetItsText() {
 try {
   Alert alert = driver.switchTo().alert();
   String alertText = alert.getText();
    if (acceptNextAlert) {
     alert.accept();
    } else {
      alert.dismiss();
    }
    return alertText;
  } finally {
    acceptNextAlert = true;
  }
}
```

What is implicit and explicit wait?

}

The all web apps load and fetch the data from server. sometimes it could not be faster. for this reason the test case should wait some events which should be happended on page. Selenium provides two kind of method to do this job.

• explicit wait - we don't know how manytimes the event should take so we predict some changes in UI. For example load the new elements : (new WebDriverWait(driver, 30)).until(ExpectedConditions.presenceOfEle mentLocated(By.id("comments"))); here we pointed 30 - timeout in milliseconds, if it exceeds the test will be

interuppeted. ExpectedConditions.presenceOfElementLocated(By.id("comments")) - expect the show new div block with id="comments"

• implicit wait - we predict that event takes definalty times. For example driver.manage().timeouts().implicitl yWait(30, TimeUnit.SECONDS);

What kind of tests does selemiun support? Selemiun supports variety tests cases:

- Functional tests
- Regressive tests
- Testing Static Content
- Testing Links
- Testing Dynamic Elements
- Ajax Tests

What is UI Map?

This is development approach when we separate tests data and login codes. For example we have some credential to auth in web site. we create the special class where we should store all such data and use it in tests codes. For example:

UserCredential.java

```
/**
 * Credential info.
 * @author parsentev
 * @since 26.11.2015
 */
public class UserCredential {
        public static final String LOGIN = "login";
        public static final String PASSWORD = "password";
}
```

So when you need to change the user credential, you go to this file and change only this code without searching all places where you use it.

What is Page Object Pattern? How is it used in Selenium?

This pattern is used in automation tests overall. The main advantages are hiding the implementation of auth details and splitting the all web apps tests on small undescended pages.

What is DDT and how is it supported in Selenium?

DDT is the data tests. Selenium does not have special mechanism for working with DDT. You can use another libraries for it. For example, read data from database by JDBC, or read data from file and so on.

What is Selenium Grid? Where is it used?

Selenium Grid is distributed servers which offers to run tests in different environments.

How to execute the Selenium tests in parallel?

It can be done by Selenium Gird or by using Thread in Java directly.

What kind of problems do you have when using Selenium?

First, it is not so faster to change the automation tests when UI in web app can be changed faster. However, these problems have all automation testing tools. Another problem is appear in dynamic generation IDs. If you use strategy search by ID you should change it to XPATH. One another problem is working with AJAX. Selenium supports AJAX executing but tester should predict how many times the ajax call takes. You can use explicit or implicit wait for solve with problem.

Can Selenium be integrated to CI?

Yes. It can, because all tests can be exported to programming languages and then it can be run independent from Selenium IDE by JUnit, TestNG and similar libraries

Can you use Selenium in commercial product?

Yes, you can. Selenium is distributed by Apache 2.0 license. It means that you can use it in commercial products free without paing any charge.

What kind of disadvantages does Selenium IDE have?

After you recorded the user activates Selenium IDE generates the case tests tables. This tests table does not support the loops, functions, conditions statements, so you cannot use all programming languages constructions.

What is Selenium Remote Control?

It is the test tool for web apps. It uses injecting JavaScript to browser for testing.

What advantages does Selenium have for compare with another web testing tools?

First, it is to start to write the tests, because you can record the user activity directly, Another thing is Selenium supports many programming languages : Java, C#, Python, Ruby.

Can you install the Selenium IDE to another browsers then FireFox?

No. Selenium IDE can be installed only to FireFox.

What kind of browsers can be executed the tests by Selenium?

- Firefox 3.x
- IE 6-8
- Safari 2-4
- Opera 8-10
- Chrome

Can you use Selenium for testing mobiles app?

Yes. if it can be run in mobiles browsers.

What is Selenese?

After you recorded the user activities, Selenium IDE generates the code, this code is written by Selenese.

How can you run the tests in HTTPS?

You need to use Selenium Server and configure the necessary securities environments.

How can you configure the Selenium Grid?

You can do in two ways: 1. adding special key when run the server, 2. using JSON config file.

What is HtmlUnitDriver?

HtmlUnitDriver is web driver which is used HtmlUnit engine. It is the faster web driver in Selenium.

Can you use navigations browser menu in Selenium tests?

Yes. you can. You can handle it with the following code:

