

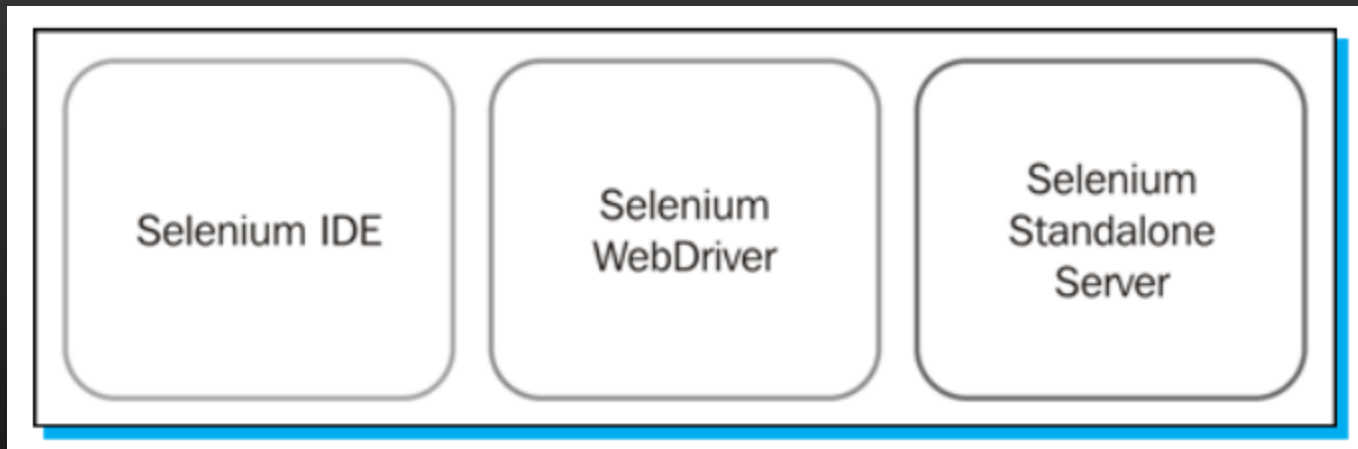
45426: Teste e Qualidade de Software

UAT & web testing with Selenium

Ilídio Oliveira

v2020-02-25

Selenium framework



Selenium IDE

Selenium IDE - the-internet*

Project: the-internet*

Executing ▾

login succeeded* http://the-internet.herokuapp.com

	Command	Target	Value
1	open	/	
2	click	linkText=Form Authentication	
3	type	id=username	tomsmith
4	type	id=password	SuperSecretPassword!
5	send keys	id=password	\${KEY_ENTER}
6	assert element present	id=flash	You logged into a secure area!\n

Runs: 1 Failures: 0

Log Reference

Command: assert element present

Target: id=flash

Value: id=flash

Description: css=#flash

Opens Window: xpath=//div[@id='flash']

