



SELENIUM

Software Testing Nedir?

- EXPECTED RESULT (beklenen sonucun), ACTUAL RESULT (gerçek sonuca) eşit olup olmadığını kontrol etme işlemidir.
 - Eğer Expected result = Actual result, durum **PASS (test başarılı)**
 - Eğer Expected result \neq Actual result, durum **FAIL (test başarısız)**
- Sonuç olarak, olması gereken şeylerin olmadığını veya olmaması gereken şeylerin olduğunu denemek ve ortaya çıkartmak yazılım testinin amacı olmalıdır.

- Positive and Negative
- Test müşteri/işletme ihtiyaçlarını karşılamak için yapılır.
- Bir uygulamayı test etmek için belirli user storyler (kullanıcı hikayeleri) veya acceptance criteria (kabul kriterleri) vardır.

Testing Neden Önemlidir?

- İnsanlar hata yaparlar, bu hatalar kodda, yazılımda, sistemde ya da dokümanda defect (Kusur) oluşturur.
 - Defect olan kod çalıştırıldığında sistem beklenen fonksiyonları gerçekleştiremez ve başarısız olur.
- Bu sebeplerden dolayı;
- Müşteriye sunulmadan önce ürün kalitesinden emin olmak,
 - Yeniden çalışma (düzeltme) ve geliştirme masraflarını azaltmak,
 - Geliştirme işleminin erken aşamalarında yanlışları saptayarak ileri aşamalara yayılmasını önlemek, böylece zaman ve maliyetten tasarruf sağlamak amaçlarıyla ürün müşteriye sunulmadan önce test edilmesi gerekmektedir.
- Manual(functional=işlevsel) testing nedir?

- Manuel test, uygulamayı herhangi bir otomasyon aracı olmadan manuel olarak test etmektir.
- Manuel test kullanıcıları dokümantasyon için sınırlı teknoloji (Excel vb.) kullanır, ancak otomasyon araçları veya dili kullanmazlar.
- Manuel testte insan hatası olabilir.
- Tüm Otomasyon Tester'lar, herhangi bir otomasyon yapmadan önce uygulamayı anlamak için mutlaka manuel test yapmalıdır.
- İyi bir otomasyon tester aynı zamanda iyi bir manuel testerdir.

Test Automation (Test Otomasyonu) nedir?

- Bir sistemi bir otomasyon aracı yardımıyla test etmeye 'Test Otomasyonu' denir.
- Giderek daha popüler hale gelmektedir.

Başarılı Bir Otomasyon Testi için Yapılması Gerekenler

- Doğru otomasyon araçlarını(tools) seçin. Genellikle şirketlerin seçtikleri araçları vardır.
- Uygulamanız hakkında iyi bilgi sahibi olun.
- Test senaryolarınızı (test cases) kısa ve bağımsız tutun.
- Test otomasyonlarınızı önem derecesine göre sıralayın.
- Otomasyondan önce test verilerini hazırlayın(id,url,environment)
- Gerekirse, test caselerinizi yönetin(manage) ve bakımını yapın(maintain). Yeni test caseleri eski otomasyon test scriptlerini kırmamalıdır.
- Her zaman test ekibinizle, özellikle team liderleri ile iyi iletişim kurun.

Başarılı Bir Otomasyon Testi için Yapılmaması Gerekenler

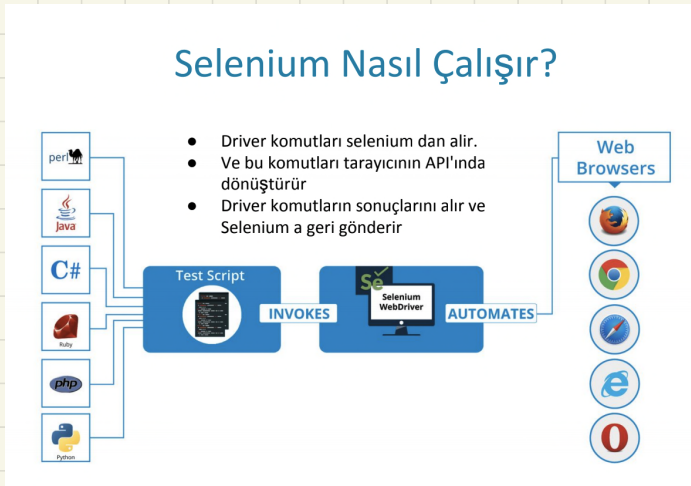
- Her şeyi otomasyonlaştırmaya çalışmayın. Ekibiniz % 100 otomasyon gerektiğini söyleyebilir. Ancak tüm uygulama için % 100 otomasyon mümkün değildir..
- Otomasyona Hemen Başlamayın. Otomasyona başlamadan önce aplikasyon hakkında detaylı bilgi sahibi olun.
- SADECE otomasyon araçlarına güvenmeyin. Zaman zaman manuel müdahaleler gerekebilir.
- İşe başladığınız ilk günden itibaren sorularınızı sormaktan çekinmeyin. Sonuçta, hatasız ürünlerden siz sorumlusunuz.

