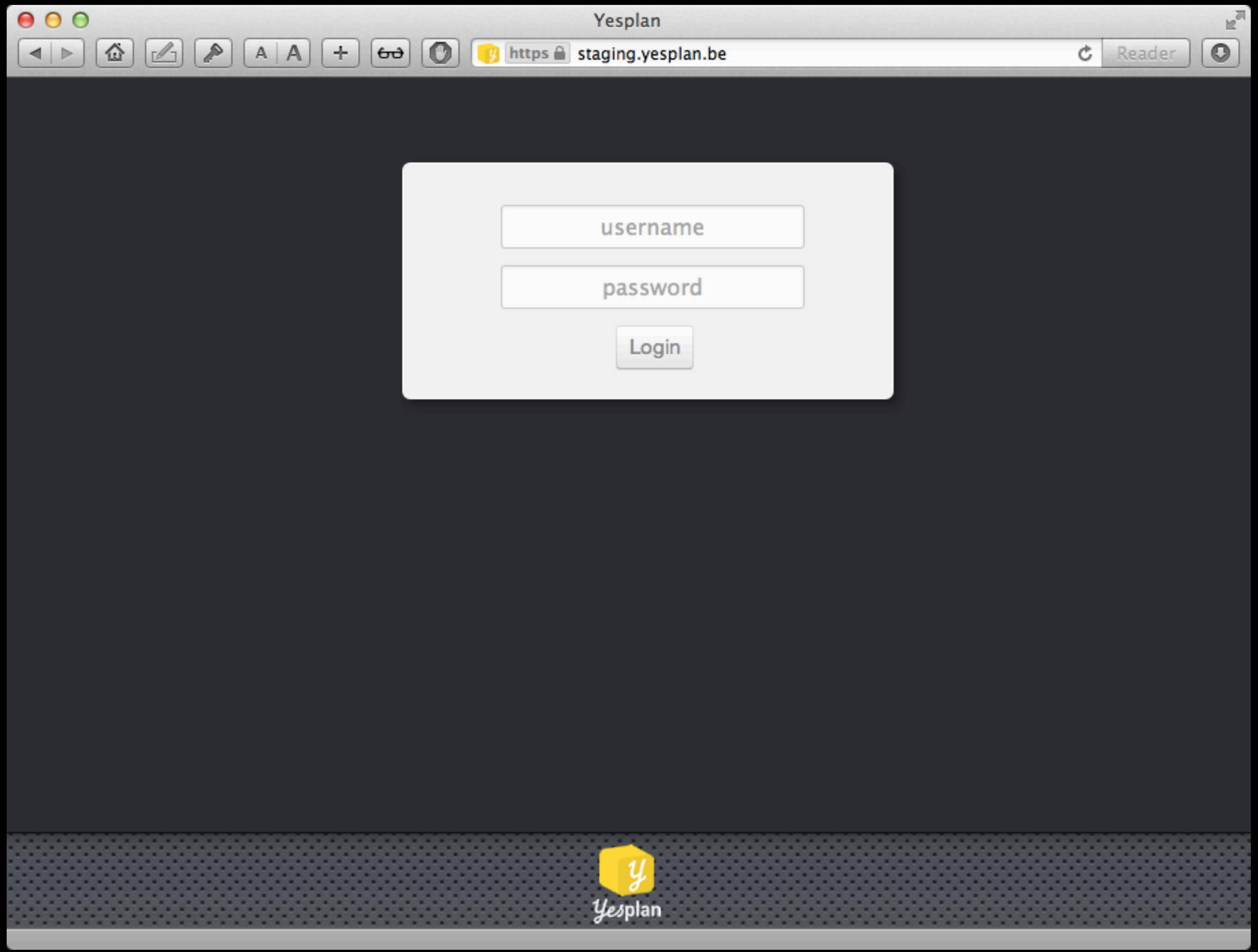
Beach Parasol

Don't get burned at the Seaside

Kris Gybels
2Rivers


Yesplan
https
staging.yesplan.be
Reader
username
password
Login
Yesplan

Croquet Edit

nextplan-dev.image

NPPresentationTestCase

Selenium – Web Browser Automation

www.seleniumhq.org

Reader

SeleniumHQ
Browser Automation

edit this page search selenium: Go

Projects | Download | Documentation | Support | About

What is Selenium?

Selenium automates browsers. That's it. What you do with that power is entirely up to you. Primarily it is for automating web applications for testing purposes, but is certainly not limited to just that. Boring web-based administration tasks can (and should!) also be automated as well.