```
driver.navigate().back(); driver.navigate().forward(); driver.navigate().refresh(); dri
ver.navigate().to();
```

How can you save the tests result?

You can take the screen shot if tests fail or you can store the testing result to database by JDBC.

Can you run the load tests by Selenium?

Yes, but Selenium is not good choose for such kind of tests. Because it is expense to open the multiple browser at the same time. if you need to test loading, you can use JMeter or Gattling.

4.3 Conclusion

In this acticle we explained most of frequently ask questions in interviews and gived the answers. Of course this is not all. if you want to improve your knowledge about Selemiun go to official web site Selenium Official Documentation.

Chapter 5

Selenium Standalone Server Example

5.1 Introduction

With this example we are going to demonstrate how to use and configure Selenium standalone servers (Selenium Grid). We are going to run the hub server and the two nodes. Each nodes will run the tests in different browsers.

Selenium Grid are the servers which compounds in distributed nodes. It offers you to run your selenium test on separate machines in different kinds environments. This is great opportunities, because your tests can run parallels and use different browsers for testing.

5.2 High level architecture

Below is shown high level architecture. It has the follow flow process. First we pass the tests to hub which send this tests to specific nodes where all tests will be executed.





Figure 5.1: Selenium Grid. High level architecture

So all tests and nodes can be located on different machines. Such architecture can be scaled easily.

Selenium Grid

5.3 Configuration

Before we can start to configure the Selenium Grid we should download the necessary library. Selenium Grid consists only from one JAR file. Go to the official site link and download the Selenium Server JAR file - selenium-server-standalone-2.48.2.jar. This jar has the good help information. We should run this jar with the key -h that to print help information on screen.

java -jar selenium-server-standalone-2.48.2.jar -h

This help information has all explanation about supported keys. This keys are used for configuration the instance server. First of all we should run this jar with key -role hub. It means the this instance will be the hub server. It will be taken all receiving tests and distributed to the specific nodes server. The hub server is run on 4444 port by default.

```
C:\Tools>java -jar selenium-server-standalone-2.48.2.jar -role hub
10:29:14.270 INFO - Launching Selenium Grid hub
2015-11-19 10:29:15.458:INFO::main: Logging initialized @1362ms
10:29:15.479 INFO - Will listen on 4444
2015-11-19 10:29:15.563 INFO - Will listen on 4444
2015-11-19 10:29:15.663:INFO:osjs.Server:main: jetty-9.2.z-SNAPSHOT
2015-11-19 10:29:15.631:INFO:osjsh.ContextHandler:main: Started o.s.j.s.ServletC
ontextHandler@13f88ab{/,null,AVAILABLE}
2015-11-19 10:29:15.770:INFO:osjs.ServerConnector:main: Started ServerConnector@
646db9{HTTP/1.1}{0.0.0.0:4444}
```

```
41 / 65
```

```
2015-11-19 10:29:15.771:INFO:osjs.Server:main: Started @1675ms
10:29:15.772 INFO - Nodes should register to https://192.168.0.102:4444/grid/regi
ster/
10:29:15.772 INFO - Selenium Grid hub is up and running
```

The secondary step is to run the node instance. It can be done with key -role node as shown below. The same time we should point where is located our hub server by key -hub https://localhost:4444/grid/register

```
C:\Tools>java -jar selenium-server-standalone-2.48.2.jar -role node -hub http:/
/localhost:4444/grid/register
10:31:08.635 INFO - Launching a Selenium Grid node
10:31:09.999 INFO - Java: Oracle Corporation 25.45-b02
10:31:10.000 INFO - OS: Windows 7 6.1 x86
10:31:10.009 INFO - v2.48.0, with Core v2.48.0. Built from revision 4lbccdd
10:31:10.089 INFO - Driver class not found: com.opera.core.systems.OperaDriver
10:31:10.090 INFO - Driver provider com.opera.core.systems.OperaDriver is not re
gistered
10:31:10.153 INFO - Selenium Grid node is up and ready to register to the hub
10:31:10.215 INFO - Starting auto registration thread. Will try to register ever
y 5000 ms.
10:31:10.216 INFO - Registering the node to the hub: https://localhost:4444/grid/
register
10:31:10.254 INFO - The node is registered to the hub and ready to use
```

How you can see above the configuration process can be done by adding the keys in command line. But Selenium Server supports another variant of configuration by file configuration in JSON format.

Firstly we should create the file which has the name - firefox_node.json. It can have any appropriate name.

```
{
  "capabilities":
      [
        {
          "browserName": "*firefox",
          "maxInstances": 1,
          "seleniumProtocol": "WebDriver"
        }
      ],
  "configuration":
  {
    "proxy": "org.openqa.grid.selenium.proxy.DefaultRemoteProxy",
    "maxSession": 5,
    "port": 6543,
    "host": 127.0.0.1,
    "register": true,
    "registerCycle": 5000,
    "hubPort": 4444,
    "hubHost": 127.0.0.1
  }
}
```

We pointed there that all tests should be run in firefox. Now we can run the new node instance with this configurations. We use -nodeConfig that to point which config file to use.

