



www.techglobalschool.com 2860 S River

Rd Suite 350, Des Plaines IL 60018

Selenium Mülakat Soruları	4
Selenium nedir?	4
Selenium Otomasyon Aracını neden tercih ediyorsunuz? / Seleniumun avantajları nelerdir?.....	4
Seleniumun dezavantajları nelerdir? /Selenium'un sınırlamaları nelerdir?	4
Selenium hangi test türlerini destekler?	4
Selenium tarafından desteklenen bazı açık kaynaklı çerçeveler nelerdir?.....	4
Selenium Bileşenleri ve farklılıkları nelerdir?	5
Selen nedir?	5
Selenium'daki farklı konumlandırıcı türleri nelerdir?.....	5
XPath, Mutlak XPath ve Göreli XPath nedir?.....	6
XPath'de tek eğik çizgi (/) ve çift eğik çizgi (//) nedir?	6
XPath kullanarak ana öğeye nasıl geçilir?	6
Bir sayfadaki tüm bağlantılar nasıl bulunur?	6
XPath ve CSS Selector nedir, farklılıkları, avantajları ve dezavantajları nelerdir?	6
Dinamik ve statik web öğeleri nelerdir? Dinamik web öğeleri nasıl işlenir?	7
Selenium WebDriver kullanarak bir tarayıcıya nasıl öğle yemeği verilir?	8
Selenium kullanarak bir web sayfasını nasıl büyütebilirim?	9
driver.close() ve driver.quit() yöntemleri arasındaki fark nedir?	9
get() ve navigasyon().to() yöntemleri arasındaki fark nedir?	9
Selenium kullanarak tarayıcıyı yenilemenin farklı yolları nelerdir?.....	9
findElement() ve findElements() yöntemleri arasındaki fark nedir?	9
getText() ve getAttribute() yöntemleri arasındaki fark nedir?	10
Selenium kullanarak açılır menüler nasıl işlenir?	10
Selenium'da Select sınıfı nedir ve neden kullanıyoruz?	11
iFrame nedir ve Selenium kullanarak iFrame'lerle nasıl başa çıkılır?.....	11
Selenium kullanarak bir web sayfasındaki toplam çerçeve sayısını nasıl bulabilirsiniz?	11
Selenium kullanarak birden çok pencere/sekme nasıl işlenir?.....	12
Uyarılar veya açılır pencereler nasıl ele alınır?.....	12
Kaç tane uyarı yöntemi biliyorsunuz?	13

Actions sınıfı nedir ve neden kullanıyoruz?	13
Eylemler sınıfı nasıl kullanılır?.....	13
Actions sınıfını kullanarak bir web ögesine nasıl çift tıklanacağını söyleyebilir misiniz?	14
Selenium WebDriver'da isDisplayed(), isEnabled() ve isSelected() yöntemlerinin farkları nelerdir?.....	14
Anahtar değiştiriciler nelerdir?	14
Selenium'da karşılaştığınız bazı istisnaları açıkla mısınız?	15
Selenium'da bulunan bekleme türleri nelerdir? Örtülü, açık ve akıcı bekleyişleri açıklayın?.....	15
Selenium kullanarak nasıl aşağı veya yukarı kaydırılır?.....	16
Selenium'da neden senkronizasyona ihtiyacımız var?	17
Selenium WebDriver'ın yetenekleri nelerdir?.....	17
WebDriver'da bulunan farklı sürücü türleri nelerdir?.....	17
WebDriver kullanarak tarayıcı nasıl başlatılır?	17
Mobil Test için desteklenen farklı sürücüler nelerdir?.....	17
Hangi WebDriver uygulaması en hızlıdır?	18
HtmlUnitDriver nedir?	18
Başsız tarayıcı testinin avantajı nedir?.....	18
Selenium WebDriver kullanılarak mobil uygulamalar test edilebilir mi?	18
Captcha, Selenium ile otomatikleştirilebilir mi?	18
Selenium, Windows tabanlı açılır pencereleri işleyebilir mi?.....	18
Windows tabanlı açılır pencereleri nasıl ele alabiliriz?.....	18
Selenium kullanarak bir web sayfasını yenilemenin farklı yöntemleri nelerdir?	19
Selenium kullanarak bir web ögesinin metni nasıl iddia edilir?	19
switchTo() komutu ne işe yarar?	19
Selenium kullanarak çerçeveler arasında nasıl geçiş yapılır?	19
Selenium kullanarak bir web sayfasının başlığı nasıl belirlenir?	20
Selenium kullanarak bir metin kutusuna nasıl yazılır?	20
sendKeys() kullanmadan metin kutusuna yazmanın bir yolu var mı?	20
Selenium'da linkText kullanarak bir hiper bağlantıya nasıl tıklanır?	20
Selenium kullanarak çift tıklama nasıl yapılır?	20
Selenium kullanarak bir web ögesinin üzerine fareyle nasıl gelinir?	21
Selenium kullanarak web sayfalarındaki tablolar nasıl işlenir?.....	21
Selenium kullanarak dosya indirme/yükleme nasıl yapılır?	22
Assert ve doğrulama komutları arasındaki fark nedir?	22
Sert iddia ve yumuşak iddia arasındaki fark nedir?	22
İstenen yetenek nedir? Neden önemli?	23

Nesne deposu nedir?	23
Nesne deposunda (sayfa sınıfları) ne saklıyorsunuz?	23
Selenium Senaryo Tabanlı Mülakat Soruları	24
Bir NoSuchElementException aldığınızı varsayalım. Sorunu gidermek için ne yapardınız?.....	24
Bir ElementClickInterceptedException aldığınızı varsayalım. Sen ne yapardın? Sorunu nasıl giderirsiniz?	
Bir sayfaya gittiğinizi ve bir açılır pencere yüklediğinizi varsayalım. Ancak bu bazen olur ve çoğu zaman açılır pencere görünmez. Bu durumu nasıl ele alırdınız?	24
Smoke ve Regression gibi test takımlarınızı Chrome, Firefox, Edge ve daha fazlası gibi farklı tarayıcılarda çalıştırmanız gerekeceğini varsayalım. Bu durumu nasıl ele alırdınız?	25

Selenium Mülakat Soruları

Selenium nedir?

- Selenium bir **yazılım araçları paketi** web tabanlı uygulamaları otomatikleştirmek için kullanılır
- O bir **açık kaynak** herhangi bir lisans maliyeti gerektirmediği anlamına gelen araç takımı

Selenium Otomasyon Aracını neden tercih ediyorsunuz? / Selenium'un avantajları nelerdir?

- O bir **açık kaynaklı araçlar paketi** yüzünden **Bedava** herhangi bir lisans maliyeti olmadan kullanmak için
- **BT çeşitli İşletim ortamlarını destekler** Windows, Linux, Macintosh vb.
- **BT gibi birden çok dili destekler** Java, Ruby, Python, C#, Perl vb.
- **BT çeşitli tarayıcıları destekler** Mozilla Firefox, Edge, IE, Google Chrome, Safari, Opera vb.
- Test çerçevelerinde öğrenmesi ve kullanması kolaydır

Seleniumun dezavantajları nelerdir? / Selenium'un sınırlamaları nelerdir?

- Selenium yalnızca web tabanlı uygulamaları destekler, masaüstü ve mobil uygulamaları desteklemez
- Grafikleri, captcha'ları, barkodları, şekilleri, görüntüleri, PDF'leri otomatikleştirmek için kullanılamaz
- Raporlar doğrudan Selenium aracılığıyla oluşturulamaz ve rapor oluşturma etkinliği için TestNG veya JUnit gibi üçüncü taraf araçlara ihtiyaç duyar.
- Selenium'un kaynak koduyla senkronize edilmesi zordur ve otomasyon test cihazlarının uyku(), örtük bekleme, açık bekleme vb. gibi bekleme yöntemlerini kullanması gerekir.
- Yerleşik raporlama aracı yoktur, rapor oluşturma etkinliği için üçüncü taraf araçlara ihtiyaç duyar
- Dosya yüklemeyi desteklemiyor (yolu tanımlarsak yapılabilir)

Selenium hangi test türlerini destekler?

- Sigara içmek
- regresyon
- Akıl sağlığı
- Fonksiyonel testler (pozitif/negatif UI testleri)
- Entegrasyon
- Baştan sona
- Çapraz tarayıcı

Selenium tarafından desteklenen bazı açık kaynaklı çerçeveler nelerdir?

- **Java tabanlı JUnit ve TestNG birim test çerçeveleri**
ÖJava tabanlı Selenium çerçeveleri, BDD deneyimi sağlamak için Salatalık ile entegre edilebilir
 - **C# tabanlı NUnit birim test çerçevesi**
ÖC# tabanlı Selenium çerçeveleri, BDD deneyimi sağlamak için SpecFlow ile entegre edilebilir Sadece
- NOT:** bunlar değil, Selenium diğer birçok dili ve bunlar kullanılarak oluşturulan çerçeveleri desteklemektedir.