Selenium has the support of some of the largest browser vendors who have taken (or are taking) steps to make Selenium a native part of their browser. It is also the core technology in countless other browser automation tools, APIs and frameworks.

Selenium is a suite of tools to automate web browsers across many platforms.

Selenium...

- runs in many browsers and operating systems
- can be controlled by many programming languages and testing frameworks.

Download Selenium

Donate to Selenium

with Google Checkout

Which part of Selenium is appropriate for me?

If you want to

- create quick bug reproduction scripts
- create scripts to aid in automation-aided exploratory testing

Then you want to use Selenium IDE; a Firefox add-on that will do simple record-and-playback of interactions with the browser

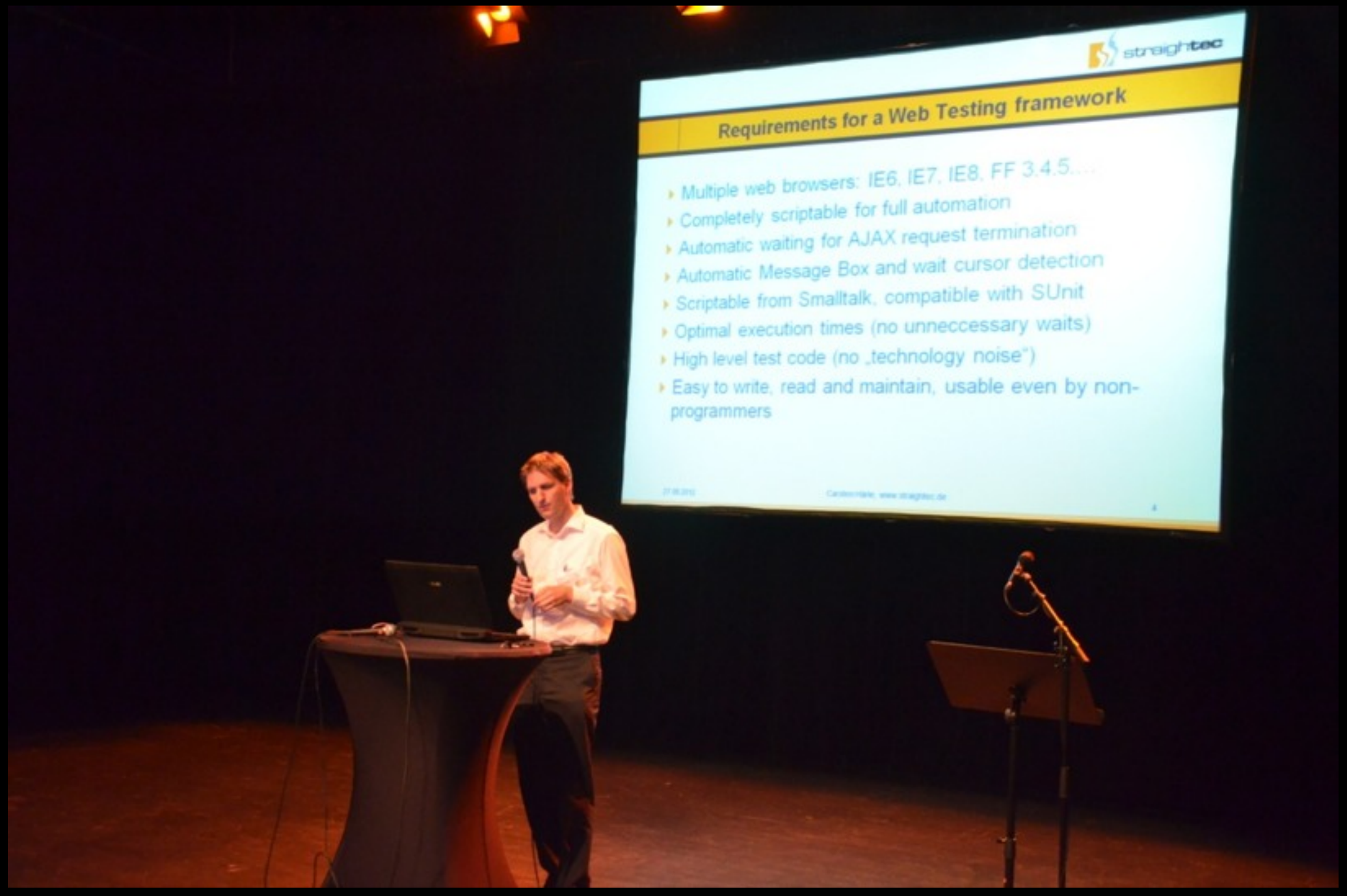
If you want to

- create robust, browser-based regression automation
- scale and distribute scripts across many environments

Then you want to use Selenium WebDriver; a collection of language specific bindings to drive a browser -- the way it is meant to be driven.