Selenium Nedir?

- Selenium, web uygulamalarını farklı tarayıcılarda ve platformlarda test etmek için ücretsiz (açık kaynaklı) bir araçtır.
- Selenium yalnızca web tabanlı uygulamaları otomasyon yapmaya odaklanır. Mobil ve Windows testi yapmak için eklentiler selenium'a eklenebilir .
- Selenium sadece bir jar dosyasıdır. Kurulum sırasında jar dosyalarını gördünüz.
- Selenium, otomasyon yapmak için kendi sınıflarına ve yöntemlerine sahip bir kütüphanedir..
- 2016'da piyasaya sürülen Selenium 3'ü öğreneceğiz
- Selenium'u çeşitli programlama dilleri ile yazabilirsiniz Python, java, ruby, .Net vb.

Selenium'un dört bileşeni vardır;

- Selenium Integrated Development Environment (IDE) (Selenium Entegre Geliştirme Ortamı (IDE))
- Selenium Remote Control (RC)(Selenium Uzaktan Kumanda (RC))
- WebDriver// (Biz Selenium WebDriver kullanacağız)
- Selenium Grid// (paralel test için kullanılıyor)



Selenium'un Avantajları nelerdir?

- Ücretsizdir. Open source
- Bir çok programlama dilini destekler (Java, Python, PHP, C#, Ruby vs.)
- Çoklu işletim sistemleriyle çalışır. (Windows, MacOS, Linux)
//Multiple operating systems
- Birden çok tarayıcı ile çalışır. (IE, Safari, Chrome, Firefox vs.)
// Multiple browsers

Selenium'un Dezavantajları nelerdir?

- Programlama bilgisi gerektirir
- Yalnızca web tabanlı uygulamaları test eder
- Profesyonel desteğe sahip değil
- Yapamadıkları:
 - performans testi
 - handle captcha (neredeyse tüm diğer otomasyon araçları gibi)

Tarayıcı üzerinden bir site'ye gidebilmek için ;

```
System.setProperty("webdriver.chrome.driver", "Users/cevdettellioglu/Documents/selenium libraries/drivers/chromedriver");
```

```
WebDriver driver = new ChromeDriver(); //driver object I olusturduk
```

Daha sonra **get** methodu kullanarak istediğimiz adrese gidiyoruz.

```
driver.get("https://www.google.com/");
```

```
Thread.sleep(5000);
```



Sol tarafta bulunan method kendinden önce çalışan kodu parametrede verilen süre boyunca uyutur ve süre bittikten sonra alt satırdaki kod çalışır

- **Navigations:** `navigate().to("url");` ya da `back()`, `forward()`, `refresh()`

- **Title or URL Verification of a page**

```
driver.getTitle();
```

```
driver.getCurrentUrl();
```

- **Farklı tarayıcılarla Test**

- `WebDriver driver = new ChromeDriver();`
- `WebDriver driver = new FirefoxDriver();`
- `WebDriver driver = new InternetExplorerDriver();`

Webelement

```
▼<tbody>
  ▶<tr>...</tr>
  ▼<tr> == $0
    ▼<td>
      <input type="email" class="inputtext login_form_input_box" name="email" id="email" data-testid="royal_email">
    </td>
    ▼<td>
      <input type="password" class="inputtext login_form_input_box" name="pass" id="pass" data-testid="royal_pass">
    </td>
    ▼<td>
      ▶<label class="login_form_login_button uiButton uiButtonConfirm" id="loginbutton" for="u_0_b">...
    </label>
    </td>
  </tr>
  ▶<tr>...</tr>
</tbody>
```

- Web sayfasındaki her şeye webelement denir

- Button,
- Search box(arama kutusu),
- Text box(metin kutusu),
- Headers(başlıklar),
- Tables(tablolar)vb.