Selenium Bileşenleri ve farklılıkları nelerdir?

1. Selenium Web Sürücüsü

- Ö Selenium WebDriver, web uygulamaları için testleri otomatikleştirmek için kullanılan bir araçtır ve herhangi bir arayüz olmadan doğrudan tarayıcılarla etkileşime girer.
- Ö Firefox, Chrome, IE, Safari vb. gibi farklı tarayıcılara karşı test uygulamalarının yürütülmesine izin verir.
- Ö Birden çok çerçeveyi, birden çok tarayıcı penceresini, açılır pencereleri, uyarıları vb. işlemek için kullanılır.
- Ö Sürükle ve bırak gibi karmaşık sayfa gezinme ve gelişmiş kullanıcı işlemleri yapmamızı sağlar.

2. Selenium Izgarası

- Ö Selenium Grid, komut dosyalarını paralel olarak (eşzamanlı olarak) yürütmemize izin verir. Bu, test sürecini hızlandırmak ve yürütme için zaman kazanmak için büyük bir avantajdır.

3. Selenium IDE:

- Ö Selenium IDE, Selenium testleri için entegre bir geliştirme ortamıdır.
- Ö Bir Firefox eklentisi olarak uygulanır ve testi Firefox'ta kaydetmemize, düzenlememize ve yeniden oynatmamıza olanak tanır Selenium
- Ö IDE, testleri (raporu) HTML, Java, Ruby komut dosyaları veya başka herhangi bir biçimde kaydetmemize olanak tanır.

4. Selenium RC

- Ö Selenium Remote Control (RC), bir web uygulaması için Java, C# vb. herhangi bir programlama dilinde otomatik testler yazmayı mümkün kılar.
- Ö Selenium RC mimarisi, Selenium WebDriver'a göre daha karmaşıktır ve kullanıcılar için kullanımı zordur. Ayrıca Selenium WebDriver, Selenium RC'den daha hızlıdır ve Selenium WebDriver'ın Selenium RC yerine yaygın olarak kullanılmasının bilinen nedenleri bunlardır.

Selen nedir?

- **selen** Selenium IDE'de test komut dosyaları yazmak için kullanılan dildir

Selenium'daki farklı konumlandırıcı türleri nelerdir?

- Konumlandırıcılar, belirli web öğelerini bulmak için kullanılan Selenium yöntemleridir.
- Selenium bulucular, Selenium jar dosyalarından/kütüphanesinden gelen yöntemler kümesidir.
- 8 farklı konumlandırıcı geliyorsınıza göre
- Selenium'daki çeşitli konumlandırıcılar aşağıdaki gibidir: Ö
 - Bağlantı Metni ->yalnızca bağlantılarla kullanılabilir, bu nedenle bağlantı etiketleri **Kısmi**
 - Ö Bağlantı Metni ->yalnızca bağlantılarla kullanılabilir, bu nedenle bağlantı etiketleri **Kimlik** ->
 - Ö her zaman benzersizdir ve varsa en çok tercih edilen konumlandırıcıdır **İsim**
 - Ö
 - Ö Etiket adı
 - Ö Sınıf adı
 - Ö XPath ->kimlikten sonra en çok kullanılan konumlandırıcılardan biri
 - Ö CSS Seçici ->id'den sonra en çok kullanılan konum belirleyicilerden biri. XPath'den daha hızlıdır.
- **ad, bağlantı metni, sınıf adı, kimlik, kısmi bağlantı metni, etiket adı**konum belirleyiciler verilen öznitelik adını arar ve eşleşen öznitelik değerini bulur
- **CSS Seçici ve XPath kendi sözdizimlerine sahiptir**ve her ikisinin de birbirine göre avantajları var
- **XPath, ebeveyn den çocuğa ve çocuktan ebeveyne atlamamızı sağlar**.böylece belirli bir web öğesini benzersiz bir şekilde bulabiliriz ve web öğelerini bulmak için daha fazla yöntemi vardır.**CSS Seçici, XPath'den daha hızlıdır**

XPath, Mutlak XPath ve Göreli XPath nedir?

- **XPath:**

- Ö XML belgelerindeki öğeleri bulmanın ve işlemenin bir yolunu sağlayan bir sözdizimi veya dildir.
- Ö HTML DOM yapısını kullanarak bir web sayfasındaki herhangi bir öğeyi benzersiz bir şekilde tanımlamak için Selenium'da kullanılır.

- **Mutlak XPath:**

- Ö HTML DOM yapısında açıklandığı gibi öğenin tam yolunu vererek doğrudan XPath'i bulmak için kullanılır.
- Ö Öğenin yolunda herhangi bir değişiklik yapılırsa, XPath başarısız olur
- Ö Tek bir eğik çizgi (/) ile başlar, kök düğümden başlayarak öğeyi seçebileceğiniz anlamına gelir.
- Ö **ÖRNEK:** /html/body/div/span/a

- **Görelî XPath:**

- Ö Görelî XPath, eşleşen öğeyi HTML DOM yapısındaki herhangi bir yerden aramaya başlar
- Ö Çift eğik çizgiyle (//) başlar, bu da öğeyi sayfanın herhangi bir yerinde arayabileceği anlamına gelir.
- Ö **ÖRNEK:** //input[@name='q']

XPath'de tek eğik çizgi (/) ve çift eğik çizgi (//) nedir?

- **Tek eğik çizgi** içinde kullanılır **Mutlak XPath** ve kökten başlayarak elemanı aramaya başlar düğüm
- **Çift eğik çizgi** içinde kullanılır **Görelî XPath** ve herhangi bir yerde eşleşen öğeyi aramaya başlar XML belgesi

XPath kullanarak ana öğeye nasıl geçilir?

- **Kullanabiliriz .. (nokta nokta)** ana öğeye geçmek için XPath'deki ifade

Bir sayfadaki tüm bağlantılar nasıl bulunur?

- Bunları tagName bulucu kullanarak bulabilir ve bir Listede saklayabiliriz.
List<WebElement> list = driver.findElements(By.tagName("a"));
- **HATIRLAMAK:** Bağlantılar <a> etiketi kullanılarak oluşturulur.

XPath ve CSS Selector nedir, farklılıkları, avantajları ve dezavantajları nelerdir?

- **XPath** XML yolu anlamına gelir ve bir XML belgesindeki düğümleri bulmak için kullanılan dildir.
- Web öğelerini tanımlamak için en esnek konumlandırıcıdır.
- **2 tür XPath vardır:** Ö **Mutlak XPath:** /html/body/form/div/span/span **Görelî XPath:** //tagName[@AttributeName='değer']

- **XPath birçok yöntem sağlar** öğelerini bulmak için bunları kullanabiliriz
 - Ö **içerir()**-> `//*[içerir(metin(),'E-posta')]` ile
 - Ö **başlar()**-> `//*[ile başlar(text(),'Pass')]`
 - Ö **Metin()**-> `//*[metin()='Şifre']`
 - Ö **ebeveyn()**-> `//*[text()='Parola']`
 - Ö **parent::parentTagName[1] Ata()**-> `//*[text()='Parola']`
 - Ö **ata::tablo azalan()**-> `//*[@id='değer']/descendant::h1`
 - Ö **önceki/sonraki kardeş()**-> `//*[@class='_li']/div[1]/following-sibling::div`
- CSS Seçici basamaklı stil sayfası yoludur
- Bir web sayfasındaki bir öğeyi bulmak için kullanılan bir kalıptır.
- **CSS Seçici bulucuları oluşturma**
 - Ö elemanı kullanabiliriz# -> `#idValue` ön ekiyle kimlik
 - Ö elemanı kullanabilirizÖn ek ile Sınıf Adı. (nokta) -> `.sınıfAdı`
 - Ö Kullanabiliriz(>) alt öğelere erişmek için -> `.className>childNodes`
 - Ö Kullanabilirizalt öğelerin konumunu tanımlamak için first-child, last-child veya nth-child(n) yöntemleri.
`className>childNodes::nth-child(n)`
 - Ö Bunun yerine XPath'ta başlar-kullanabiliriz^ CSS Seçici'de -> `[özellik-adı^='değer']`
 - Ö Bunun yerine XPath ile biter, kullanabiliriz\$ CSS Seçici -> `[özellik-adı$='değer']` Bunun
 - Ö yerine XPath'de içerir, kullanabiliriz* CSS Seçici'de -> `[öznitelik-adı*='değer']`

FARKLAR:

- XPath daha esnek ve text(), include(), vb. öğelerin yerini belirleme yöntemlerine sahiptir. Ancak CSS Selector, bu yöntemlerden bazıları yerine kullanılacak bazı sözdizimlerini sağlar.
- XPath, DOM'a kadar gezinmemize izin verir; bu, alt öğeden üst öğeye geçebileceğimiz anlamına gelir, ancak şu anda bir öğenin üst öğesini CSS Seçici kullanarak seçmenin bir yolu yoktur.
- CSS Seçici, XPath'in yaptığı hemen hemen her şeyi yapabilir ve öğrenmesi daha kolaydır
- CSS Seçici, XPath'den daha hızlıdır
- **NOT:** XPath ve CSS Selector, id'den sonra en sık kullanılan 2 konumlandırıcıdır.