id

css:finder

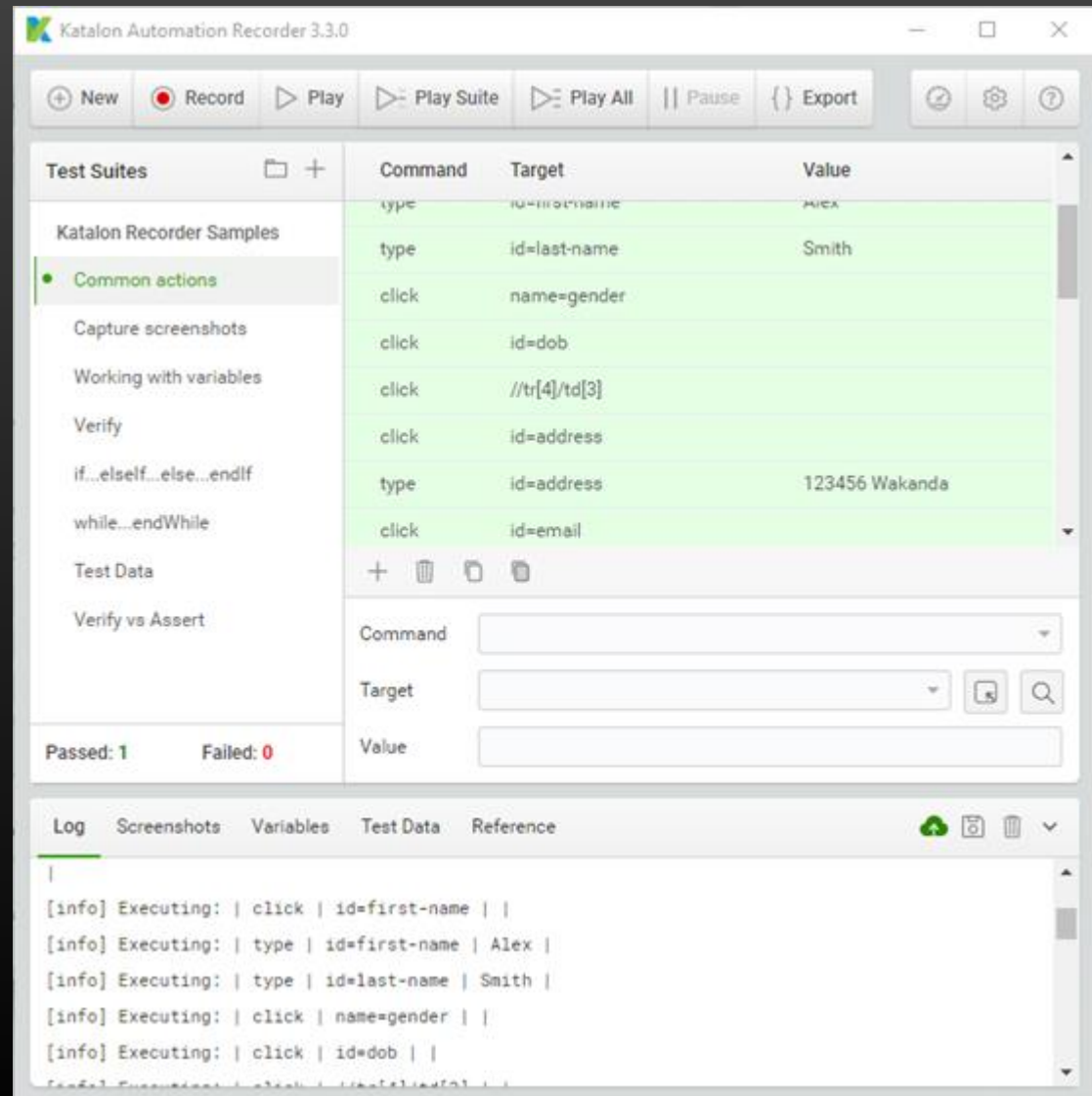
css

xpath:attributes

xpath:idRelative

xpath:position

Katalon Automation Recorders (browser add-on)