- Farklı türde WebElement tagları(etiketleri) vardır. =====EMAIL TEXT BOX=====

<html>,<body>,<form>,<label>,<input>,<a>,<input type="email" class="form-control" placeholder="Email" data-test="email">

- Otomasyon için unique(tek) web öğelerini(element) tanımlamak üzere HTML kodunu inceleyeceğiz(inspect).

- Web elementleri birlikte kullanıcı arayüzünde(UI) bir web sayfası oluştururlar.

LOCATORS(Konum Bulucular)

=====EMAIL TEXT BOX=====

<input type="email" class="form-control" placeholder="Email" data-test="email" name="session[email]" id="session_email">

Locator ile WebElement Bulmak ??

```
driver.findelement(By.id("session_email"));
```

```
driver.findelement(By.name("session[email]"));
```

```
driver.findelement(By.className("form-control"));
```

`driver.findElement(By.linkText("link text"));` ==>email text box için bunu kullanamam

`driver.findElement(By.partialLinkText("partial link text"));` ==>email text box için bunu kullanamam

`driver.findElement(By.tagName("input"));`

`driver.findElement(By.xpath("session_email"));` => birden fazla syntax var

`driver.findElement(By.cssSelector("session_email"));` => birden fazla syntax var

Locator'lara Göre Elementleri Bulma: 8 locator vardır

1. By.id => `driver.findElement(By.id(""));`

2. By.name => `driver.findElement(By.name(""));`

3. By.className => `driver.findElement(By.className(""));`

4. By.tagName => `driver.findElement(By.tagName(""));`

5. By.linkText => `driver.findElement(By.linkText(""));`

6. By.partialLinkText => `driver.findElement(By.partialLinkText(""));`

7. By.xpath => `driver.findElement(By.xpath(""));`

8. By.css => `driver.findElement(By.css(""));`

1. By.id() method

WebElement sessionEmail=driver.findElement(By.id("session_email"));

- Web öğesini tanımlamanın en popüler yolu id kullanmaktır.
- id en güvenli ve en hızlı locator seçeneği olarak kabul edilir ve her zaman birden çok locator arasında ilk öncelik olmalıdır.
- Eğer yanlış id locate edilirse; **NoSuchElementException** hatası oluşur.

2. By.name() method

WebElement passwordTextBox =driver.findElement(By.name("session[password]"));

- Name ve value unique ise bu metodu da kullanabilirsiniz.

3. By.className() method

WebElement passwordTextBox=driver.findElement(By.className("form-control"));

- Class attribute'u olduğunda kullanılabilir.
- Class ve value unique ise, bu metodu da kullanabilirsiniz.

4. By.linkText() method

```
<a class="nav-item nav-link" data-test="addresses" href="/addresses">Addresses</a>
```

```
WebElement addressesLink = driver.findElement(By.linkText("Addresses"));
```

- Bu yalnızca HTML bağlantılarını(link) tanımlamak için kullanılabilir.
- HTML link elementleri, bir web sayfasında bağlantı etiketi(tag) kısaltması olan <a> etiketi(tag) kullanılarak temsil edilir.
- Kullanıcı arayüzündeki(UI) hyperlinkleri kolayca tanıyabilir ve sonra bu yöntemi kullanabilirsiniz
- Büyük / küçük harfe duyarlıdır(case sensitive) ve bağlantı(link) metniyle eşleşmelidir

5. By.partialLinkText() method

```
<a class="nav-item nav-link" data-test="addresses" href="/addresses">Addresses</a>
```

```
WebElement addressesLink =driver.findElement(By.partialLinkText("dresse"));
```

- linkText () yöntemine benzer.
- Tek fark, tam metin vermek zorunda kalmamanızdır.
- Metnin yalnızca belirli bir bölümünü verebilirsiniz.
- Metnin tamamını verdiğinizde de kabul eder.