Dinamik ve statik web öğeleri nelerdir? Dinamik web öğeleri nasıl işlenir?

- **Statik web öğeleri** öznitelik değerleri statik olan ve değişmeyen olanlardır. bu yüzden kullanırken her zaman güvenli **dinamik web öğeleri** nitelik değerleri dinamik olan ve her tarayıcı oturumu için değişiklikler. Bu nedenle, kullanımı güvenli değildir ve bu web öğelerinde bazı eylemler gerçekleştirirken dikkatli olmamız gerekir.
- **Bir web öğesinin dinamik olduğunu nasıl anlarız?** Öznitelik değerlerinin tamamının veya bir kısmının rastgele sayılar veya harfler veya bunların kombinasyonları
- **Dinamik web öğelerinin riski nedir?** Bu web öğelerini değişen parçalarla bulursanız, komut dosyanız değer yeniden oluşturulacağından bir dahaki sefere başarısız olur ve betiğiniz bir istisna atar. Bu nedenle web elemanlarının yerini tespit ederken asla bu değerlere güvenmemeliyiz.
- **Dinamik web öğeleri nasıl işlenir:** Web öğesini statik değerlerine veya statik
 - § id niteliğinin dinamik ve statik parçalara sahip olduğunu varsayarak dinamik öğeleri bulmak için 3 yöntem kullanabiliriz `//div[contains(@name, "abd")]` `//div[birlikte başlar(@ad, "abd")]` `//div[birlikte biter(@id, "bca")]`
 - § Veya statik bir üst web öğesini bulabilir ve dinamik alt web öğesine gidebiliriz.

Selenium WebDriver kullanarak bir tarayıcıya nasıl öğle yemeği verilir?

- Selenium kullanarak tarayıcıyı öğle yemeği yemenin 2 yolu vardır

Ö1. System.setProperty() yöntemini kullanma

Ö CHROME SÜRÜCÜSÜNÜ BAŞLATMA ÖRNEĞİ

```
//webdriver.chrome.driver özelliğini yürütülebilir dosyasının konumuna ayarlama  
System.setProperty("webdriver.chrome.driver", "/lib/chromeDriver/chromedriver.exe");
```

```
//WebDriver arabirimini referans alan bir sürücü nesnesi oluşturma ve sürücü nesnesini başlatma  
WebDriver sürücüsü = yeni ChromeDriver();
```

```
//Bir web sayfasını açmak için get() yöntemini kullanma  
driver.get("https://www.techglobalschool.com");
```

```
//Tarayıcıyı kapatma  
sürücü.quit();
```

Ö FIREFOX SÜRÜCÜSÜNÜ BAŞLATMA ÖRNEĞİ //webdriver.gecko.driver özelliğinin ayarlanması

```
System.setProperty("webdriver.gecko.driver", pathToGeckoDriver + "\\geckodriver.exe");
```

```
//WebDriver arabirimini referans alan bir sürücü nesnesi oluşturma ve sürücü nesnesini başlatma
```

```
WebSürücüsü sürücüsü= yeni FirefoxDriver();
```

```
//Bir web sayfasını açmak için get() yöntemini kullanma  
driver.get("https://www.techglobalschool.com/");
```

```
//Tarayıcıyı kapatma  
sürücü.quit();
```

Ö2. WebDriverManager (Boni Garcia) Bağımlılığını Kullanma

Ö CHROME SÜRÜCÜSÜNÜ BAŞLATMA ÖRNEĞİ

```
WebDriverManager.chromedriver().setup();  
WebDriver sürücüsü = yeni ChromeDriver();
```

Ö FIREFOX SÜRÜCÜSÜNÜ BAŞLATMA ÖRNEĞİ

```
WebDriverManager.firefoxdriver().setup();  
WebDriver sürücüsü = yeni FirefoxDriver();
```

Ö NOT:İkinci yol için pom.xml dosyasında WebDriverManager bağımlılığını tanımlamamız gerekiyor.

Ö NOT:WebDriver'in Selenium ve ChromeDriver'da bir arayüz olduğunu ve FirefoxDriver'ın WebDriver Interface uygulayan 2 sınıf olduğunu bilmemiz gerekiyor ve bir Interface olduğu için WebDriver'i yeni WebDriver ile başlatamıyoruz.

Selenium kullanarak bir web sayfasını nasıl büyütebilirim?

- `driver.manage().window().maximize();`
- `driver.manage().window().fullscreen();`

`driver.close()` ve `driver.quit()` yöntemleri arasındaki fark nedir?

- **sürücü.kapat()** Yöntem, WebDriver'ın kontrol ettiği mevcut pencereyi kapatmak için kullanılır.
- **sürücü.quit()** Yöntem, WebDriver tarafından açılan tüm pencereleri kapatmak için kullanılır.

`get()` ve `navigasyon().to()` yöntemleri arasındaki fark nedir?

- **sürücü.get(url)** yöntem belirli bir URL'ye gitmek için kullanılır ve sayfa yüklenene kadar bekler
- **driver.navigate.to(url)** yöntemi belirli bir URL'ye gitmek için kullanılır ve sayfa yüklenene kadar beklemez ve bent tarayıcı geçmişini ve çerezleri korur, böylece komut dosyalarının yürütülmesi sırasında sayfalar arasında gezinmek için ileri ve geri işlevleri kullanabiliriz
- **BAZI DİĞER driver.navigate() yöntemleri aşağıdaki gibidir** Ö
 driver.navigate().back() sayfada geriye doğru gezinmek için
 Ö kullanılır **driver.navigate().forward()** sayfada ileri gitmek için
 Ö kullanılır **driver.navigate().refresh()** sayfayı yenilemek için kullanılır

Selenium kullanarak tarayıcıyı yenilemenin farklı yolları nelerdir?

- Tarayıcıyı yenilemenin farklı yollarından bazıları aşağıdaki gibidir
- `sürücü.navigate.refresh()`
- `driver.getCurrentURL()`
- `driver.navigate.to(driver.getCurrentURL())`
- `sendKeys(Anahtarlar.F5)`

`findElement()` ve `findElements()` yöntemleri arasındaki fark nedir?

- Her ikisi de öğeleri bulmak için kullanılır
- **findElement() yöntemi** tek bir web öğesi döndürür ve bir WebElement değişkeninde saklanabilir. eğer varsa verilen konumlandırıcıyla bulunan eşleşmelerden daha fazlaysa, DOM'deki ilk eşleşmeyi döndürür. Bu method **NoSuchElementException**'i atar belirtilen konumlandırıcı ile öğeyi bulamaması durumunda.
- **findElements() yöntemi** web öğelerinin bir listesini döndürür ve bunlar bir `ArrayList<WebElement>` içinde saklanabilir ve oboş bir liste döndürür belirtilen konumlandırıcıya sahip öğeleri bulamaması durumunda. Herhangi bir istisna atmaz.

getText() ve getAttribute() yöntemleri arasındaki fark nedir?

- **getText() yöntemi** bir öğenin metnini almak için kullanılır ve bir öğenin iç metnini bir Sicim. Alt öğelerle birlikte sayfada görünen metindir. Tüm baştaki ve sondaki boşlukları yok sayar.
- **getAttribute() yöntemi** belirli bir özelliğin değerini almak için kullanılır ve bir bir html belgesindeki öznitelik. Bir öznitelik için bir değer ayarlanmazsa, boş değer döndürülür. Öznitelik, yönteme parametre olarak iletilir.

Selenium kullanarak açılır menüler nasıl ele alınır?

- **Web sayfalarında 2 tür açılır menü vardır.** Biri seçme ve seçenek etiketleri kullanılarak oluşturulur, diğeri ise div gibi diğer etiketler tarafından oluşturulur.
- **Açılır menü, seç ve seçenek etiketleriyle oluşturulmamışsa**, sonra açılır menüyü bulmamız ve seçeneklerini bir ArrayList<WebElement> içinde saklamamız ve belirli seçenek üzerinde işlem yapmamız gerekir.
- **Açılır menü, seç ve seçenek etiketleriyle oluşturulmuşsa** Ö
Bu, geliştiriciler tarafından kullanılan en yaygın açılır menüdür
Ö Açılır menüdeki tüm seçenekler, seçenek etiketi kullanılarak
Ö oluşturulur. **Sınıf seçin** açılır menüden bir seçenek seçmek için

Ö **Select sınıfını kullanma adımları:**

1. Select öğesini bulmalıyız
2. Select sınıfından bir nesne oluşturun (Selenium kitaplığından gelir)
3. Select nesnesi oluştururken, içine select öğenizi iletmelisiniz.