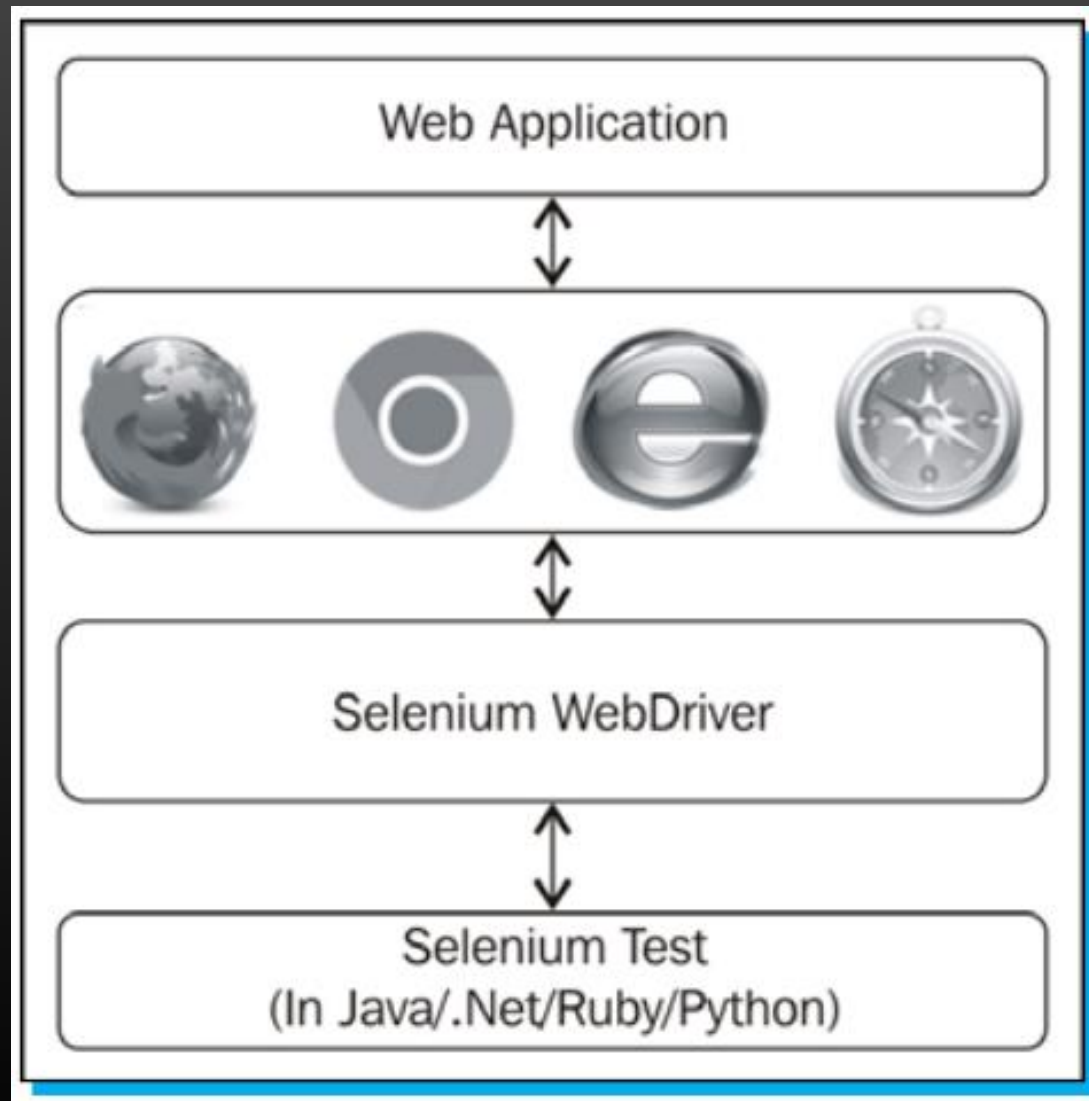


Command	Description
open	Opens a specified URL in the browser.
assertTitle, VerifyTitle	Returns the current page title and compares it with the specified title
assertElementPresent, verifyElementPresent	Verify / Asserts the presence of an element on a web page.
assertTextPresent, verifyTextPresent	Verify / Asserts the presence of a text within the web page.
type, typeKeys, sendKeys	Enters a value (String) in the specified web element.
click, clickAt, clickAndWait	Clicks on a specified web element within a web page.
waitForPageToLoad	Sleeps the execution and waits until the page is loaded completely.
waitForElement Present	Sleeps the execution and waits until the specified element is present
chooseOkOnNextConfirmation, chooseCancelOnNextConfirmation	Click on "OK" or "Cancel" button when next confirmation box appears.

Selenium framework



Selenium WebDriver



Test code can be in different languages.

Java

```
WebDriver driver = new FirefoxDriver();  
driver.get("http://localhost:8080");  
  
driver.findElement(By.name("email"))  
    .sendKeys("jane.smith@acme.com");
```

Ruby

```
driver = Selenium::WebDriver.for :firefox  
driver.navigate.to "http://localhost:8080"  
  
element = driver.find_element(:name, 'email')  
element.send_keys "jane.smith@acme.com"
```