6. By.tagName() method

```
<input class="form-control" placeholder="Password" data-test="password" type="password" name="session[password]" id="session_password">
```

```
WebElement passwordBox=driver.findElement(By.tagName("input"));
```

Bu, diğer konum belirleyicilerden biraz farklıdır.

<div>, <a>, <input>, ... gibi belirli bir etiketi ilettiğinizde, birden fazla aynı ad etiketine sahip olabileceğiniz için birden çok öğeyi döndürür. Çoğunlukla öğelerin bir listesini almak için kullanılır. Bu nedenle findElements() yöntemiyle kullanılması önerilir.

Örneğin, kullandığımız bir sayfadaki tüm bağlantıları döndürmek için:

```
List<WebElement> linksOnThePage=driver.findElements(By.tagName("a"));
```

findElement() method

```
WebElement elementName=driver.findElement(By.LocatorStrategy("LocatorValue"));
```

Bir elementi bulmak için findElement () yöntemini kullanınız.

Bu, tek bir web elementini döndürür.

Driver elementi bulamazsa, runtime exception verir : NoSuchElementException.

NoSuchElementException'ı gördüğünüzde, locatorı tekrar kontrol etmelisiniz.

findElements() method

`List<WebElement> elementName = driver.findElements(By.LocatorStrategy("LocatorValue"));`

Web elementlerinin bir listesini döndürür

Locator stratejisiyle eşleşen web elementi yoksa boş bir liste döndürür.

`NoSuchElementException` hatası vermez.

Her Web elementi, tıpkı array'deki gibi 0'dan başlayan bir sayı indeksi alır.

WebElement Üzerindeki İşlemler

Bir WebElement üzerinde eylemler gerçekleştirmek otomasyon tester'ları için çok önemlidir.

`sendKeys("yazdığımız metin");` => metin kutusuna yazmak için kullanılır.

`clear();` => Metin kutusunun içindekileri temizler.

`submit();` => `click();` bir form veya form içindeki bir öğeyi gönderir. Genellikle form veya arama kutusundaki metni göndermek için `sendKeys()` yönteminden sonra kullanılır.

Parent-Child-Sibling Terimleri Nedir?

```
▼ <tbody>
  ▶ <tr>...</tr>
  ▼ <tr> == $0
    ▼ <td>
      <input type="email" class="inputtext login_form_input_box" name="email" id="email" data-testid="royal_email">
      </td>
      ▼ <td>
        <input type="password" class="inputtext login_form_input_box" name="pass" id="pass" data-testid="royal_pass">
        </td>
        ▼ <td>
          ▶ <label class="login_form_login_button uiButton uiButtonConfirm" id="loginbutton" for="u_0_b">...
          </label>
          </td>
        </td>
      </tr>
    ▶ <tr>...</tr>
  </tbody>
```

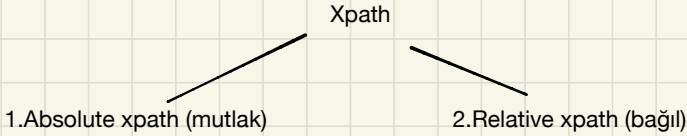
Tags: tbody, tr, td, input, label => parent-child-sibling relationship(Ebeveyn-çocuk-kardeş ilişkisi) hakkında konuştuğumuzda, yalnızca tag adları önemlidir.

Attributes: type, class, name, id, etc.

tr is child of tbody, and tbody is the parent of tr => "tr", "tbody" nin çocuğudur ve "tbody", "tr" in ebeveynidir

td is the child of tr, and td has 3 siblings => "td", "tr" in çocuğudur ve "td" nin 3 kardeşi vardır

By.xpath() method



1.Absolute xpath: Parent'tan (root element) child a tek tek gidilir. Yaygın olarak

kullanılmaz. '/' kullanılıyor

Örnek

- Xpath = //parent/child/child/child/... (Absolute xpath = //body/div/header/div/div/div/div)

By.xpath() method

2.Relative xpath: Yaygın olarak kullanılır. '/' kullanarak belirli bir elemente gidebilirsiniz.Xpath yazmanın birden çok yolu vardır.