Ö **Hareketler:**

getFirstSelectedOption() yöntemi ile varsayılan olarak seçili seçeneği alabiliriz. **Belirli bir seçeneği seçmenin 3 yolu**

- § 1. dizine göre
- § 2. görünür metinle
- § 3. değere göre

Tüm seçenekleri getOptions() yöntemiyle alabiliriz

ÖÖRNEK:

```
ÖSelect select = new Select (sürücü. findElement(By.id("idOfTheSelectElement")));
```

- Ö seçme. selectByVisibleText("Metin");
- Ö seçme. selectByIndex(intIndex);
- Ö seçme. selectByValue("Değer");

NOT: Select sınıfı yalnızca select ve options etiketleri kullanılarak oluşturulan açılır listeleri destekler, açılır listelerde kullanılan diğer HTML etiketleriyle birlikte kullanmaya ÇALIŞMAYIN, çalışmayacaktır.

Selenium'da Select sınıfı nedir ve neden kullanıyoruz?

- Select, Selenium kitaplıklarından gelen bir sınıftır.
- Select etiketi ile oluşturulan açılır menüden seçenekleri seçmek için kullanılır.
- Açılır menülerle kullanıma hazır birçok önceden tanımlanmış yöntemle sahiptir.

Neden kullanıyoruz?

Ö Açılır seçenekler üzerinde bazı işlemler yapmak için önceden tanımlanmış yöntemleri kullanabilmek

Bu nasıl kullanılır?

- Ö önce web ögesini seç ögesini bulun,
- Ö ardından bir Select sınıfı nesnesi oluşturun
- Ö ve bulunan web ögesini yapıcının içine parametre olarak iletin

Ne tür açılır menülerle kullanabiliriz?

Öyalnızca select etiketiyle özel olarak oluşturulan açılır menülerle kullanılabilir

iFrame nedir ve Selenium kullanılarak iFrame'ler nasıl ele alınır?

- **iFrame, başka bir html kodunun içindeki html kodu olarak bilinir.** ve bir sayfadan içerik koymak için kullanılır başka bir. İçerik reklamlar, videolar, oynatıcılar vb. olabilir.
- iFrame, iFrame etiketi kullanılarak oluşturulabilir
- Selenium aynı anda yalnızca bir html koduna odaklanabilir ve test kullanıcılarının iç iFrame'lerde bazı eylemleri gerçekleştirebilmeleri için iFrame'i açıkça değiştirmeleri gerekir.

Çerçeveye 3 farklı şekilde geçiş yapabiliriz:

- Ö **1 - Bir web ögesi olarak iç iFrame'i bulma ve onu switchTo() yöntemine geçirme**

Sözdizimi: `WebElement innerIframe = driver.findElement(By.id(""));`
`driver.switchTo().frame(innerIframe);`

- Ö **2 - Dizine göre geçiş; bu, iframe Sözdiziminizi bulmak için statik bir şeyiniz olmadığında faydalı olabilir:**`driver.switchTo().frame(0);`

- Ö **3- İsim veya kimliğe göre; dinamik Sözdizimi değilse:**

`driver.switchTo().frame("isim veya kimlik");`

- Ö **NOT: Üst iFrame'e geri dönmenin iki yolu vardır.**

§ `driver.switchTo().parentFrame();` --> bu, ana sürücüye geri

§ `dönecektir.switchTo().defaultContent();` --> bu varsayılan html'ye geri dönecek

Selenium kullanarak bir web sayfasındaki toplam çerçeve sayısını nasıl bulabilirsiniz?

- Tüm iFrame öğelerini tagName bulucu kullanarak bulabilir ve bunları bir Listede saklayabiliriz. Listenin boyutu, bir wen'de kaç çerçevemiz olduğunu döndürür

Sözdizimi:

`List<WebElements> çerçeveler= driver.findElements(By.tagName("iFrame"));`
`çerçeveler.size();` // bu, sayfadaki kare sayısını döndürür

Selenium kullanarak birden çok pencere/sekme nasıl işlenir?

- Pencere tutamacı, her pencere veya sekme için otomatik olarak oluşturulan bir kimliktir
- Birden çok pencereyi/sekmeyi kullanarak işleyebiliriz **getWindowHandle()**, **getWindowHandles()** ve **switchTo()**m
- Selenium için pencereler veya sekmeler aynıdır ve her birinin kendi benzersiz kimliği vardır
Ö **getWindowHandle()** yöntemi, geçerli pencerenin benzersiz kimliğini bir Dize olarak döndürür
§String currentWindowId = driver.getWindowHandle();
Ö **getWindowHandles()** yöntemi, Selenium WebDriver tarafından tarayıcıda açılan tüm pencere kimliklerini döndürür ve bir dizi benzersiz pencere kimlikleri döndürür. UNUTMAYIN, Set herhangi bir çoğaltmaya izin vermeyen ve benzersiz öğeler alan bir koleksiyondur.
§Set<String> allWindowIds = driver.getWindowHandles();
Ö **çevirmek()** yöntem, pencereyi veya sekmeyi değiştirmek için kullanılabilir
§driver.switchTo().window(windowToBeSwitched);

Uyarılar veya açılır pencereler nasıl ele alınır?

- 2 tür açılır pencere veya uyarı vardır
- **1. HTML açılır pencereleri veya uyarıları**
Ö Bu, HTML kodunun bir parçasıdır ve diğer herhangi bir web öğesini işlemek gibi işlenebilir. Böylece,
Ö driver.findElement() kullanarak açılır pencereyi bulabilir ve üzerinde işlem yapabiliriz.
- **2. JavaScript açılır pencereleri veya uyarıları**
Ö Bunlar HTML kodunun parçası değil
Ö Bu nedenle, normal Selenium konumlandırıcıları kullanarak onları bulamıyoruz ve bunları Alert arayüzü ile halletmemiz gerekiyor.
Ö **3 tür JavaScript uyarısı vardır**
§ **1 - Uyarı / Bilgi uyarısı:**
 - Bu uyarının işlem yapmak için yalnızca bir düğmesi vardır. Yalnızca alert.accept() yöntemini kullanarak kabul edebilirsiniz.
§ **2 - Onay uyarısı:**
 - Bu uyarının iki düğmesi vardır ve kabul edebilir veya reddedebiliriz
 - alert.accept() veya alert.dismiss() yöntemlerini kullanarak kabul edebilir veya reddedebilirsiniz.
§ **3 - Hızlı uyarı:**
 - Bu uyarı, kullanıcı girişi istiyor. Böylece, ona biraz metin gönderebilir ve ardından alert.sendKeys("text") ve alert.accept() veya alert.dismiss() kullanarak kabul edebilir veya reddedebilirsiniz.

ÖBir Alert örneği nasıl oluşturulur ve yöntemleri nasıl kullanılır?

```
Uyarı uyarısı = driver.switchTo().alert();  
uyarı.kabul();  
alarm.dismiss();  
alert.sendKeys("metin");
```

ÖBir JavaScript uyarısının veya açılır pencerenin metni nasıl alınır?

```
alert.getText() yöntemi kullanılabilir.
```

NOT: Tüm bu uyarı yöntemlerini Alert örneği oluşturmadan kullanabiliriz. Bunun yerine, driver.switchTo().alert() ve sonra kullanmak istediğimiz herhangi bir yöntemi kullanabiliriz.

Kaç tane uyarı yöntemi biliyorsunuz?

- Bu soru ilgili **JavaScript uyarıları**
- JavaScript uyarıları olarak da bilinir **tarayıcı uyarıları**
- Sahibiz **4 uyarı yöntemi**
 - Ö **Reddet()** yöntemi reddetmek için kullanılır (uyarıdaki İPTAL düğmesine tıklar) **kabul()**
 - Ö kabul etmek için yöntem kullanılır (uyarıdaki KABUL düğmesine tıklar) **sendKeys()**
 - Ö uyarıdaki giriş kutusuna anahtar göndermek için yöntem kullanılır **getText()** uyarıda
 - Ö görüntülenen metni almak için yöntem kullanılır

Actions sınıfı nedir ve neden kullanıyoruz?