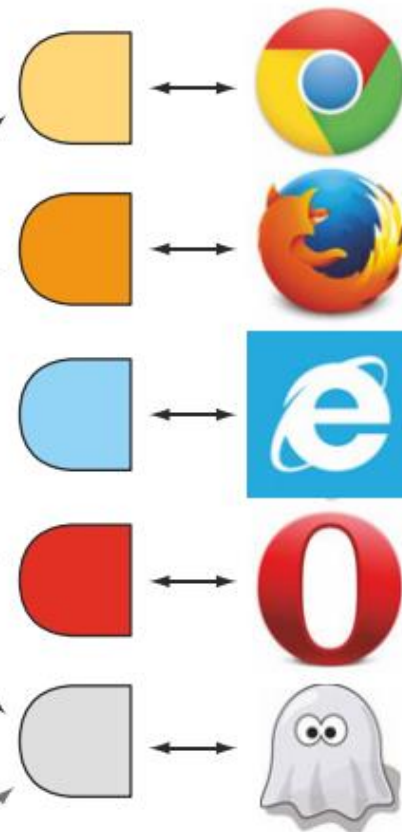
Python

```
driver = webdriver.Firefox()  
driver.get("http://localhost:8080")  
  
element = driver.find_element_by_name("email")  
element.send_keys("jane.smith@acme.com")
```

C#

```
IWebDriver driver = new FirefoxDriver();  
driver.Navigate().GoToUrl("http://localhost:8080");  
  
IWebElement element = new driver.FindElement(By.name("email"));  
element.SendKeys("jane@acme.com");
```

WebDriver presents a common API for all browsers.



Each supported browser has a specific driver that implements the common API.

Interfaces

Alert
Capabilities
ContextAware
HasCapabilities
JavascriptExecutor
OutputType
Rotatable
SearchContext
TakesScreenshot
WebDriver
WebDriver.ImeHandler
WebDriver.Navigation
WebDriver.Options
WebDriver.TargetLocator
WebDriver.Timeouts
WebDriver.Window
WebElement

Classes

By
By.ByClassName
By.ByCssSelector
By.ById
By.ByLinkText
By.ByName
By.ByPartialLinkText
By.ByTagName
By.ByXPath
Cookie
Cookie.Builder

static interface

WebDriver.Window

Method Summary

All Methods	Instance Methods	Abstract Methods
Modifier and Type		Method and Description
void		close() Close the current window, quitting the browser if it's the last window current
WebElement		findElement(By by) Find the first WebElement using the given method.
java.util.List<WebElement>		findElements(By by) Find all elements within the current page using the given mechanism.
void		get(java.lang.String url) Load a new web page in the current browser window.
java.lang.String		getCurrentUrl() Get a string representing the current URL that the browser is looking at.
java.lang.String		getPageSource() Get the source of the last loaded page.
java.lang.String		getTitle() The title of the current page.

<https://seleniumhq.github.io/selenium/docs/api/java/org/openqa/selenium/WebDriver.html>

```

@Test
public void testSearch() throws Exception {
    driver.get(baseUrl + "/");
    driver.findElement(By.id("search")).clear();
    driver.findElement(By.id("search")).sendKeys("Nokia");
    driver.findElement(By.cssSelector("button.button")).click();
    assertTrue(isElementPresent(By.linkText("Nokia 2610 Phone")));
    driver.findElement(By.linkText("Nokia 2610 Phone")).click();
    // Warning: assertTextPresent may require manual changes
    assertTrue(driver.findElement(By.cssSelector("BODY")).
getText().matches("^([\\s\\S]*Nokia 2610 Phone[\\s\\S]*$"));
}

```

Locating elements in the page

Strategy	Description	Selenium IDE	Selenium WebDriver (Java)
identifier=<id>	Locates the first element that has the specified value for the @id attribute. IDs are uniquely assigned by developers for key elements on a page. This strategy is considered the most reliable and faster.	id=search	<code>driver.findElement(By.id("search"))</code>
name=<name>	Locates the first element with the specified value for the @name attribute.	name=search	<code>driver.findElement(By.name("search"))</code>
link=<text Pattern>	Locates the link (anchor) element that matches the specified pattern in link text.	link=Nokia 2610 Phone	<code>driver.findElement(By.linkText("Checkout"))</code> or <code>driver.findElement(By.partialLinkText("Inbox"))</code> (You can specify a partial link text instead of complete link text. This might be useful when links partially contain dynamic values.)
css=<cssSelector Syntax>	Locates elements using CSS selectors (refer to http://goo.gl/a9v7T and http://	css=input.search (this will locate input element whose @class	<code>driver.findElement(By.cssSelector("input.search"))</code>

