

keytorc

Selenium 2 Cheat Sheet

Navigation

Şu iki metod aynı görevi yerine getirir, Her ikisi de belli bir url'yi açar:

- <WebDriver>.get(<Url>)
- <WebDriver>.navigate.to(<Url>)

Bulunulan sayfayı yenilemek için şu metod kullanılabilir:

- <WebDriver>.navigate.refresh()

<WebDriver> ile kastedilen,

```
private WebDriver driver = new FirefoxDriver();
```

örneğindeki “driver” nesnesidir, yani bu örnek için:

- driver.get(“www.google.com.tr”); olacaktır, ya da
- driver.navigate.to(“www.google.com.tr”);
- driver.navigate.refresh();

XPath & CSSSelector:

“div.foo” tag adı div olan ve “class” attribute’ünde “foo” geçen, sayfadaki bütün web elemanlarını bulur, xpath ile:

```
“//div[@class='foo']”
```

“div.foo.bar[title^=“Help”]” tag adı div olan ve “class” attribute’ünde “foo” ve “bar”geçen ve title’ı “Help” ile başlayanları bulur.

“div.foo.bar[title\$=“Help”]” tag adı div olan ve “class” attribute’ünde “foo” ve “bar”geçen ve title’ı “Help” ile bitenleri bulur.

“div.foo.bar[title*=“Help”]” tag adı div olan ve “class” attribute’ünde “foo” ve “bar”geçen ve title’ı “Help” içerenleri bulur (title’in içinde herhangi bir yerde “Help” kelimesi geçiyorsa), xpath ile:

```
“//div[@class='foo bar'] [contains(@title, 'Help')]”
```

“ul#navigation” id’si “navigation” olan, sayfadaki bütün “ul” web elemanlarını bulur, xpath ile:

```
“//ul[@id='navigation']”
```

“a[href^=“http://”]” href’i “http://” ile başlayan, sayfadaki bütün “a” web elemanlarını bulur.

“input[type=“submit”]” type’ı “submit” olan, sayfadaki bütün “input” web elemanlarını bulur, xpath ile:

```
“//input[@type='submit']”
```

Şu web elemanını içerdiği metne göre bulmak istersek: Mail

```
<WebDriver>.findElement(By.cssSelector(“a:contains(‘Mail’)”));
```

diyebiliriz, xpath ile ise şöyle bulunur:

```
<WebDriver>.findElement(By.xpath(“//a[text()='Mail']”));
```

Interrogation (Inspection)

Mozilla Firefox browser’ı için “Firebug” addon’unun eklenmesi gerekmekte, “Firebug” kullanılması web elemanlarının belirlenmesini kolaylaştırır. Ayrıca “Firepath” tabına xpath ya da cssSelector yazıldığında ilgili elemanı gösterir.



Bulunulan sayfanın başlığını (title) döndürür:

```
<WebDriver>.getTitle()
driver.getTitle()
```

Bulunulan sayfanın url’sini döndürür:

```
<WebDriver>.getCurrentUrl()
diver.getCurrentUrl()
```

Bulunulan sayfanın sayfa kaynağını döndürür:

```
<WebDriver>.getPageSource()
driver.getPageSource()
```

<WebElement> ile kastedilen sayfadaki <div>, <input>, <label>, <select>, <table>, <tbody>, <tr>, <td>, , gibi elemanlardan herhangi biridir, örneğin: WebElement table =

```
<WebDriver>.findElement(By.id(“containerTable”));
```

İle bir tabloyu alabiliriz, bu tablodaki bütün satırları(tr) almak için:

```
List<WebElement> allRows =
table.findElements(By.tagName(“tr”));
İle tablodaki bütün satırlar “allRows” Listine kopyalanır, “List<WebElement>” ile List içindeki bütün öğelerin tipinin “WebElement” olduğunu belirtiyoruz, farklı tipte bir öge bu List’e kopyalanamaz.
```

Bir sayfa elemanını bulmak için id, xpath, cssSelector, className veya tagName kullanılabilir:

- <WebElement>.findElement(By.id(“”))
- <WebElement>.findElement(By.xpath(“”))
- <WebElement>.findElement(By.cssSelector(“”))
- <WebElement>.findElement(By.className(“”))
- <WebElement>.findElement(By.tagName(“”))

Bir web elemanının içerdiği metni almak için:

```
<WebElement>.getText()
```

Bir web elemanının belli bir attribute değerini almak için:

- <WebElement>.getAttribute()
- <WebElement>.getAttribute(“class”)
- <WebElement>.getAttribute(“value”)

Bir web elemanına ait tag adını almak için:

```
<WebElement>.getTagName()
```

Web elemanının durumunu control etmek için:

- <WebElement>.isDisplayed()
- <WebElement>.isEnabled()
- <WebElement>.isSelected()

“findElement” ile aradığını bulamazsa, önce timeout süresi kadar bekler, sonra exception fırlatır (Unable to locate element), bunun yerine “findElements(By...()).size()” kullanılabilir, eğer size()=0 ise, bu aranan elemanın bulunamadığı anlamına gelir, böylece timeout süresi kadar beklemeye gerek kalmaz.