- **Actions, Selenium'dan gelen bir sınıftır.** ve otomasyonda gelişmiş fare ve klavye eylemleri gerçekleştirmemizi sağlar
- Eylemler yöntemi çalışmadığı sürece çalışmayacaktır. **rol yapmak()** yapmak istediğiniz gerçek eylemden sonra kullanılır

Eylemler sınıfı işleyebilir

Ö 1. Fare Eylemleri

- § **tıklama() yöntemi** basitçe bir öğeye tıklar
- § **doubleClick() yöntemi** bir öğeye çift tıklama gerçekleştirir **bağlamClick() yöntemi** bir öğe üzerinde bağlam tıklaması (sağ tıklama) gerçekleştirir **clickAndHold() yöntemi** geçerli fare konumunda tıklamalar (bırakmadan) **dragAndDrop(kaynak, hedef) yöntemi** kaynak öğenin konumunda tıkla ve tut gerçekleştirir, ardından kaynak öğeyi hedef öğenin konumuna taşır, ardından fareyi serbest bırakır
- § **dragAndDropBy(source, x-offset, y-offset) yöntemi** kaynak öğenin konumunda tıkla ve tut gerçekleştirir, belirli bir ofset ile hareket ettirir, ardından fareyi serbest bırakır
- § **moveToElement(toElement) yöntemi** fareyi öğenin ortasına taşır **moveByOffset(x-offset, y-offset) yöntemi** fareyi mevcut konumundan (veya 0,0) verilen ofset kadar hareket ettirir
- § **yayın() yöntemi** geçerli fare konumunda basılı sol fare düğmesini serbest bırakır

Ö2. Klavye Eylemleri gibi

- § **keyDown(modifier_key) yöntemi** bırakmadan bir tuşa basar
- § **keyUp(modifier_key) yöntemi** bir anahtar bırakma gerçekleştirir
- § **sendKeys(onElement, karakter dizisi) yöntemi** öğeye bir dizi tuş vuruşu gönderir

Eylemler sınıfı nasıl kullanılır?

- Aşağıdaki üç adımı izleyin
 - Ö 1. **Actions sınıfının bir nesnesini oluşturun** Eylemler eylemler = yeni Eylemler(WebDriver);
 - Ö 2. **aşağıdaki gibi Actions sınıfından gelen herhangi bir yöntemi yürütmek için oluşturulan nesne referansını kullanın** eylemler.moveToElement(element).actions.doubleClick(element)

- Ö 3. Aşağıdaki gibi nesne referansı ile çağıracağınız eylemi gerçekleştirmek için perform() yöntemini kullanın
eylemler.moveToElement(element).perform(); eylemler.doubleClick(element).perform();

NOT: perform() yöntemini kullanmazsak Actions yöntemleri yürütülmez.

Actions sınıfını kullanarak bir web öğesine nasıl çift tıklanacağını söyleyebilir misiniz?

- yapabilmek için aşağıdaki adımları izleyin: **çift tıklama gerçekleştirmek için Actions sınıfını kullanın** Ö
Önce Actions örneğini oluşturun, aksi takdirde Actions sınıf yöntemlerini kullanamazsınız
Actions act = new Actions(driver);
act.doubleClick(webElement).perform();

Selenium WebDriver'da isDisplayed(), isEnabled() ve isSelected() yöntemlerinin farkları nelerdir?

- Tüm bu yöntemler web öğeleriyle kullanılır ve true veya false döndürür.
- Çoğunlukla Boole koşulları olarak Assert.assertTrue() veya Assert.assertFalse() yöntemlerine iletilirler.
- görüntüleniyor()** yöntemi, belirli bir web öğesinin varlığını doğrulamak için kullanılır ve eğer olursa true değerini döndürür. eleman bulunur, aksi halde false
- etkin()** çoğunlukla düğmelerle kullanılır ve web öğesinin etkin mi yoksa devre dışı mı olduğunu doğrular
- seçildi()** çoğunlukla radyo düğmeleri, açılır menüler ve onay kutuları ile kullanılır ve web öğesinin seçili olup olmadığını doğrular.

Anahtar değiştiriciler nelerdir?

- Otomasyon sırasında genellikle klavyeden enter, kontrol, sekme, ok tuşları, işlev tuşları ve diğer metin olmayan tuşlara basmamız gerekir. Selenium, bu görevleri gerçekleştirmek için Keys enum sağlar

ÖRNEKLER

- Ö Anahtarlar.ENTER
- Ö Tuşlar.KONTROL
- Ö Tuşlar.SHIFT
- Ö Tuşlar.ALT
- Ö Tuşlar.TAB
- Ö Anahtarlar.GERİ DÖN
- Ö Anahtarlar.ARROW_DOWN
- Ö Anahtarlar.ARROW_UP
- Ö Anahtarlar.ARROW_LEFT
- Ö Anahtarlar.ARROW_RIGHT
- Ö Tuşlar.F5
- Ö Keys.DELETE ve daha fazlası

Selenium'da karşılaştığınız bazı istisnaları açıkla mısınız?

- **NoSuchWindowException** bir eleman bulunamazsa oluşur
- **TimeoutException** bir komutun tamamlanması için yeterli zaman olmadığında oluşur. Örneğin aranan eleman belirtilen sürede bulunamadı.
- **StaleElementReferenceException** web ögesi geçerli DOM'den ayrıldığında, yani web ögesinin referansını kaybettiğinde oluşur. Bu istisna genellikle tarayıcı yenilendiğinde ortaya çıkar.
- **ElementClickInterceptedException** eylemleri alan öge, tıklanması istenen ögeyi gizlediğinde oluşur
- **ElementNotInteractableException** DOM'da herhangi bir öge sunulduğunda oluşur. Ancak, böyle bir unsurla etkileşime girmek imkansızdır.
- **NoSuchFrameException** var olmayan bir çerçeveyi değiştirmeye çalıştığımızda oluşur
- **NoAlertPresentException** sunulmayan bir uyarıya geçtiğinizde oluşur
- **NoSuchWindowException** geçiş yapılacak pencere hedefi yoksa oluşur

NOT: Otomasyon yaparken karşılaşılabileceğiniz daha birçok Selenium istisnası vardır ancak bunlardan yalnızca bazıları yukarıda listelenmiştir.

Selenium'da bulunan bekleme türleri nelerdir? Örtük, açık ve akıcı beklemeleri açıklayın?

- Selenium'da beklemeler gereklidir **sürücüyü kodla senkronize et**
- Bizim **kod her zaman sürücünden daha hızlıdır** ve bu, kodunuzla ilgili herhangi bir sorun olmamasına rağmen bazı istisnaların atılmasına neden olabilir, ancak bunun nedeni, kodunuzun sürücünün kendisinden çok daha hızlı olmasıdır ve otomatik olarak istediğiniz sayfa yüklenmeyebilir.
- Sütlerimizde bu tür test hatalarından kaçınmak için şunları kullanabiliriz: **örtük, açık ve akıcı bekleyişler**
- önerilmez ama **Konu.uyku()** yöntem ayrıca kodumuzu belirli bir süre dondurur ve sürücüyü kodla senkronize eder.
- **HATIRLAMAK:** Kodumuz her zaman sürücü ile aynı sayfada olmalıdır, aksi takdirde beklenmedik durumlarla karşılaşırız.

sonuçlar ve başarısızlıklar

Öörtük bekle

Ö Web sürücüsüne, bir mesaj atmadan önce belirli bir süre beklemesini söylemek için kullanılır.

NoSuchElementException

Ö Varsayılan ayar 0'dır. Zamanı ayarladıktan sonra, web sürücüsü bir istisna atmadan önce o zaman için ögeyi bekleyecektir.

Ö Sözdizimi:

driver.manage().timeouts().implicitlyWait(TimeOut, TimeUnit.SECONDS);

NOT: Öörtülü bekleme, sürücünün bulunan ögeyi bulmasını bekleyecek ve sürücü ögeyi bulur bulmaz devam edecek, ancak öge verilen zaman diliminde bulunmazsa, istisna atacaktır.

ÖAçık Bekleme

Ö Web sürücüsüne belirli koşulları beklemesini söylemek için kullanılır. **(Beklenen Koşullar)** veya fırlatmadan önce maksimum süre aşıldı **ElementNotVisibleException**

Ö Akıllı bir bekleme türüdür ve yalnızca belirtilen öğeler için uygulanabilir. Dinamik olarak yüklenen Ajax öğelerini beklerken örtük beklemeden daha iyi seçenekler sunar

Ö Sözdizimi:

//WebDriverWait sınıfının bir nesnesini oluşturun WebDriverWait bekle =
yeni WebDriverWait(sürücü, zaman aşımı);

//Belirli bir koşul için açık beklemeyi tanımlamak için oluşturulan WebDriverWait nesnesini
kullanın wait.until(ExpectedConditions.visibilityOf(someElement));

Öakıcı bekle

- Ö Web sürücüsünün bir koşulu beklemesi için maksimum süreyi ve ayrıca bir uyarı atmadan önce durumu kontrol etmek istediğimiz sıklığı tanımlamak için kullanılır. **ElementNotVisibleException** Nesne bulunana
- Ö veya zaman aşımı gerçekleşene kadar web ögesini düzenli aralıklarla kontrol eder. **ÖRNEK:** Bir ögeyi
- Ö bulmanız ve üzerinde bazı eylemler gerçekleştirmeniz gerektiğini hayal edin. Ancak, bu belirli öge bazen 2 saniyede yüklenir, bazen de yüklenmesi 30-50 saniye sürer. Bu durumda veya benzeri durumlarda, belirli bir süre sonra açık bekletme ve istisna atacağı için açık bekleme yerine akıcı bekleme kullanmak faydalı olabilir. Akıcı bekleme, ögeyi bulana kadar tanımlanmış zaman aralığında sık sık bulmaya çalışır ve son zamanlayıcı bittiğinde istisna atar

Ö Sözdizimi:

Bekle bekle = new FluentWait(driver).withTimeout(timeout, timeUnit.SECONDS) .
pollingEvert(zaman aşımı, timeUnit.SECONDS).ignoring(Exception.class);

ÖKonu.uyku()

- Ö Tüm programın yürütülmesini duraklatmak için kullanılır ve herhangi bir koşula bağlı değildir Basitçe,
- Ö kodu dondurur ve sürücüyü kodla senkronize etmek için iyi bir uygulama olarak kabul edilmez.