Purpose	Selenium IDE	Selenium WebDriver (Java)
Click on an element	click	element.click() Acting in the page
Type a text	type	element.sendKeys()
Check a checkbox/ radio button	check	If(!element.isSelected()) { element.click() }
Uncheck Checkbox/ Radio Button	uncheck	If(element.isSelected()) { element.click() }
Select item(s) in a list or drop-down list	select addSelection (for multi select)	element. selectByVisibleText("Option"); or element. selectByVisibleValue("Option"); or element.selectByVisibleIndex(1);
Remove selection from a list or a drop- down list	removeSelection removeAllSelection	element.deselectByVisibleText("O ption"); or element.deselectByVisibleValue("O ption"); or element. deselectByVisibleIndex(1);

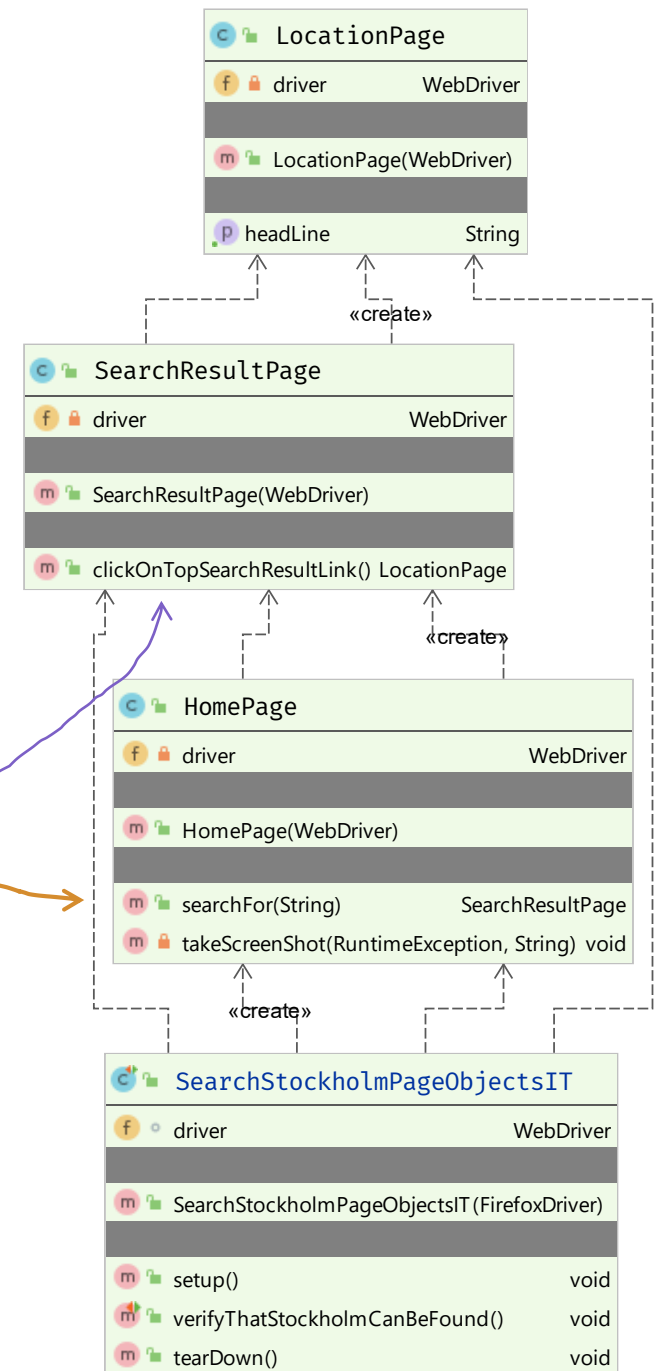
Page Object Model

Create Classes to map the Pages

Encapsulate WebDriver low-level interaction in the target class method

```
@Test
public void verifyThatStockholmCanBeFound() {
    HomePage home = new HomePage(driver);
    SearchResultPage searchResult = home.searchFor( location: "Stockholm");

    LocationPage stockholm = searchResult.clickOnTopSearchResultLink();
    String actual = stockholm.getHeadLine();
    assertTrue(actual.contains("Stockholm"));
}
```



Suggested links

Book: “Instant Selenium
Testing Tools Starter”

Comprehensive [tutorial](#)