- //tagName[@attribute='value']; => önemli!
- Relative xpath = //div[@id='logo'] or //div[@class='flex-col logo']

Ders öncesi sorular

1.Kac tane locator vardır? isimlerini yazınız.

8 tane. Id, name, className, tagName, linkText, PartialLinkText, xpath, cssSelector

2.Kac cesit xpath vardır? isimlerini yazınız. Hatırladığınız farklarını yazınız.

1.Absolute xpath 2. Relative xpath

Absolute ->Parenttan childa doğru yazılır. / kullanılır. Yaygın değildir.

Relative -> İstenilen yerden başlanabilir. // kullanılır. En yaygın kullanım şeklidir.

3. Relative xpath'in söz dizimini (syntax) yazınız.

//tagName[@attribute= 'value']

By.xpath() method-Relative xpath

- **//tagName[@attribute='value']; => önemli!**

- EXACT TEXT(Belirli bir text) ile element bulma:

- **//tagName[.='text name']** ⇒ Belirli bir tagname , herhangi bir attribute, belirli bir text
- **//*[.='text name']** ⇒ Herhangi bir tag, belirli bir text.
- **//*[(text()='exact text with extra space and all')]** ⇒ Herhangi bir tag, belirli bir text

- Belirli bir metni içeren bir öğeyi bulmak için şunları kullanabiliriz:

- **//*[contains(text(),'piece of text')];**

- Birden fazla attribute yazabiliriz

- **//tag[@attribute1='value 1' or attribute2='value2']**

Example: `//div[@id='logo' or class='flex-col logo']`

2.Relative xpath:

Relative xpath'i bulurken, ortadaki herhangi bir yerden başlayabiliriz.

Yani root elementten başlamak zorunda değiliz.

// Kullanırız .

1. **//tag[@attribute = 'value'];** => belirli tag, belirli attribute, belirli attribute value '' içinde olmalı, "" değil

```
<input type="email" class="form-control" placeholder="Email" data-test="email" name="session[email]" id="session_email">
```

ornek: `//input[@type='email'];`
`//input[@class='form-control']`

```
<span class="navbar-text" data-test="current-user">testtechproed@gmail.com</span>
```

2. **//tagName[.='text name'];** => belirli bir tag, herhangi bir attribute, belirli bir text

ornek: `//span[.='testtechproed@gmail.com'];`

3. **//*[.='text name'];** => herhangi bir tag, herhangi bir attribute, belirli bir text

ornek: `//*[.='testtechproed@gmail.com'];`

4. **//*[(text()='text name')];**

ornek: `//*[(text()='testtechproed@gmail.com')];`

5. **//*[contains(text(),'text name parçası')];**

ornek: `//*[contains(text(),'testtechproed@gmail')]`

6. **//tag[@attribute1='value 1' or attribute2='value2']**

```
<input type="email" class="form-control" placeholder="Email" data-test="email" name="session[email]" id="session_email">
```

ornek: `//input[@type='email' or name='session[email]'];`

By.cssSelector() method

Css selector xpath'e benzer. Üç ana tip kullanılır

```
<input type="email" class="form-control" placeholder="Email" data-test="email" name="session[email]" id="session_email">
```

- **css = "tagName[attribute name='value']"**

- `driver.findElement(By.cssSelector("input[name='session[email]']"));`

- **css="tagName#id value" or just css="#id value"=>yalnızca id value ile çalışır**

- `driver.findElement(By.cssSelector("input#session_email"));`

- **css="tagName.class value" or just css=".class value"=>yalnızca class value ile çalışır**

- `driver.findElement(By.cssSelector("input.form-control"));`

CSS SELECTOR

Email text box locate ediniz.

```
<input type="email" class="form-control" placeholder="Email" data-test="email" name="session[email]" id="session_email">
```

1.css = "tagName[attribute name='value']"

ornek: `driver.findElement(By.cssSelector("input[type='email']"));`

2. css="tagName#id value" VEYA css="#id value"=>yalnızca id value ile çalışır

ornek: `driver.findElement(By.cssSelector("input#session_email"));` VEYA `"#session_email"`

3. css="tagName.class value" VEYA css=".class value"=>yalnızca class value ile çalışır

ornek: `driver.findElement(By.cssSelector("input.form-control"));` VEYA `".form-control"`