Selenium WebDriver is the successor of Selenium Remote Control which has been officially deprecated.

'http://www.seleniumhq.org/' openen



GUI-Testing Smalltalk-AJAX/SJAX web applications with Selenium

Carsten Härle @ ESUG 2012

Selenium Remote Control

Proxy server + Javascript injection

Selenium WebDriver

Native browser automation support (plugin, ...)

✔ 

✘ 

```
<a
  id="the_link"
  style="display: none"
  href="http://www.wikipedia.org">
  Invisible Link to Wikipedia</a>
```

Selenium Remote Control

```
selenium.click("id=the_link");
```

Selenium WebDriver

```
WebElement element = driver.findElement(By.id("the_link"));
element.click();
```

Selenium Remote Control

```
Selenium selenium = new DefaultSelenium("localhost", 4444,
  "*firefox", "http://localhost/");
selenium.start();
selenium.open("/page.html");
selenium.click("id=the_link");
```

Selenium WebDriver

More Object-Oriented API

```
WebDriver driver = new FirefoxDriver();
driver.get("http://localhost/page.html");
WebElement element = driver.findElement(By.id("the_link"));
element.click();
```

```
WebDriver driver = new FirefoxDriver();
                       ChromeDriver
                       IPhoneDriver
                       AndroidWebDriver
                       HtmlUnitDriver
                       ...
                       RemoteWebDriver
```

Network connection

selenium-server-standalone-2.31.0.jar

```
WebDriver remoteDriver = ...
```

driver := BPRemoteWebDriver new.

Command Reference

Command Summary

HTTP Method	Path	Summary
GET	/status	Query the server's current status.
POST	/session	Create a new session.
GET	/sessions	Returns a list of the currently active sessions.
GET	/session/:sessionId	Retrieve the capabilities of the specified session.
DELETE	/session/:sessionId	Delete the session.
POST	/session/:sessionId/timeouts	Configure the amount of time that a particular type of operation can execute for before they are aborted and a |Timeout| error is returned to the client.
POST	/session/:sessionId/timeouts/async_script	Set the amount of time, in milliseconds, that asynchronous scripts executed by `/session/:sessionId/execute_async` are permitted to run before they are aborted and a |Timeout| error is returned to the client.
POST	/session/:sessionId/timeouts/implicit_wait	Set the amount of time the driver should wait when searching for elements.
GET	/session/:sessionId/window_handle	Retrieve the current window handle.
GET	/session/:sessionId/window_handles	Retrieve the list of all window handles available to the session.
GET	/session/:sessionId/url	Retrieve the URL of the current page.
POST	/session/:sessionId/url	Navigate to a new URL.
POST	/session/:sessionId/forward	Navigate forwards in the browser history, if possible.
POST	/session/:sessionId/back	Navigate backwards in the browser history, if possible.

RemoteWebDriver

WebElement findElementById(String using)
List<WebElement> findElementsByXPath(String using)
String getPageSource()
Keyboard getKeyboard()

BPRemoteWebDriver

findElementByID: idString
findElementsByXPath: xpathString
getPageSource
getKeyboard

WebElement

String getAttribute(String name)
void click()

BPWebElement

getAttribute: nameString
click

Keyboard

void sendKeys(CharSequence... keysToSend)

BPKeyboard

sendKeys:

Actions

Actions doubleClick()
Actions doubleClick(WebElement onElement)

BPActions

doubleClick
doubleClick:

BPRemoteWebDriver

Parasol-Core
Parasol-Tests
Parasol-Pharo
Fuel

BPRemoteWebDriver
BPSelect

Instance | ? | Class

initialize-release
misc
private
private-http

getKeyboard
getPageSource
getScreenshotAsByteArra
getTitle

Browse | Hierarchy | Variables | Implementors | Inheritance | Senders | Versions | View

getPageSource

```
"
    http://selenium.googlecode.com/svn/trunk/docs/api/java/org/openqa/selenium/WebDriver.html#getPageSource()
"

^ self handleResponse: (self httpGet: (self baseSessionURL addPathSegment: 'source'))
    onSuccess: [ :result | result at: 'value' ]
```