```
C:\Tools>java -jar selenium-server-standalone-2.48.2.jar -role node -nodeConfig
firefox_node.json
11:36:22.804 INFO - Launching a Selenium Grid node
11:36:23.789 INFO - Java: Oracle Corporation 25.45-b02
11:36:23.789 INFO - OS: Windows 7 6.1 x86
11:36:23.798 INFO - v2.48.0, with Core v2.48.0. Built from revision 41bccdd
11:36:23.884 INFO - Driver class not found: com.opera.core.systems.OperaDriver
11:36:23.885 INFO - Driver provider com.opera.core.systems.OperaDriver is not re
```

gistered 11:36:23.973 INFO - Selenium Grid node is up and ready to register to the hub 11:36:24.028 INFO - Starting auto registration thread. Will try to register ever y 5000 ms. 11:36:24.029 INFO - Registering the node to the hub: https://127.0.0.1:4444/grid/ register 11:36:24.041 INFO - The node is registered to the hub and ready to use

Sometimes you can need only one instance server. For this reason you should run the selenium server without keys. All tests will be executed in this instance in this case.

Another good opportunities are configure special browsers. For example, below we set the chrome browser environment.

0:4	Администратор: C:\Windows\System32\cmd.exe - java -jar selenium-server-standalone-2.48.2.jar 💷 💷 🔤
C: er 14 Se 14	<pre>\Tools>java -jar selenium-server-standalone-2.48.2.jar -Dwebdriver.chrome.driv >="c:\Program Files <x86>\Google\Chrome\Application\chrome.exe" 4:55:40.439 INFO - Launching a standalone Selenium Server etting system property webdriver.chrome.driver to c:\Program Files <x86>\Google Chrome\Application\chrome.exe 4:55:40.662 INFO - Java: Oracle Corporation 25.45-b02 4:55:40.663 INFO - OS: Windows 7 6.1 x86</x86></x86></pre>
14	1:55:40.677 INFO - v2.48.0, with Core v2.48.0. Built from revision 41bccdd 1:55:40.781 INFO - Driver class not found: com.opera.core.systems.OperaDriver 1:55:40.782 INFO - Driver provider com.opera.core.systems.OperaDriver is not re
9-14 1-14	4:55:41.112 INFO - RemoteWebDriver instances should connect to: http://127.0.0. :4444/wd/hub 4:55:41.113 INFO - Selenium Server is up and running
	Java Code Geeks
	JAWA 2 JAWA DEVELOPERS RESOURCE CENTER

Figure 5.2: Single Selenium Instance. Chrome Config.

We set the properties : -Dwebdriver.chrome.driver="c:Program Files (x86)GoogleChromeApplicatio nchrome.exe"

Right now we have the three instances: one the hub and two nodes and one single server. Let's start to run our tests.

5.4 Run tests

First of all we should create the new maven project.

pom.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="https://maven.apache.org/POM/4.0.0"</pre>
```

```
xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
 xsi:schemaLocation="https://maven.apache.org/POM/4.0.0 https://maven.apache.org/ ↔
    xsd/maven-4.0.0.xsd">
<modelVersion>4.0.0</modelVersion>
<groupId>ru</groupId>
<artifactId>parsentev</artifactId>
<version>1.0-SNAPSHOT</version>
<dependencies>
        <dependency>
                <groupId>junit</groupId>
                <artifactId>junit</artifactId>
                <version>4.12</version>
        </dependency>
        <dependency>
                <groupId>org.seleniumhq.selenium</groupId>
                <artifactId>selenium-java</artifactId>
                <version>2.48.2</version>
        </dependency>
</dependencies>
```

```
</project>
```

That we need to add the simple tests.

ruparsentevSeleniumStantaloneServerTest.java

```
package ru.parsentev;
import com.thoughtworks.selenium.DefaultSelenium;
import com.thoughtworks.selenium.Selenium;
import org.junit.Test;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.remote.DesiredCapabilities;
import org.openqa.selenium.remote.RemoteWebDriver;
import java.net.MalformedURLException;
import java.net.URL;
import static org.hamcrest.core.Is.is;
import static org.junit.Assert.assertThat;
/**
* Tests for selenium standalone server.
* @author parsentev
* @since 19.11.2015
 */
public class SeleniumStandaloneServerTest {
        @Test
        public void executeFirefoxDriver() throws MalformedURLException {
                this.execute(DesiredCapabilities.firefox());
        }
        @Test
        public void executeChrome() throws MalformedURLException {
                this.execute(DesiredCapabilities.chrome());
        }
        private void execute(final DesiredCapabilities capability) throws ↔
           MalformedURLException {
```

Now we can run the our test.

mvn clean test

We can see details information about tests process on the node logs or screens. You should see somethings similar:

```
12:14:25.891 INFO - Executing: [new session: Capabilities [{browserName=firefox,
version=, platform=ANY}]])
12:14:25.903 INFO - Creating a new session for Capabilities [{browserName=firefo
x, version=, platform=ANY}]
12:14:30.143 INFO - Done: [new session: Capabilities [{browserName=firefox, vers
ion=, platform=ANY}]]
12:14:30.196 INFO - Executing: [get: https://www.javacodegeeks.com/])
12:14:34.283 INFO - Done: [get: https://www.javacodegeeks.com/]
12:14:34.299 INFO - Executing: [find element: By.name: s])
12:14:34.671 INFO - Done: [find element: By.name: s]
12:14:34.689 INFO - Executing: [send keys: 0 [[FirefoxDriver: firefox on WINDOWS
 (2ca50141-8460-4012-bec4-b291e4042f55)] -> name: s], [selenuim]])
12:14:34.774 INFO - Done: [send keys: 0 [[FirefoxDriver: firefox on WINDOWS (2ca
50141-8460-4012-bec4-b291e4042f55)] -> name: s], [selenuim]]
12:14:34.784 INFO - Executing: [submit: 0 [[FirefoxDriver: firefox on WINDOWS (2
ca50141-8460-4012-bec4-b291e4042f55)] -> name: s]])
12:14:39.270 INFO - Done: [submit: 0 [[FirefoxDriver: firefox on WINDOWS (2ca501
41-8460-4012-bec4-b291e4042f55)] -> name: s]]
12:14:39.281 INFO - Executing: [get title])
12:14:39.311 INFO - Done: [get title]
12:14:39.327 INFO - Executing: [delete session: a459baef-2980-4fcc-8093-4ff4eecb
f03f1)
12:14:39.806 INFO - Done: [delete session: a459baef-2980-4fcc-8093-4ff4eecbf03f]
```

When the test is received by node it looks appropriate browser in local machine. Then the node opens this browser and starts to do the tests.

