**Java'da Selenium WebDriver ile Çerezleri Kullanma**

**1. Cookies (Çerezler)**

Bir HTTP çerezi, kullanıcı ve tercihleri hakkındaki bilgilerden oluşur. Bir anahtar/değer çifti kullanarak bilgileri depolar. Kullanıcı o web sitesinde gezinirken Web Uygulamasından gönderilen ve Web Tarayıcısında depolanan küçük bir veri parçasıdır.

Çerezler, belirli kullanıcıları tanımlamak ve bu kullanıcıların göz atma deneyimini iyileştirmek için kullanıcının bilgisayarı ile web sunucusu arasında takas edilen, kullanıcı adı ve parola gibi küçük veri parçalarını içeren dosyalardır.

İnternette gezinirken ziyaret ettiğiniz web sayfaları, bilgisayarınıza ve telefonunuza küçük bilgi dosyaları kaydeder. Bu dosyalar telefon veya bilgisayarınızın hafızasında saklanır. Daha sonra aynı siteleri ziyaret ettiğinizde bu kayıtlı bilgi dosyaları sayesinde siteler sizi tanıyabilir. Bilgileriniz bu dosyalara yazıldığından dolayı tekrar aynı web sayfalarını ziyaret ettiğinizde bilgilerinizi yeniden girmeye gerek duymazsınız. Cookies, kişisel bilgiler de dahil olmak üzere birçok bilgiyi içerebilir. Web siteleri, ancak sizin izin verdiğiniz bilgilere erişebilir. Bu web sayfaları, sizin vermediğiniz bilgilere erişemez ya da bilgisayarınızdaki diğer dosyaları görüntüleyemez.

Birkaç farklılıkla, siber dünyadaki çerezlerin oturum çerezi ve kalıcı çerez olmak üzere iki çeşidi vardır. Oturum çerezleri yalnızca bir web sitesinde gezinirken kullanılır. Bunlar rastgele erişimli bellekte saklanır ve hiçbir zaman sabit sürücüye yazılmaz. Oturum sona erdiğinde oturum çerezleri otomatik olarak silinir. Kalıcı çerezler bir bilgisayarda sonsuza kadar kalır ancak birçoğunun bir son kullanma tarihi olup bu tarihe gelindiğinde otomatik olarak kaldırılırlar. Üçüncü taraf çerezler daha sıkıntılıdır. Bunlar, genellikle kullanıcıların halihazırda gezindiği web sayfalarındaki reklamlarla bağlantılı olduklarından bu sayfalardan farklı web siteleri tarafından oluşturulur.

**2. Çerezlerle Çalışmak**

Çerezleri manipüle etmek için günlük kullanım durumu, testler arasında oturumumuzu sürdürmektir.

Daha da basit bir senaryo, arka ucumuzun çerezleri doğru şekilde ayarlayıp ayarlamadığını test etmek istediğimiz zamandır.

Selenium ile cookies otomasyonu yapabiliriz.

Driver.manage( ). method’u ile cookie’leri

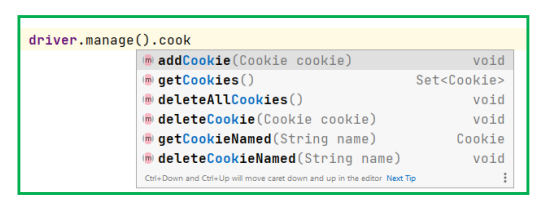
- listeleyebilir.

- Isim ile cagirabilir

- Yeni cookie ekleyebilir

- Var olanlari ismi silebilir

- Var olan tum cookie’leri silebiliriz



**2.1. Kurmak**

Projemize Selenium-Java bağımlılığını eklememiz gerekecek :

<dependency>

<groupId>org.seleniumhq.selenium</groupId>

<artifactId>selenium-java</artifactId>

<version>3.14.0</version>

</dependency>

**2.2. Çerezleri Okumak**

Ardından, bir web sayfasına gittikten sonra sürücümüzde çerezlerin bulunduğunu doğrulamak için basit bir test uygulamalıyız:

@Test  
public void whenNavigate\_thenCookiesExist() {  
 driver.navigate().to(navUrl);  
 Set<Cookie> cookies = driver.manage().getCookies();  
  
 assertThat(cookies, is(not(empty())));  
}

**2.3. Çerez Özellikleri**

Bir çerez bir alan adı ile ilişkilendirilebilir, bir son kullanma tarihine sahip olabilir ve çok daha fazlası.