Selenium kullanarak nasıl aşağı veya yukarı kaydırılır?

- Otomasyon yaptığımızda, sayfanın altında görünmeyebilecek bazı öğeler bulunabilir ve bu görünmez öğeler üzerinde bazı eylemler yapmaya çalışırsak komut dosyamız beklenmedik bir istisna oluşturabilir.
- Bu gibi durumlarda, bu beklenmedik istisnalardan kaçınmak için sayfayı aşağı veya yukarı kaydırmamız gerekir. Selenium'da 3 farklı kaydırma yolu görelim

§ 1. Actions sınıfından action.moveToElement().perform() yöntemini kullanma

- Ö Aşağı kaydırmak istediğiniz ögeyi bulun
- Ö moveToElement(element).perform() yöntemini
- Ö kullanın Bu, bulunan ögeye sayfa gönderir

§ 2. action.sendKeys(Keys.PAGE_DOWN).perform() işlevini kullanma

- Ö Bu satır, kullanıcı klavyede PAGE_DOWN tuşuna basarken taklit edilecektir.
- Ö Üzerinde bir görev yapmak istediğiniz web ögesine ulaşana kadar bu satırı istediğiniz kadar iletebilirsiniz.

§ 3. JavascriptExecutor'ı Kullanma

- Ö Aşağı veya yukarı kaydırmak için kullanabileceğimiz farklı JavaScriptExecutor yöntemleri vardır.

//Önce bir JavaScriptExecutor örneği oluşturmamız gerekiyor

JavascriptExecutor js = (JavascriptExecutor)sürücü;

js.executeScript("window.scrollTo(x, +y)");// yukarı kaydırmak için kullanılır

js.executeScript("window.scrollTo(x, -y)");// aşağı kaydırmak için kullanılır

js.executeScript("argümanlar[0].scrollIntoView(true);", eleman);// belirli bir ögeye

kaydırmak için kullanılır

Selenium'da neden senkronizasyona ihtiyacımız var?

- Komut dosyalarımız çalıştığında hem Java hem de Selenium kullanır ve kodumuz her zaman WebDriver'dan daha hızlıdır. Kodumuz ve Web Sürücümüz paralel veya senkronize çalışmıyorsa, beklenen sonuçları alamayabiliriz. Yani, yapma senkronize veya paralel olarak çalışırlar. **senkronizasyon**
- **Senkronizasyon sorunlarını önlemek için**, kodu yavaşlatmak ve senkronize etmek için beklemleri (örtük, açık) kullanmamız gerekiyor WebDriver hızı ile **NOT:**Thread.sleep(), kodu duraklattığı için senkronizasyon için uygun bir yol değil

Selenium WebDriver'ın yetenekleri nelerdir?

- Birden çok çerçeveyi, açılır pencereyi, birden çok tarayıcı penceresini ve uyarıları işleme
- Sayfa gezinme
- Sürükle ve bırak gibi gelişmiş fare ve klavye eylemleri
- Ajax tabanlı UI öğeleri
- Chrome, IE, Firefox gibi çoklu tarayıcı testleri
- Farklı işletim sistemlerini ve programlama dillerini destekleme

WebDriver'da bulunan farklı sürücü türleri nelerdir?

- FirefoxSürücüsü
- İnternet GezginiSürücü
- ChromeSürücüsü
- SafariSürücüsü
- OperaSürücüsü
- AndroidSürücüsü
- iPhoneSürücüsü
- HtmlUnitSürücü

WebDriver kullanarak tarayıcı nasıl başlatılır?

- Aşağıdaki sözdizimleri için kullanılabilir **tarayıcıyı başlat:**
WebDriver sürücüsü = yeni FirefoxDriver(); // Firefox tarayıcısının başlatılması
WebDriver sürücüsü = yeni ChromeDriver(); // Chrome tarayıcı başlatılıyor
WebDriver sürücüsü = yeni InternetExplorerDriver(); // IE tarayıcısını başlatma

Mobil Test için desteklenen farklı sürücüler nelerdir?

- AndroidSürücüsü
- IphoneSürücüsü
- OperaMobileSürücü

Hangi WebDriver uygulaması en hızlıdır?

- **HtmlUnitSürücü** uygulama en hızlıdır
- **HtmlUnitSürücü** arayüze **başsız tarayıcı testi** bu sadece bir kullanıcı olmadan Selenium testlerinizi çalıştırıyor izin verir
- Tipik tarayıcınızın yaptığı gibi çalışır, ancak kullanıcı arayüzü olmadan ve diğerlerine kıyasla çok daha hızlıdır.

HtmlUnitDriver nedir?

- **HtmlUnitSürücü** WebDriver'ın en hızlı uygulamasıdır
- Olarak bilinir **başsız tarayıcı sürücüsü**
- Chrome, IE veya Firefox sürücüsü ile aynıdır, ancak kullanıcı arayüzü yoktur, bu nedenle testleri yürütürken ekranda test yürütmesini göremiyoruz
- **NOT:**Chrome, Firefox veya diğerleri gibi gerçek tarayıcılar kadar güvenilir değildir

Başsız tarayıcı testinin avantajı nedir?

- Başsız tarayıcı **Daha hızlı** UI göstermediği ve test cihazlarının diğerlerine kıyasla kısa sürede test takımlarını çalıştırmasını sağladığı için diğer gerçek tarayıcılardan farklıdır.

Selenium WebDriver kullanılarak mobil uygulamalar test edilebilir mi?

- Hayır, Selenium yerel mobil uygulamaları test etmek için kullanılamaz
- Ancak Selenium web tabanlı bir test aracıdır; bu nedenle, yalnızca mobil tarayıcılardaki uygulamaları test etmek için kullanılabilir

Captcha, Selenium ile otomatikleştirilebilir mi?

- Hayır, captcha ve barkod okuyucu otomatikleştirilemez ve bu Selenium kısıtlamalarından biridir.
- Captcha'nın tüm konsepti, botların ve otomatikleştirilmiş programların hassas bilgilere erişmemesini sağlamaktır - bu nedenle Selenium bunu otomatikleştiremez.
- Otomasyon test mühendisi, diğer alanlar otomatik olarak doldurulabilirken captcha'yı manuel olarak geçmek zorundadır.

Selenium, Windows tabanlı açılır pencereleri işleyebilir mi?

- Hayır, Windows tabanlı açılır pencereler Selenium ile işlenemez ve bu Selenium kısıtlamalarından biridir.
- Selenium, yalnızca web uygulaması testini destekleyen bir otomasyon test aracıdır. Bu nedenle, Windows açılır pencereleri Selenium kullanılarak işlenemez.

Windows tabanlı açılır pencereleri nasıl halledebiliriz?

- Selenium, yalnızca web uygulaması testini destekleyen bir otomasyon test aracıdır, yani Windows tabanlı uygulamaların test edilmesini desteklemez.
- Ancak bu sorun aşılabılır
- Bir kaç tane var **üçüncü taraf araçları** AutoIT, Robot sınıfı vb. gibi Selenium ile birlikte pencere tabanlı açılır pencereleri işlemek için kullanılabilir.

Selenium kullanarak bir web sayfasını yenilemenin farklı yöntemleri nelerdir?

- Selenium kullanarak bir sayfayı yenilemenin birden çok yolu vardır.
 1. driver.navigate komutunu kullanma
`driver.navigate().refresh();`
 2. driver.getCurrentUrl()'u driver.get() komutuyla kullanma
`driver.get(driver.getCurrentUrl());`
 3. driver.getCurrentUrl()'u driver.navigate() komutuyla kullanma
`driver.navigate().to(driver.getCurrentUrl());`
 4. SendKeys komutunu kullanarak herhangi bir metin kutusunda F5 tuşuna basmak
`driver.findElement(By.textboxLocator).sendKeys(Keys.F5);`
 5. F5 anahtarının ASCII değerini iletme, yani sendKeys komutunu kullanarak "\uE035"
`driver.findElement(By.textboxLocator).sendKeys("\uE035");`

Selenium kullanarak bir web öğesinin metni nasıl iddia edilir?