It can look like below:



Figure 5.3: Firefox selenium tests



Figure 5.4: Chrome selenium tests

5.5 Download the Code Project

Download

You can download the full source code of this example here: Selenium

Chapter 6

Selenium JUnit Example

6.1 Introduction

In this article, we are going to show how you can write automation tests by Selenium and JUnit.

Selenium is tools for building automation tests. Selenium can be used only for testing web applications. When Selenium executes the test, it injects the JavaScript codes to browser or it uses the native browser API. It does not mean that you should write all codes only on JavaScript. Selenium supports all most popular programming languages : Java, C#, Python, Ruby and so.

JUnit is a unit testing framework for the Java programming language. In this example, we will integrate Selenium to this framework. Actually, Selenium IDE has all functionality to write, build and execute automation tests, but if you want to execute tests independently from Selenium IDE you need to use JUnit or another libraries for automation tests.

6.2 Record test cases

The simplest way to get the base test cases code is to record user activities in Selenium IDE. Firstly, you should install Selenium IDE. Actually, Selenium IDE is the Firefox add-ons. After you install this plugin, you can see the Selenium IDE button in the right top corner in Firefox. This plugin is supported only by Firefox.



Figure 6.2: Plugins page for Selenium IDE

		Seleniur
😻 Selenium IDE 2.9.1		X
Eile Edit Actions Options Help	2	
Base URL http://examples.javacod	egeeks.com/	÷
Fast Slow	0	Q - C
Test Case	Table Source	
Untitled	Command Target	Value
	Command Target	
Runs:	Value	
Runs: Failures:	0 Value	

Figure 6.3: Base view of Selenium IDE

Then you need to turn the record button on and start to navigate in the necessary web site. In this case, we want to test the search function in the https://ebay.com. EBay has the advanced search functions. It is the good example to show most useful abilities to test web apps by Selenium IDE.

Advanced Search	n				
ltems	Find Items				
Find items	Enter keywords or item num	nber			
By seller By item number	JUnit Selenium	b	Exact words, exact order	•	
Stores	Exclude words from your searc				
Items in stores Find Stores	See general search tips or usin In this category: Books	ng advanced search optio	ins		
monnooro					
Find a member Find contact information	Search	nium IDE 2.9.1 *			
Find a member Find contact information	Search Untitled (untitled suite) - Sele <u>File</u> <u>Edit</u> <u>Actions</u> <u>Options</u> H	nium IDE 2.9.1 * Help			
Find a member Find contact information	Search Untitled (untitled suite) - Sele <u>Eile Edit Actions Options H</u> Base URL http://www.ebay.com	nium IDE 2.9.1 * Help n/			
Find a member Find contact information	Search Untitled (untitled suite) - Sele Eile Edit Actions Options H Base URL http://www.ebay.com East Slow	nium IDE 2.9.1 * Help n/			
Find a member Find contact information	Search Untitled (untitled suite) - Sele File Edit Actions Options H Base URL http://www.ebay.com Fast Slow Test Case	nium IDE 2.9.1 * Help n/ Source			
Find a member Find contact information	Search Untitled (untitled suite) - Sele File Edit Actions Options H Base URL http://www.ebay.com Fast Slow Test Case Untitled *	Help n/ Table Source	Target		√alue
Find a member Find contact information	Search Untitled (untitled suite) - Sele File Edit Actions Options H Base URL http://www.ebay.com Fast Slow Test Case Untitled *	nium IDE 2.9.1 * Help n/ Source Command click	Target id=LH_Complete		/alue
Find a member Find contact information	Search Untitled (untitled suite) - Sele File Edit Actions Options H Base URL http://www.ebay.com Fast Slow Test Case Untitled *	Help n/ Table Source Command click click	Target id=LH_Complete id=LH_Complete		/alue
Find a member Find contact information	Search Untitled (untitled suite) - Sele File Edit Actions Options H Base URL http://www.ebay.com Fast Slow Test Case Untitled *	nium IDE 2.9.1 * Help n/ Table Source Command click click click	Target id=LH_Complete id=LH_Complete id=LH_Complete		/alue
Find a member Find contact information	Search Untitled (untitled suite) - Sele File Edit Actions Options H Base URL http://www.ebay.com Fast Slow Test Case Untitled *	Help n/ Table Source Command click click click click	Target id=LH_Complete id=LH_Complete id=LH_Complete id=LH_Complete id=LH_Complete id=LH_Complete	Code (Value Geek

Figure 6.4: Recourd user cases

Now we are ready to export this recorded test cases into your favorite programming . In this example, we use Java. For this reason, we are going to export test cases to Java and ask the Selenium IDE that it generate the necessary structures for JUnit framework too.