Manipulation (Interaction)

```
<WebElement>.click()
```

```
<WebElement>.clear()
```

Bazı durumlarda seçim yapma işlemini tamamlamak için sendKeys’ten sonra “sendKeys(Keys.ENTER)” , göndermek gerekebilir.

```
<WebElement>.sendKeys()
```

```
<WebElement>.sendKeys(Keys.ENTER)
```

Eğer bir combobox’tan seçim yapmakta zorlanılıyor ise “sendKeys(Keys.ARROWDOWN)” metodu ile bir sonraki seçeneğe ulaşılabilir.

```
<WebElement>.sendKeys(Keys.ARROWDOWN)
```

Alternatif bir yöntem olarak combobox’a, istenilen seçeneğin ilk harfi gönderilebilir, <WebElement>.sendKeys(“A”) gibi, bu durumda comboboxtan “A” ile başlayan ilk seçenek seçilecektir.

Kodun hem Mac’te hem de Windows’ta çalışması isteniyorsa

```
<Select>.selectByVisibleText() ve
<Select>.selectByIndex() yerine
<Select>.getOptions().get(index) kullanılmalı.
```

Alternatif bir tıklama yöntemi olarak şu kullanılabilir:

```
n e w
Actions(<WebDriver>).moveToElement( <WebElement> )
.click().build().perform()
```

Checkbox’lar tıklanırken Windows’ta çalışan kod Mac’te çalışmayabilir, buna dikkat etmek gerek, yazılan kodun, checkbox’ın tam olarak içini aldığına dikkat etmek gerekli (Mac için).

Synchronization

Senkronizasyon açısından her click(), sendKeys() veya select işleminden hemen sonra bekletme yapan kod eklenmelidir, bu bir WebDriverWait() nesnesi de olabilir veya AJAX’ın tamamlanmasını bekleyen bir kod da olabilir.

• Explicit Wait örneği:

```
WebDriverWait wait= new WebDriverWait(driver,20 );
wait.until( ExpectedConditions.visibilityOfElementLocat
ed (By.xpath(“//*[@id='someld']”)));
```

• Implicit Wait örneği:

```
driver.manage().timeouts().implicitlyWait(30,
TimeUnit.SECONDS);
```

Domain

• Cookies

```
<WebDriver>.manage().deleteAllCookies();
sadece bulunduğunuz domaindeki cookie’leri siler.
```

```
<WebDriver> ile kastedilen,
```

```
private WebDriver driver = new FirefoxDriver();
```

örneğindeki “driver” nesnesidir, yani bu örnek için:

```
driver.manage().deleteAllCookies(); olacaktır.
```

• Windows

Bulunulan pencerenin String tipinde handle’ını almak için:

```
<WebDriver>.getWindowHandle()
```

<WebDriver>.getWindowHandles() , açık olan bütün pencerelerin String tipinde handle’larını içeren bir Set döndürür.

```
<WebDriver>.switchTo().window(<WindowHandle>);
Bütün window-handle’lar alınıp bir döngü içinde birer birer sırayla hepsine focuslanırsa(switchTo), en son açılan pencere focuslanılmış(switchTo) olur.
```

```
//Store the current window handle
String windowHandleBefore =
driver.getWindowHandle();
//Switch to new window opened
Set<string> windowHandles =
driver.getWindowHandles();
for(String windowHandle : windowHandles){
driver.switchTo().window(windowHandle);}
```

```
<WebDriver>.manage().window().maximize();
bazı testcase’ler koşулurken eğer pencere tam boyutlu değilse fail olabiliyor , bu nedenle testin başında pencerenin tam boyutlu hale getirilmesinde fayda var.
```

• Frames

Eğer yeni bir frame açıldıysa bu frame’e focuslanmak için şunlar kullanılabilir:

- > <WebDriver>.switchTo().frame(<name_or_id>)
- > <WebDriver>.switchTo().frame(<index>)
- > <WebDriver>.switchTo().frame(<WebElement>)

Yeni açılan frame’den önceki sayfaya dönmek için de şu kullanılabilir:

```
<WebDriver>.switchTo().defaultContent() , eğer bu çalışmazsa,
driver.switchTo().window(<WindowName>) kullanılabilir.
```

• JavascriptExecutor

Java kodu içinde Javascript çalıştırmak mümkün, bunun için şu kullanılabilir:

```
((JavascriptExecutor)<WebDriver>).executeScript(“alert
('hello world');”)
```

Bu örnekte, “hello world” yazısı içeren bir alert görüntülenir.

Örneğin, şöyle bir sayfa elemanını görünür yapmak için, <select id=“periodId” name=“period” style=“display: none;”>

Şöyle bir kod kullanılabilir:

```
JavascriptExecutor executor =
(JavascriptExecutor)<WebDriver>;
executor.executeScript(“document.getElementById(‘peri
odId’).style.display=‘block;”);
```

• Alerts

Açılan Alert penceresine focuslanmak için şu kullanılabilir:

```
<WebDriver>.switchTo().alert();
```

<Alert> için kullanılabilecek bazı metotlar şunlar:

- > <Alert>.accept()
- > <Alert>.dismiss()
- > <Alert>.getText()

Se



keytorc

www.keytorc.com
blogs.keytorc.com

Telefon : +90 212 290 76 60
E-posta : info@keytorc.com