Bazı yaygın çerez özelliklerine bir göz atalım:

@Test  
public void whenNavigate\_thenLpCookieHasCorrectProps() {  
 driver.navigate().to(navUrl);  
 Cookie lpCookie = driver.manage().getCookieNamed("lp\_120073");  
  
 assertThat(lpCookie.getDomain(), equalTo(".amazon.com"));  
 assertThat(lpCookie.getPath(), equalTo("/"));  
 assertThat(lpCookie.getExpiry(), is(not(nullValue())));  
 assertThat(lpCookie.isSecure(), equalTo(false));  
 assertThat(lpCookie.isHttpOnly(), equalTo(false));  
}

**2.4. Çerez Ekleme**

Çerez eklemek basit bir işlemdir.

Tanımlama bilgisini oluşturup, addCookie yöntemini kullanarak sürücüye ekliyoruz :

@Test  
public void whenAddingCookie\_thenItIsPresent() {  
 driver.navigate().to(navUrl);  
 Cookie cookie = new Cookie("foo", "bar");  
 driver.manage().addCookie(cookie);  
 Cookie driverCookie = driver.manage().getCookieNamed("foo");  
  
 assertThat(driverCookie.getValue(), equalTo("bar"));  
}

**2.5. Çerezleri Silme**

Beklediğimiz gibi, bir çerezi de deleteCookie yöntemini kullanarak silebiliriz:

@Test  
public void whenDeletingCookie\_thenItIsAbsent() {  
 driver.navigate().to(navUrl);  
 Cookie lpCookie = driver.manage().getCookieNamed("lp\_120073");  
  
 assertThat(lpCookie, is(not(nullValue())));  
  
 driver.manage().deleteCookie(lpCookie);  
 Cookie deletedCookie = driver.manage().getCookieNamed("lp\_120073");  
  
 assertThat(deletedCookie, is(nullValue()));  
}

**2.6. Çerezleri Geçersiz Kılma**

Bir tanımlama bilgisini geçersiz kılmak için açık bir yöntem olmamasına rağmen, basit bir yolu vardır.

Tanımlama bilgisini silebilir ve aynı ada ancak farklı bir değere sahip yeni bir tanımlama bilgisi ekleyebiliriz:

@Test  
public void whenOverridingCookie\_thenItIsUpdated() {  
 driver.navigate().to(navUrl);  
 Cookie lpCookie = driver.manage().getCookieNamed("lp\_120073");  
 driver.manage().deleteCookie(lpCookie);  
  
 Cookie newLpCookie = new Cookie("lp\_120073", "foo");  
 driver.manage().addCookie(newLpCookie);  
  
 Cookie overriddenCookie = driver.manage().getCookieNamed("lp\_120073");  
  
 assertThat(overriddenCookie.getValue(), equalTo("foo"));  
}

* **Çerezler için Selenyum Sorgu Komutları:**

Selenium Webdriver'da, aşağıdaki yerleşik yöntemle çerezleri sorgulayabilir ve bunlarla etkileşim kurabiliriz:

driver.manage().getCookies(); // Tüm Çerezlerin Listesini Döndür  
driver.manage().getCookieNamed(arg0); //Ada göre belirli çerezi döndür  
driver.manage().addCookie(arg0); //Çerez oluştur ve ekle  
driver.manage().deleteCookie(arg0); // Belirli çerezi sil  
driver.manage().deleteCookieNamed(arg0); // Adına göre belirli çerezi sil  
driver.manage().deleteAllCookies(); // Tüm çerezleri sil

* **Çerezleri Neden Selenyumda Kullanmalısınız?**

Her çerez bir ad, değer, etki alanı, yol, sona erme ve güvenli olup olmadığı durumuyla ilişkilendirilir. Bir sunucu, bir istemciyi doğrulamak için bu değerlerin tümünü bir tanımlama bilgisinde ayrıştırır.

Selenium web sürücüsünü kullanarak bir web uygulamasını test ederken, bir tanımlama bilgisi oluşturmanız, güncellemeniz veya silmeniz gerekebilir.