_			
	New Test Case Ctrl+N Open Ctrl+O Save Test Case Ctrl+S Save Test Case As	Contraction Contra	- (
Ex	Export Test Case As •	C# / NUnit / WebDriver Value	
	Recent Test Cases 🔹 🕨	C# / NUnit / Remote Control JUnit	*
	A <u>d</u> d Test Case Ctrl+D Properties	Java / JUnit 4 / WebDriver label=Books	
	New Test Suite	Java / JUnit 4 / WebDriver Backed btn.btn-prim	
New Test Suite Open Test Suite Save Test Suite Save Test Suite As Export Test Suite As Recent Test Suites	Java / JUnit 4 / Remote Control Java / JUnit 3 / Remote Control Java / TestNG / Remote Control Python 2 / unittest / WebDriver Python 2 / unittest / Remote Control Ruby / RSpec / WebDriver	- - -	
	Close (X) Ctrl+W	Ruby / Test::Unit / WebDriver	
ilur	res:	Ruby / RSpec / Remote Control	
	1 1	Ruby / Test::Unit / Remote Control	
Log	Reference UI-Elemen	Rollup	
Ar	ect(selectLocator, optionLoca guments: • selectLocator - an eleme • optionLocator - an option lect an option from a drop-dow	or) t locator identifying a drop-down menu locator (a label by default) n using an option locator.	Iek

Figure 6.5: Export to Java with JUnit structures

6.3 Integrate to JUnit

The next step is to create the new maven project with JUnit and Selenium dependencies. We will create this new project from default archetype by this follows command

mvn -B archetype:generate -DarchetypeGroupId=org.apache.maven.archetypes -DgroupId=ru. ↔
parsentev.app -DartifactId=EbayAdvancedSearch

C:\projects>mvn -B archetype:generate -DarchetypeGroupId=org.apache.maven.archet ypes -DgroupId=ru.parsentev.app -DartifactId=EbayAdvancedSearch [INFO] Scanning for projects
[INFO] [INFO] [INFO] Building Maven Stub Project (No POM) 1 [INFO]
[INFO] [INFO] >>> maven-archetype-plugin:2.3:generate (default-cli) > generate-sources @ standalone-pom >>> [INFO]
[INFO] [INFO] <<< maven-archetype-plugin:2.3:generate (default-cli) < generate-sources @ standalone-pom <<< [INFO]
[[NFO] maven-archetype-plugin:2.3:generate (default-cli) @ standalone-pom
- [INFO] Generating projent in Batch mode [INFO] No archetype defined. Using maven-archetype-quickstart (org.apache.maven. archetypes:maven-archetype-quickstart:1.0) [INFO]
 [INFO] Using following parameters for creating project from Old (1.x) Archetype: maven-archetype-quickstart:1.0 [INFO]
 [INFO] Parameter: groupId, Value: ru.parsentev.app [INFO] Parameter: packageName, Value: ru.parsentev.app [INFO] Parameter: package, Value: ru.parsentev.app [INFO] Parameter: artifactId, Value: EbayAdvancedSearch [INFO] Parameter: basedir, Value: C:\projects [INFO] Parameter: version, Value: 1.0-SNAPSHOT [INFO] project created from Old (1.x) Archetype in dir: C:\projects\EbayAdvanced Search
[INFO] BUILD SUCCESS
[INFO]
C:\projects>

Figure 6.6: Build the new project

Now, we can open this project in Eclipse.



Figure 6.7: Import to Eclipse

Next, we need to move the export Java code from Selenium IDE to the new project. You should put this file to test directory. In my case, it is srctestjavaruparsentevapp



Java - EbayAdvancedSearch/src/test/java/ru/parsentev/app/EbayAdvancedSearch.java - Eclipse File Edit Source Refactor Navigate Search Project Run Window Help 📸 🕶 🖆 🖷 🐚 🔌 🎋 🖌 🗶 🕶 🥵 🖉 🥴 😂 😂 🖉 🖓 🖓 🖓 🖓 🗐 👘 🖕 🖓 🗸 🍄 🍊 🔶 🔶 🔶 Quick Ad 📲 Package Explorer 🔀 - 8 🛃 *EbayAdvancedSearch.java 🔀 2 1 package ru.parsentev.app; ∇ E 8 2 EbayAdvancedSearch 3 3 import com.thoughtworks.selenium.Selenium; ▲ ([™]) src/main/java 13 a 🌐 ru.parsentev.app 14 public class EbayAdvancedSearch { 15 private Selenium selenium; App.java 16 ▲ 🖶 src/test/java 179 **@Before** ▲ ru.parsentev.app 18 public void setUp() throws Exception { AppTest.java 19 WebDriver driver = new FirefoxDriver(); EbayAdvancedSearch.java String baseUrl = "http://www.ebay.com/" 20 selenium = new WebDriverBackedSelenium(driver, baseUrl); ▷ Mathematics JRE System Library [J2SE-1.5] 21 22 } Maven Dependencies 23 Src 240 @Test - target dCanaala () there are

Figure 6.8: Exported Java code

How you can see this code is highlighted by Selenium. it is happened, because we need to add the Selenium library to dependencies block.

pom.xml

```
<project xmlns=&quot;https://maven.apache.org/POM/4.0.0&quot; xmlns:xsi=&quot;https://www. </pre>
   w3.org/2001/XMLSchema-instance"
         xsi:schemaLocation="https://maven.apache.org/POM/4.0.0 https://maven.apache. ↔
            org/maven-v4_0_0.xsd">
        <modelVersion>4.0.0</modelVersion>
        <groupId>ru.parsentev.app</groupId>
        <artifactId>EbayAdvancedSearch</artifactId>
        <packaging>jar</packaging>
        <version>1.0-SNAPSHOT</version>
        <name>EbayAdvancedSearch</name>
        <url>https://maven.apache.org</url>
        <dependencies>
                <dependency>
                        <groupId>org.seleniumhq.selenium</groupId>
                        <artifactId>selenium-java</artifactId>
                        <version>2.48.2</version>
                        <scope>test</scope>
                </dependency>
                <dependency>
                        <groupId>junit</groupId>
                        <artifactId>junit</artifactId>
                        <version>4.12</version>
                        <scope>test</scope>
                </dependency>
        </dependencies>
</project>
```

You can find the full source code of this case below.