2. Web öğesini bulun
`WebElement öğesi = driver.findElement(By.id("ElementID"));`
3. Öğenin metnini getText() yöntemini kullanarak alın ve saklayın
`String trueText = element.getText();`
4. Beklenen sonuçla metni onaylayın
`Assert.assertEquals("Beklenen Metin", aktüelMetin);`

KISA - UYGUN VERSİYON

WebElement öğesi = driver.findElement(By.id("ElementID")); // bir Sayfa sınıfında olabilir
Assert.assertEquals("Beklenen Metin", element.getText()); // bir yöntem veya adımda olabilir

switchTo() komutu ne işe yarar?

- switchTo() komutu, uygulama içinde pencereler, çerçeveler veya açılır pencereler arasında geçiş yapmak için kullanılır

Selenium kullanarak çerçeveler arasında nasıl geçiş yapılır?

- Çerçeveler arasında geçiş yapmak için driver.switchTo().frame() yöntemi kullanılabilir
- Çerçeveler arasında 3 şekilde geçiş yapmak mümkündür Ö
 - İç iFrame öğesini bulun ve şunu kullanın - `driver.switchTo().frame(innerIframe);` Dizini
 - Ö kullanarak geçiş yap - `driver.switchTo().frame(index);`
 - Ö iFrame adına veya kimliğine göre geçiş yap - `driver.switchTo().frame("isim veya kimlik");`

Selenium kullanarak bir web sayfasının başlığı nasıl iddia edilir?

1. getTitle() yöntemini kullanarak başlığı alın ve bir String değişkeninde saklayın

```
String factTitle = driver.getTitle();
```

2. Beklenen sonuçla saklanan gerçek başlığı iddia edin

```
Assert.assertEquals("Beklenen başlık", actiTitle);
```

KISA - UYGUN VERSİYON - `>Assert.assertEquals("Beklenen başlık", driver.getTitle());`

Selenium kullanarak bir metin kutusuna nasıl yazılır?

1. Web ögesini bulun

```
WebElement ögesi = driver.findElement(By.id("ElementID")); //bir Sayfa sınıfında olabilir
```

2. sendKeys() yöntemini kullanarak metni yazın `element.sendKeys("gönderilecek`

```
tuşlar"); //bir yöntem veya adımda olabilir
```

sendKeys() kullanmadan metin kutusuna yazmanın bir yolu var mı?

- Evet. JavaScriptExecutor JavascriptExecutor jse = (JavascriptExecutor) sürücüsü kullanılarak metin bir metin kutusuna girilebilir;
jse.executeScript("document.getElementById('elementId').value='gönderilecek anahtarlar'");

Selenium'da linkText kullanarak bir hiper bağlantıya nasıl tıklanır?

1. linkText bulucu ile köprü web ögesini bulun

```
WebElement ögesi = driver.findElement(By.linkText("metin")); //bir Sayfa sınıfında olabilir
```

2. Öğeye tıklayın

```
element.click(); //bir yöntem veya adımda olabilir
```

NOT:Yukarıdaki bağlantı, kısmiLinkText bulucu kullanılarak da bulunabilir.

```
driver.findElement(By.partialLinkText("kısmi metin")); //bir Sayfa sınıfında olabilir
```

Selenium kullanarak çift tıklama nasıl

yapılır? Actions class 1'i kullanarak çift tıklama yapabiliriz.

```
Eylemler eylemler = yeni Eylemler(sürücü); //Bu, sürücüyü yapıcıda parametre olarak alacaktır.
```

2. Varsayalım ki çift tıklayacağınız eleman daha önce yer almaktadır.

```
WebElement ögesi = driver.findElement(By.id("elementId")); //bir Sayfa sınıfında olabilir
```

3. Bulunan öğeye çift tıklamak için Actions sınıfı doubleClick() yöntemini gerçekleştirin

```
hareketler. doubleClick(element).perform(); //bir yöntem veya adımda olabilir
```

Selenium kullanarak bir web öğesinin üzerine fareyle nasıl gelinir?

WebDriver, kullanıcının fare ve klavye olaylarını otomatikleştirmek için kullanılabileceği çok çeşitli etkileşim yardımcı programları sunar. **Eylemler sınıfı**

1. Bir Eylem nesnesi somutlaştırın

Eylemler eylemler = yeni Eylemler(sürücü); //Bu, sürücüyü yapıcıda parametre olarak alacaktır.

2. Üzerine gelinecek öğenin daha önce yer aldığını varsayalım.

WebElement öğesi = driver.findElement(By.id("elementId")); //bir Sayfa sınıfında olabilir

3. İmleci bulunan öğeye taşımak için Actions sınıfı moveToElement() yöntemini gerçekleştirin

eylemler.moveToElement(element).perform(); //bir yöntem veya adımda olabilir

Selenium kullanarak web sayfalarındaki tablolar nasıl işlenir?

- Web tabloları, verilerin satırlar ve sütunlar kullanılarak yapılandırılmış bir biçimde sunulduğu normal tablolar gibidir.
- Tek fark, web üzerinde HTML kodu yardımıyla görüntülenmeleridir.
- **<table>** bir web tablosunu tanımlamak için kullanılan HTML etiketidir. Süre **<th>** başlığını tanımlamak için kullanılır. masa, **<tr>** ve **<td>** etiketleri, web tablosu için sırasıyla satırları ve sütunları tanımlamak için kullanılır.
- Web'de yayınlanan iki tür HTML tablosu vardır. Ö

1. Statik tablolar: Veri statiktir, yani satır ve sütun sayısı sabittir

Ö **2. Dinamik tablolar:** Veri dinamiktir, yani satır ve sütun sayısı sabit değildir

NOT: Statik tabloyu işlemek kolaydır ancak satırlar ve sütunlar sabit olmadığı için Selenium'da dinamik tabloları işlemek biraz zordur

HTML kullanarak bir web tablosu yazma örneği:

```
<table>
<tr>
<th>First Name</th>
<th>Last Name</th>
<th>Age</th>
</tr>
<tr>
<td>Jill</td>
<td>Ann</td>
<td>24</td>
</tr>
<tr>
<td>Eve</td>
<td>Anderson</td>
<td>34</td>
</tr>
</table>
```

First Name	Last Name	Age
Jill	Ann	24
Eve	Avans	34

- Elemanları bularak hem statik hem de dinamik tablolar işlenebilir
- Diğer herhangi bir web elemanı gibi, web tablosunda herhangi bir işlemi gerçekleştirmek için incelemek tablo verilerini ve öğeleri bulun
- XPath, ebeveyninden çocuğa ve çocuktan ebeveyne geçişe izin verdiği için tablolarla çok yaygın olarak kullanılır.
- CSSSelector, XPath kadar esnek olmasa da kullanılabilir.
- Çerçevelerdeki tabloları işlemek için bazı yardımcı yöntemler oluşturmak çok yaygındır.
 - Ö Bir tablo web öğesini parametre olarak alan ve tüm tablo sütun başlıklarını liste olarak döndüren bir yöntem oluşturabilirsiniz.
 - Ö Bir tablo alıp tüm verileri döndüren başka bir yöntem oluşturabilirsiniz veya başka bir yöntem belirli bir satırdaki tüm verileri döndürebilir.

Selenium kullanarak dosya indirme/yükleme nasıl yapılır?

• İndirmek

ÖSelenium, dosya indirme işlemlerini gerçekleştiremez

- Ö Web sürücüsünde bir dosyanın indirme düğmesini bulabilir ve bulunan indirme düğmesine tıklayarak dosyayı indirebiliriz. Ancak Selenium'un kapsamı web sürücüsü ile sınırlıdır ve dosyanın gerçekten indirilip indirilmediğini kontrol etmek için kullanılamaz veya indirilen dosyanın içeriğini doğrulamak için kullanılamaz.
- Ö Dosyanın indirilip indirilmediğini kontrol etmek veya dosyanın içeriğini doğrulamak için Robot ve AutoIT gibi üçüncü kısım araçlar kullanılabilir.
- Ö Sadece dosyanın indirilip indirilmediğini kontrol etmek için Java, File sınıfını kullanarak ve dosyanın yolunu geçerek de kullanılabilir.

• Yüklemek

ÖSelenium dosya yüklemeyi işleyebilir

Öadımlar

- § Genellikle 2 elementi bulmamız gerekir. 1, yüklemek istediğimiz dosyanın yolunu geçmek için giriş alanı ve 2, yükle düğmesi
- § Bu 2 elemanı bulduktan sonra sendKeys(path) metodunu kullanarak dosya yolunu girdi alanına geçireceğiz ve click() metodu ile upload butonuna tıklayacağız.

Assert ve doğrulama komutları arasındaki fark nedir?