Örneğin, Online Alışveriş Uygulamasını otomatikleştirirken, sipariş verme, Sepeti Görüntüle, Ödeme Bilgileri, sipariş onayı vb. gibi test senaryolarını otomatikleştirmeniz gerekir.

Çerezler saklanmazsa, yukarıda listelenen test senaryolarını çalıştırmadan önce her seferinde oturum açma eylemi gerçekleştirmeniz gerekecektir. Bu, kodlama çabanızı ve yürütme sürenizi artıracaktır.

Çözüm, çerezleri bir Dosyada saklamaktır. Daha sonra, bu dosyadan çerez değerlerini alın ve mevcut tarayıcı oturumunuzu ekleyin. Sonuç olarak, sürücü oturumunuzda bu bilgiler bulunduğundan, her Test Vakasında oturum açma adımlarını atlayabilirsiniz.

Uygulama sunucusu artık tarayıcı oturumunuzu kimliği doğrulanmış olarak ele alır ve sizi doğrudan istediğiniz URL'ye götürür.

* **Bu bilgiler doğrultusunda örnek bir çalışma yapalım:**

package Cookies;  
  
import org.junit.Assert;  
import org.junit.Test;  
import org.openqa.selenium.Cookie;  
import utilities.TestBaseBeforeAfter;  
  
import java.util.Set;  
  
public class C01\_Cookie extends TestBaseBeforeAfter {  
  
 @Test  
 public void cookiesTesti() {  
  
 //Yeni bir class olusturun : cookiesAutomation  
 //1- Amazon anasayfaya gidin  
 driver.get("https://amazon.com");  
  
 //2- tum cookie’leri listeleyin  
 Set<Cookie> cookieSet = driver.manage().getCookies();  
 System.out.println(cookieSet);  
 int sayac =1;  
 for (Cookie each:cookieSet) {  
 System.out.println(sayac+".ci "+each);  
 System.out.println("name " + each.getName());  
 System.out.println("value " + each.getValue());  
 sayac++;  
 }

//3- Sayfadaki cookies sayisinin 5’den buyuk oldugunu test edin  
 int cookieSayisi = cookieSet.size(); //önce size ile sayısını aldık  
 Assert.assertTrue(cookieSayisi>5);

//4- ismi i18n-prefs olan cookie degerinin USD oldugunu test edin  
 for (Cookie each:cookieSet) {  
 if (each.getName().equals("i18n-prefs")){  
 Assert.assertEquals("USD",each.getValue());  
 }  
 }  
  
 //5- ismi “en sevdigim cookie” ve degeri “cikolatali” olan bir cookie  
 //olusturun ve sayfaya ekleyin  
 Cookie cookie = new Cookie("en sevdigim cookie","cikolatali"); //Burada biz cookie'i oluşturduk  
 driver.manage().addCookie(cookie); //Oluşturduktan sonra ekleme yaptık  
  
 cookieSet = driver.manage().getCookies(); //Yeniden bütün cookieler sete atanmalıdır yoksa yazdırmaz  
 sayac =1;  
 for (Cookie each:cookieSet) {  
 System.out.println(sayac+".ci "+each);  
 System.out.println("name " + each.getName());  
 System.out.println("value " + each.getValue());  
 sayac++;  
 }  
  
 //6- eklediginiz cookie’nin sayfaya eklendigini test edin  
 Assert.assertTrue(cookieSet.contains(cookie));  
  
 //7- ismi skin olan cookie’yi silin ve silindigini test edin  
 driver.manage().deleteCookieNamed("skin");  
 Assert.assertFalse(cookieSet.contains("skin"));  
  
 //8- tum cookie’leri silin ve silindigini test edin  
 driver.manage().deleteAllCookies();  
  
 //silindiğini test edebilmek için yine bir set e atarız  
 cookieSet = driver.manage().getCookies();  
 Assert.assertTrue(cookieSet.isEmpty()); // isEmpty() boş demektir  
  
  
 }  
}

**Kaynak:**

* <https://www.selenium.dev/documentation/webdriver/interactions/cookies/>
* <https://www.baeldung.com/java-selenium-webdriver-cookies>
* <https://www.guru99.com/handling-cookies-selenium-webdriver.html>
* <https://techproeducation.com/>