WebDriver

selenium.googlecode.com/svn/trunk/docs/api/java/org/openqa/selenium/WebD

Reader

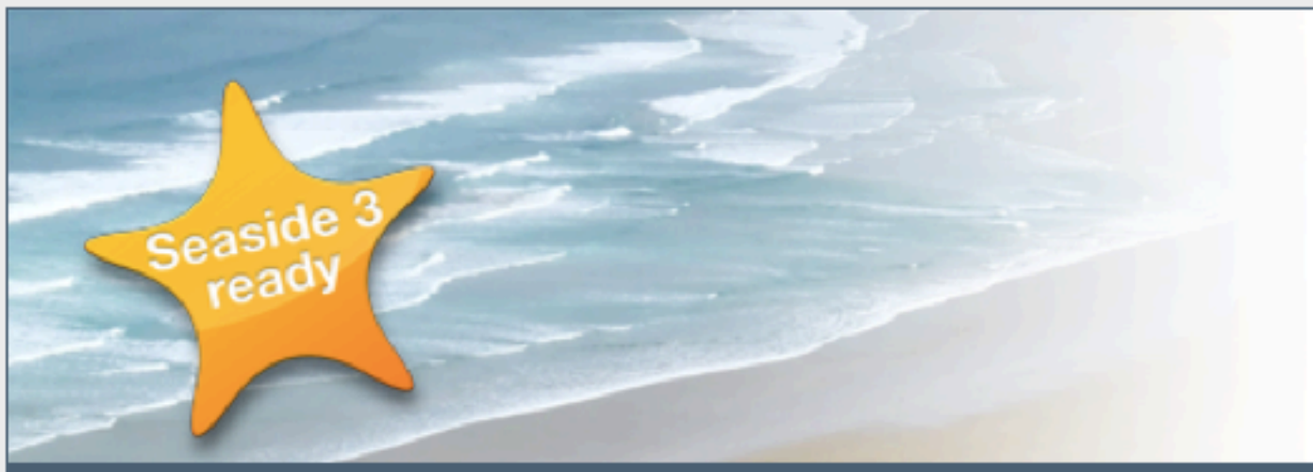
getPageSource

```
java.lang.String getPageSource()
```

Get the source of the last loaded page. If the page has been modified after loading (for example, by Javascript) there is no guarantee that the returned text is that of the modified page. Please consult the documentation of the particular driver being used to determine whether the returned text reflects the current state of the page or the text last sent by the web server. The page source returned is a representation of the underlying DOM: do not expect it to be formatted or escaped in the same way as the response sent from the web server. Think of it as an artist's impression.

Returns:

The source of the current page



HPI Hasso Plattner Institut

Seaside Tutorial

Software Architecture Group

11 - Testing Seaside Applications

What you are going to learn

- Introduction to Testing in Seaside
- Unit Testing in Smalltalk
- Introduction to and Installation of Selenium
- A Component Test Case
- Testing with a Separate Testing Environment
- Testing a Task
- Further Selenium Testing
- Improving your Code Quality with SwaLint and Slime

Beach Parasol

About Beach Parasol

Beach Parasol is a Pharo Smalltalk framework to automate web browsers. It's particularly useful to write automated tests for Seaside web applications. Its design and implementation are based on the Java Selenium WebDriver API.

Example

Here's a straightforward annotated example of using Parasol to automate a search for “Pharo” on Wikipedia:

```
"Open a web browser on the English-language Wikipedia home page."
driver := BPRemoteWebDriver new.
driver get: 'http://en.wikipedia.org/'.

"Click on the search box and type in 'Pharo' followed by a press of the Return key."
(driver findElementByID: 'searchInput') click.
driver getKeyboard sendKeys: ('Pharo' , (String with: BPKeys return)).

"Get the text of the article's first paragraph and show it on the transcript."
Transcript show: ((driver findElementByID: 'mw-content-text') findElementByTagName: 'p') getText.

"Tell the browser to quit."
driver quit.
```

Join!

Porting:

VisualWorks, *Stephan Eggermont*
Squeak, *Tobias Pape*
Your favorite Smalltalk, *You*

Completing

Demo

Explicit & Implicit Waiting

kris@yesplan.be

Yesplan

Let's make it happen