- Assert komutu verilen koşulun doğru mu yanlış mı olduğunu kontrol eder. Ö
Diyelim ki verilen öğenin web sayfasında olup olmadığını iddia ediyoruz.
 - Ö Koşul doğruysa, program kontrolü bir sonraki test adımını yürütür, ancak koşul yanlışsa, yürütme hemen durur ve test bir istisna atar.
 - Ö İstisna atıldıktan sonra akış mevcut test ile devam etmeyecek ve varsa bir sonraki test ile devam edebilir.
 - Ö Sert ve yumuşak iddialar olarak bilinen 2 tür iddia vardır.
- Verify komutu ayrıca verilen koşulun doğru veya yanlış olup olmadığını kontrol eder ve bir uygulamanın kritik olmayan fonksiyonlarını kontrol etmek için kullanılır.
 - Ö Koşulun doğru veya yanlış olmasına bakılmaksızın, program yürütmesi durmaz ve istisna atılmaz
 - Ö Doğrulama sırasındaki herhangi bir hata, yürütmeyi durdurmaz ve bazı adımlarda testin başarısız olmasına bakılmaksızın tüm test adımları yürütülür.

Sert iddia ve yumuşak iddia arasındaki fark nedir?

• Zor iddia, hemen bir AssertException oluşturur

bir assert ifadesi başarısız olduğunda ve test paketi devam ettiğinde sonraki @Test ile.

ÖAssert adımları başarısız olursa, testin yürütülmesi o noktada durur! ve varsa bir sonraki teste gidecek!

• Soft assert, @Test sırasında hataları toplar

- Ö Soft Assert, bir assert başarısız olduğunda bir istisna oluşturmaz ve assert ifadesinden sonraki adımla devam eder.
- Ö Yumuşak onaylama adımı başarısız olursa, başarısız olduğunu bildirir ancak yürütmeye devam eder

İstenen yetenek nedir? Neden önemli?

- İstenen yetenek, çalışma zamanında tarayıcının davranışını belirlemek için tarayıcı adı, tarayıcı sürümü, sistemdeki tarayıcı sürücüsünün yolu vb. gibi tarayıcı özelliklerini depolayan bir dizi anahtar/değer çiftidir.
- Test senaryolarını farklı işletim sistemleri ve sürümleri ile farklı bir tarayıcıda çalıştırmak istediğinizde Selenium WebDriver'ın sürücü örneğini yapılandırmak için kullanılabilir.

Nesne deposu nedir?

- Nesne deposu, web öğelerinizi konum belirleyicileriyle birlikte depolamak için oluşturduğunuz sayfadır.
- Testçilerin, test komut dosyaları içinde sabit kodlamayla sınırlı olarak, bir veya daha fazla merkezi konumda, konumlandırıcı değerleriyle birlikte uygulamanın web öğelerini toplamasına olanak tanır.
- Kod okunabilirliğini artırır
- Bir nesne deposunda depolanan öğeler, birden çok test komut dosyası için kullanılabilir
- Kodun yeniden kullanılabilirliğini sağlar
- Konum belirleyiciler için bir güncelleme olduğunda eleman bulucuların bakımını kolaylaştırır

Nesne deposunda (sayfa sınıfları) ne saklıyorsunuz?

- Öncelikle uygulamada her sayfa için ayrı sayfa sınıfları oluşturulması uygundur.
- Test cihazı, belirli bir sayfayla ilgili tüm web öğelerini ve sayfayı test etmek için yararlı olabilecek yöntemleri merkezileştirebilir (depolayabilir)

Bir nesne deposu örneği:

Kamusal sınıf Ana Sayfa(){

//PageFactory sınıfını ve initElements() yöntemini kullanarak web öğelerini başlatan yapıcı halka

açık Ana Sayfa(){

PageFactory.initElements(driverInstance, **Bu**);

}

//Uygulamanın Ana Sayfası ile ilgili tüm web öğelerini @FindBy() ek açıklamasını kullanarak depolayın - Örnek

@FindBy(id = "webElementId") **halka açık** WebElement öğeAdı;

//Ana Sayfayı test etmek için kullanılacak kullanışlı - yeniden kullanılabilir yöntemler oluşturun - Örnek

Kamu geçersiz validateUserIsOnHomePage(){

//Yöntem gövdesi - kullanıcının Ana Sayfada olduğunu doğrulama kodu

}

}

Selenium Senaryo Tabanlı Mülakat Soruları

Bir NoSuchElementException aldığınızı varsayalım. Sorunu gidermek için ne yapardınız?

- Bu, özellikle uygulama geliştirme aşamasındayken ve henüz yayında değilken Otomasyon Testçilerinin karşılaştığı yaygın bir sorundur.
 - NoSuchElementException** WebDriver bir öğeyi bulamadığında ve bulamadığında oluşur
 - Bu tür sorular için aşağıdaki gibi aklınıza gelebilecek kadar senaryo yazmaya çalışın.
 - Öğeler bulucunuzu kontrol edin ve doğru olduğundan emin olun. Konumlandırıcı, yeni geliştirici kodu dağıtımıyla değiştirilebilir. Ayrıca, fırlatan web öğesinin de farkına varabilirsiniz. NoSuchElementException dinamik bir öğedir. Bu durumda, bulucunuzu güncellemeniz ve kararlı hale getirmeniz gerekir.
 - Sayfanın beklendiği gibi yüklendiğinden emin olun
 - Konumlandırıcınızın doğru ve kararlı olduğundan eminseniz, bazı senkronizasyon sorunları olabileceğinden ve kodunuz yüklenmeden önce web öğesi üzerinde bazı işlemler yapmaya çalıştığınızdan bazı beklemler uygulamaya çalışın.
 - Bekleme uyguladıysanız ancak yine de istisna alıyorsanız, öğenin bir iframe içinde olup olmadığını kontrol edin. Öyleyse, web öğesi üzerinde işlem yapmadan önce çerçeveyi değiştirin ve çalışıp çalışmadığını görmek için testinizi bir kez daha gerçekleştirin.
 - Öğeler ekranda gerçekte görünmeyebilir ve gerçek eylemden önce aşağı kaydırmanız veya o öğeye yukarı kaydırmanız gerekebilir.
- NOT:** Bazı durumlarda, web öğesi mevcut olmayabilir ve sayfadan kaldırılabilir. Gereksinimlerinizi dikkatlice kontrol edin ve PO, BA veya geliştiricilerinizden sorun hakkında net olmalarını isteyin.

Bir ElementClickInterceptedException aldığınızı varsayalım. Sen ne yapardın? Sorunu nasıl giderirsiniz?

- Bu aynı zamanda Otomasyon Testçilerinin komut dosyası oluşturma sırasında aldığı yaygın bir istisnadır.
- ElementClickInterceptedException** eylemleri alan öğe, tıklanması istenen öğeyi gizlediğinde oluşur
- Temel olarak, bir web öğesini tıklamaya çalıştığınızı, ancak öğeniz o ikinci öğenin arkasında olduğu için tıklamayı başka bir web öğesinin aldığını hayal edin.
- Bu durumda aşağıdaki çözümleri deneyebilirsiniz Ö
 - İlk olarak, web öğesinde gerçekleştirmek istediğiniz eylem için açık beklemeyi uygulamayı deneyebilir ve çalışıp çalışmadığını kontrol edebilirsiniz.
 - Ö Açık bir şekilde bekleme işe yaramazsa, eylemi gerçekleştirmek için JavaScriptExecutor'ı kullanabilirsiniz.

Bir sayfaya gittiğinizi ve bir açılır pencere yüklediğinizi varsayalım. Ancak bu bazen olur ve çoğu zaman açılır pencere görünmez. Bu durumu nasıl ele alırdınız?

- Bazı sayfalarda bir açılır pencere gösterilir, ancak bu sabit değildir ve sayfadaki diğer işlemlere devam etmeden önce açılır pencerenin ele alınması gerekir.
- Açılır pencereyi her zaman göstereceğini ve işleyeceğini varsayarsanız, açılır pencere gösterilmediğinde kodunuz atar ve istisna olur.
- Bu nedenle, kodunuzu bir try bloğuna yazmak ve açılır pencereyi yalnızca mevcutsa işlemek için en iyi uygulama

Smoke ve Regression gibi test takımlarınızı Chrome, Firefox, Edge ve daha fazlası gibi farklı tarayıcılarda çalıştırmanız gerekeceğini varsayalım. Bu durumu nasıl ele alırdınız?

- Bu, doğrudan tarayıcılar arası testle ilgilidir ve genellikle tüm genel yapılandırmalarınızın depolandığı ve kontrol edildiği yapılandırma dosyanızla işlenir.
- Bu nedenle, yürütme başlamadan önce sürücümü bir yöntemle ayarladığımı söyleyebilirsiniz ve bu yöntem yapılandırma dosyamdan gerekli tüm yetenekleri alır ve sürücüyü Chrome, Firefox, Edge, Safari vb. olarak ayarlayabilir.
- Bununla, testin yürütülmesi için istenen tarayıcıyı, ortamı ve diğer bazı yetenekleri kurar.