EbayAdvancedSearch.java

```
package ru.parsentev.app;
import com.thoughtworks.selenium.Selenium;
import org.openqa.selenium.By;
```

```
55 / 65
```

```
import org.openga.selenium.firefox.FirefoxDriver;
import org.openga.selenium.WebDriver;
import com.thoughtworks.selenium.webdriven.WebDriverBackedSelenium;
import org.junit.After;
import org.junit.Before;
import org.junit.Test;
import static org.junit.Assert.*;
import java.util.regex.Pattern;
import static org.apache.commons.lang3.StringUtils.join;
public class EbayAdvancedSearch {
        private Selenium selenium;
        @Before
        public void setUp() throws Exception {
                WebDriver driver = new FirefoxDriver();
                String baseUrl = "https://www.ebay.com/";
                selenium = new WebDriverBackedSelenium(driver, baseUrl);
        }
        @Test
        public void testEbayAdvancedSearch() throws Exception {
                selenium.open("/");
                selenium.waitForPageToLoad("30000");
                selenium.click("id=gh-as-a");
                selenium.type("id=_nkw", "JUnit");
                selenium.select("id=e1-1", "value=267");
                selenium.click("id=LH_TitleDesc");
                selenium.click("css=button.btn.btn-prim");
                selenium.waitForPageToLoad("30000");
                for (int second = 0;; second++) {
                        if (second >= 60) fail("timeout");
                        try { if (selenium.isElementPresent("css=span.listingscnt")) break; ↔
                             } catch (Exception e) {}
                        Thread.sleep(1000);
                }
                assertEquals("Classified: 488", selenium.getText("css=span.listingscnt"));
        }
        @After
        public void tearDown() throws Exception {
                selenium.stop();
        }
```

Now, we can run this project by this follows command mvn test

C:\projects\EbayAdvancedSearch>mvn_test
[INFO] Scanning for projects
[INF0]
[INFO] Building EbayAdvancedSearch 1.0-SNAPSHOT
[INFO] maven-resources-plugin:2.6:resources (default-resources) @ EbayAdvanc
edSearch [WARNING] Using platform encoding (Cn1251 actually) to conv filtered resources.
i.e. build is platform dependent!
[INFO] skip non existing resourceDirectory C:\projects\EbayAdvancedSearch\src\ma in\wesources
[INFO]
[INFO] maven-compiler-plugin:3.1:compile (default-compile) @ EbayAdvancedSea
ren [[NFO] Changes detected - recompiling the module!
[WARNING] File encoding has not been set, using platform encoding Cp1251, i.e. b
uild is platform dependent! [INFO] Compiling 1 source file to C:\projects\Fbau0duancedSearch\target\classes
[INFO]
[INFO] maven-resources-plugin:2.6:testResources (default-testResources) @ Eb
[WARNING] Using platform encoding (Cp1251 actually) to copy filtered resources.
i.e. build is platform dependent!
LINFVJ skip non existing resourceDirectory C:\projects\EbayAdvancedSearch\src\te st\resources
[INFO]
[INFO] maven-compiler-plugin:3.1:testCompile (default-testCompile) @ EbayAdv
[INFO] Changes detected - recompiling the module!
[WARNING] File encoding has not been set, using platform encoding Cp1251, i.e. b
[INFO] Compiling 2 source files to C:\projects\EbayAdvancedSearch\target\test-c]
LWAKNINGJ /C:/projects/EbayAdvancedSearch/src/test/java/ru/parsentev/app/EbayAdv ancedSearch_jaua:_C:\projects\EbayAdvancedSearch\src\test\java\ru/parsentev\app/EbayAdv
EbayAdvancedSearch.java uses or overrides a deprecated API.
[WARNING] /C:/projects/EbayAdvancedSearch/src/test/java/ru/parsentev/app/EbayAdv
[INFO]
[INFO] maven-surefire-plugin:2.12.4:test (default-test) @ EbayAdvancedSearch
[INFO] Surefire report directory: C:\projects\EbayAdvancedSearch\target\surefire
-reports
Running ru.parsentev.app.AppTest
lests run: 1, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.076 sec
Results :
Tests run: 1 Failures: 0 Errors: 0 Skinned: 0 (100) Loug Codo Coolo
Java bulle uters
[INF0]

Figure 6.9: Run tests

6.4 Conclusion

In this article, we have shown how you can integrate Selenium and JUnit frameworks. If you want to get more information about this framework, please vitis official web sites: Selenium and JUnit.

6.5 Download the source code

Download

You can download the full source code of this example here: Selenium JUnit

Chapter 7

Selenium Grid Example

7.1 Introduction

In this tutorial, we are going to explain what Selenium Grid is and how you can use it in your project. We are going to install and configure the Selenium Grid, write and execute the tests on it. Selenium Grid is one of tool from Selenium framework. It is the distributed system for execution tests.

It has few benefits:

- You can execute tests in parallel, so it can reduce the execution time.
- You can set particular environments for tests.

Actually. It is the daunting task to reduce the tests executions time, because some of functional tests spent about hour for executions process. Selenium team offers the great solutions for it. It uses the separate nodes, which compound in full distributed system. For this reason, it can be scaled easily.

Below you can see the highly level architecture.



Figure 7.1: Selenium Grid. High level

7.2 Installing Selenium Grid

Firstly, you should download the necessary library. The great thing about Selenium Grid is that you need only one JAR file. You need to go to the official web site seleniumhq.org/download and download the Selenium Server. It can be configured to Selenium Grid.

Below you can see the download page:

SeleniumH Browser Automation	Q <u>edit this page</u> search selenium: Constant of Support Ab			
Salanium Downloads	Downloads			
Latest Releases	Below is where you can find the latest releases of all the Selenium components. You can als list of <u>previous releases</u> , <u>source code</u> , and additional information for <u>Maven users</u> (Maven is			
Previous Releases	popular Java build tool).			
Source Code	Selenium Standalone Server			
Maven Information	The Selenium Server is needed in order to run either Selenium RC style scripts or Remote Sel WebDriver ones. The 2.x server is a drop-in replacement for the old Selenium RC server and designed to be backwards compatible with your existing infrastructure.			
Donate to Selenium	Download version 2.48.1			
with PayPal	To use the Selenium Server in a Grid configuration see the wiki page.			
Donate	The Internet Explorer Driver Server			
through sponsorship	This is required if you want to make use of the latest and greatest features of the WebDriver InternetExplorerDriver. Please make sure that this is available on your \$PATH (or %PATH% or			
You can <u>sponsor the Selenium</u> project if you'd like some public recognition of your	Download version 2.49.0 for (recommended) <u>32 bit Windows IE</u> or <u>64 bit Windows IE</u> or <u>64</u>			

Figure 7.2: Selenium Grid download page

Right now, you can run this jar and use Selenium Grid. You can find the details about it in the next section.

CHANGELOG

7.3 Usage cases

generous contribution.

How we said before, all you need is the Selenium Server JAR. You can run it this key --help, that get help information about supported operations.

```
c:\Users\parsentev\Downloads>java -jar selenium-server-standalone-2.49.1.jar -help
Running as a standalone server:
Usage: java - jar selenium-server. jar [-interactive] [options]
  -port : the port number the selenium server should use
    (default 4444)
  -timeout : an integer number of seconds we should allow a
   client to be idle
  -browserTimeout : an integer number of seconds a browser is
    allowed to hang
  -interactive: puts you into interactive mode. See the tutorial for
   more details
  -singleWindow: puts you into a mode where the test web site
   executes in a frame. This mode should only be selected if the
    application under test does not use frames.
  -profilesLocation: Specifies the directory that holds the profiles
   that java clients can use to start up selenium. Currently
    supported for Firefox only.
  -forcedBrowserMode : sets the browser mode to a single
    argument (e.g. "*iexplore") for all sessions, no matter what is
   passed to getNewBrowserSession
  -forcedBrowserModeRestOfLine : sets the browser mode to
```

all the remaining tokens on the line (e.g. "*custom /some/random/place/iexplore.exe") for all sessions, no matter what is passed to getNewBrowserSession -userExtensions : indicates a JavaScript file that will be loaded into selenium -browserSessionReuse: stops re-initialization and spawning of the browser between tests -avoidProxy: By default, we proxy every browser request; set this flag to make the browser use our proxy only for URLs containing '/selenium-server' -firefoxProfileTemplate : normally, we generate a fresh empty Firefox profile every time we launch. You can specify a directory to make us copy your profile directory instead. -debug: puts you into debug mode, with more trace information and diagnostics on the console -browserSideLog: enables logging on the browser side; logging messages will be transmitted to the server. This can affect performance. -ensureCleanSession: If the browser does not have user profiles, make sure every new session has no artifacts from previous sessions. For example, enabling this option will cause all user cookies to be archived before launching IE, and restored after IE is closed. -trustAllSSLCertificates: Forces the Selenium proxy to trust all SSL certificates. This doesn't work in browsers that don't use the Selenium proxy. -log : writes lots of debug information out to a log file and disables logging to console -logLongForm: writes information out to console in long format (for debugging purpose) -htmlSuite : Run a single HTML Selenese (Selenium Core) suite and then exit immediately, using the specified browser (e.g. "*firefox") on the specified URL (e.g. "https://www.google.com"). You need to specify the absolute path to the HTML test suite as well as the path to the HTML results file we'll generate. -proxyInjectionMode: puts you into proxy injection mode, a mode where the selenium server acts as a proxy server for all content going to the test application. Under this mode, multiple domains can be visited, and the following additional flags are supported: -dontInjectRegex : an optional regular expression that proxy injection mode can use to know when to bypss injection -userJsInjection : specifies a JavaScript file which will then be injected into all pages -userContentTransformation : a regular expression which is matched against all test HTML content; the second is a string which will replace matches. These flags can be used any number of times. A simple example of how this could be useful: if you add "-userContentTransformation https http" then all "https" strings in the HTML of the test application will be changed to be "http". This synopsis lists options available in standalone role only. To get help on the options available for other roles run the server with -help option and the corresponding -role option value.

How you can see in the architecture diagram, firstly, we need to run the HUB nodes. It will takes all receiving tests and route to participated nodes, which has appropriate environment.

You need to run the JAR with follow key -role hub

c:\Users\parsentev\Downloads>java -jar selenium-server-standalone-2.49.1.jar -ro le hub 18:03:01.618 INFO - Launching Selenium Grid hub 2016-01-23 18:03:02.766:INFO::main: Logging initialized @1342ms 18:03:02.846 INFO - Will listen on 4444 2016-01-23 18:03:02.856:INFO:osjs.Server:main: jetty-9.2.z-SNAPSHOT 2016-01-23 18:03:02.896:INFO:osjs.ContextHandler:main: Started o.s.j.s.ServletC ontextHandler@1e2adc7/,null,AVAILABLE> 2016-01-23 18:03:02.976:INFO:osjs.ServerConnector:main: Started ServerConnector@ 4a2ea6(HTTP/1.1)<0.0.0:0:4444> 2016-01-23 18:03:02.976:INFO:osjs.Server:main: Started @1553ms 18:03:02.976 INFO - Nodes should register to http://192.168.1.234:4444/grid/regi ster/ 18:03:02.976 INFO - Selenium Grid hub is up and running



Now, we need to add the node. Node is used for executing the particular test on special environments. Therefore it means that you can run the node on separate machine. You can do by following command java -jar selenium-server-standalone-2.49.1.jar -role node -hub https://localhost:4444/grid/register



Figure 7.4: Selenium GRID Node

Selenium Grid is run and we can use it.

Let's create the simple maven project in order to demonstrate how it works.

pom.xml

```
<?xml version=&quot;1.0&quot; encoding=&quot;UTF-8&quot;?>
<project xmlns=&quot;https://maven.apache.org/POM/4.0.0&quot;
         xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="https://maven.apache.org/POM/4.0.0 https://maven.apache. ↔
            org/xsd/maven-4.0.0.xsd">
        <modelVersion>4.0.0</modelVersion>
       <groupId>ru</groupId>
       <artifactId>parsentev</artifactId>
        <version>1.0-SNAPSHOT</version>
        <dependencies>
               <dependency>
                       <groupId>junit</groupId>
                       <artifactId>junit</artifactId>
                       <version>4.12</version>
               </dependency>
               <dependency>
                       <groupId>org.seleniumhq.selenium</groupId>
                       <artifactId>selenium-java</artifactId>
                        <version>2.48.2</version>
               </dependency>
        </dependencies>
</project>
```

How you can see, we need to add the Selenium library in dependencies. Then, let's create the simple test cases, when we want to test the search function in https://www.javacodegeeks.com/

ruparsentevSeleniumStantaloneServerTest.java

```
package ru.parsentev;
import com.thoughtworks.selenium.DefaultSelenium;
import com.thoughtworks.selenium.Selenium;
import org.junit.Test;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.remote.DesiredCapabilities;
import org.openqa.selenium.remote.RemoteWebDriver;
import java.net.MalformedURLException;
import java.net.URL;
import static org.hamcrest.core.Is.is;
import static org.junit.Assert.assertThat;
/**
* Tests for selenium standalone server.
* @author parsentev
 * @since 19.11.2015
 */
public class SeleniumStandaloneServerTest {
        @Test
        public void executeFirefoxDriver() throws MalformedURLException {
                this.execute(DesiredCapabilities.firefox());
        }
        @Test
        public void executeChrome() throws MalformedURLException {
                this.execute(DesiredCapabilities.chrome());
```

```
private void execute(final DesiredCapabilities capability) throws ↔
           MalformedURLException {
               WebDriver driver = new RemoteWebDriver(
                               new URL("https://localhost:4444/wd/hub"), capability
                );
                driver.get("https://www.javacodegeeks.com/");
               WebElement element = driver.findElement(By.name("s"));
                element.sendKeys("selenuim");
                element.submit();
                assertThat(
                                driver.getTitle(),
                                is("You searched for selenuim | Java Code Geeks")
                );
                driver.quit();
        }
}
```

In example above, we pointed that we want to execute the test on Chrome and Firefox browsers by following commands: DesiredCapabilities.chrome().

You can set more specific requirments for execution environments by additional API, which offer the Selenium library:

```
capability.setBrowserName();
capability.setPlatform();
capability.setVersion()
capability.setCapability(,);
```

The same time, you need to configure your nodes for particular environment. Selenium Grid supports two way, how you can do it.

- It is used command-line key. For example, we want that this node execute only tests in Internet Explorer 9. We can configure in like: -browser browserName=iexplorer, version=9, platform=WINDOWS
- It is used JSON configuration file.

```
{
  "capabilities":
     [
        {
          "browserName": "*firefox",
          "maxInstances": 1,
          "seleniumProtocol": "WebDriver"
        }
      ],
  "configuration":
  {
    "proxy": "org.openqa.grid.selenium.proxy.DefaultRemoteProxy",
    "maxSession": 5,
    "port": 6543,
    "host": 127.0.0.1,
    "register": true,
    "registerCycle": 5000,
    "hubPort": 4444,
    "hubHost": 127.0.0.1
 }
}
```

7.4 Conclusion

In this article, we explained what Selenium Grid is and shown how to configure, run, execute tests. We could not cover all narrow things about Selenium Grid, so if you want to improve your knowledge about Selenium and particular about Selenium Grid, please, visit the official web site seleniumhq.org

7.5 Download the Maven project

Download

You can download the full source code of this example here: SeleniumGrid