



ebebek Java & QA & SAP Spartacus Practicum
Test Otomasyon Ödevi

Yiğit Arda YILMAZ

24.11.2022

İçindekiler

1.Giriş	3
2.Manuel Test Senaryosu	3
2.1 Bilgilendirme	3
2.2 Senaryo	3
3.Test Otomasyon (UI)	3
3.1 Amaç	3
3.2 Genel Bakış	3
3.3 Test Kapsamı	3
3.4 Uygulanan Test	3
3.5 Test Ortamı ve Araçları	4
3.6 Yazılım	4
3.6.1 FirstTest Class	4
3.6.2 Log Class	5
4. Sonuç	6

1.Giriş

Bu rapor Patika.dev aracılığı ile düzenlenmiş olan “ ebebek Java & QA & SAP Spartacus Practicum” isimli eğitim sürecinin bir parçasıdır. Ve ilgili şirket ebebek tarafından istenilen Test Otomasyon ödevi için hazırlanmıştır.

2.Manuel Test Senaryosu

2.1 Bilgilendirme

Raporun bu bölümünde manuel olarak bir test senaryosu hazırlanmıştır.

2.2 Senaryo

1. Ana sayfa açılır. (ebebek.com)
2. Hesabım ikonuna tıklanır.
3. Giriş yap ikonuna tıklanır, giriş yapılır.(önceden tanımlanmış demo hesap ile)
4. Hesabım ikonuna tıklanır.
5. Çıkış yap ikonuna tıklanır.
6. Hesabım ikonuna tıklanır.
7. Giriş yapılır.(hatalı kullanıcı ile)

3.Test Otomasyon (UI)

3.1 Amaç

Raporun bu kısmında ebebek şirketinin sitesi olan “ebebek.com” üzerinden test otomasyon uygulaması gerçekleştirilmiştir. Gerçekleştirilen çeşitli adımlar raporun devamında detaylıca açıklanmıştır

3.2 Genel Bakış

“ebebek.com” içerisinde anne-bebek ürünlerini barındıran çevrimiçi alışveriş sitesidir. Bu alışveriş sitesinin içerisinde birçok kategori vardır. Ayrıca sitede üyelik servisi de bulunmaktadır. Üye hesapları vasıtasıyla ürünleri favorilerimize ekleyip değerlendirebiliyoruz. Ürünlerimizi sepete ekleyip siparişlerimizi görebiliyoruz. Bizimde testimiz tam bu noktada başlamaktadır.

3.3 Test Kapsamı

Kapsamında : Aşağıdaki modüller için Test uygulanmıştır.

- İçerik Araması
- Sepete ekleme
- Sepeti Görme
- Sepeti Kontrol Etme

Kapsam dışında: Üyelik olma, giriş yapma, kullanıcı doğrulama, performans testi gibi modüller uygulanmamıştır.

3.4 Uygulanan Test





Test sırasında aşağıdaki adımlar sırası ile uygulanmaktadır.

1. Ana sayfa açılır. (ebebek.com)
2. Arama çubuğuna “biberon” yazılır.

3. Arama yapılır.
4. İlk ürüne tıklanır.
5. “Sepete Ekle” butonuna tıklanır.
6. “Sepeti Gör” butonuna tıklanır.
7. Ürünün sepete eklendiği kontrol edilir.

3.5 Test Ortamı ve Araçları

Tablo 1 Test Ortamı ve Araçları

Araçlar	Ortamlar
IDE	Eclipse IDE 
Yazılım dili	Java 
Kütüphaneler	Selenium  junit 

3.6 Yazılım

Yazılım kısmı iki classtan oluşmaktadır.

1. FirstTest.java
2. Log.java

3.6.1 FirstTest Class

İlk classımız testin gerçekleşti ana classtır. Bu classta 3 kısımdan oluşmaktadır. Bu kısımlar aşağıdaki gibidir.

- @Before
- @Test
- @After

@Before kısmında @Testten önce çalışacak setUp() kısmı bulunmaktadır. Şekil 1’de görüldüğü üzere sitenin url’si ve chromedriver tanımlanmıştır. Sonrasında ilgili siteye gidilmiştir. Gerekli loglar bırakılmıştır.

```
@Before
public void setUp() throws Exception {
    log.info("Testing Started");
    String url = "https://www.e-bebek.com/";
    driver = new ChromeDriver();
    driver.get(url);
    log.info("Homepage Opened");
}
```

Şekil 1 @Before Code

@Test kısmında asıl test işlemleri yapılmaktadır. Raporun “3.4 “başlığının altında belirtilen adımlar burada uygulanmaktadır. Şekil 2’de görüldüğü üzere gerekli tıklamalar yapılmış ve en sonun da try-catch ile sepete eklenen ürün gerçekten eklenmiş mi kontrol ediliyor.

```
@Test
public void ebebekTest() throws Exception {

    WebElement searchElement = driver.findElement(By.id("txtSearchBox"));

    searchElement.click();
    log.info("Clicked on search bar");
    searchElement.sendKeys("biberon");
    log.info("Typed in \"biberon\"");
    searchElement.sendKeys(Keys.ENTER);
    log.info("Word searched");

    WebDriverWait wait = new WebDriverWait(driver, Duration.ofSeconds(10));
    wait.until(ExpectedConditions
        .presenceOfElementLocated(By.xpath("//div/eb-product-list-item[1]/div/eb-generic-link/a")));

    WebElement firstProduct = driver.findElement(By.xpath("//div/eb-product-list-item[1]/div/eb-generic-link/a"));
    String productLink = firstProduct.getAttribute("href");

    driver.get(productLink);
    log.info("Clicked on first product");

    WebElement element = driver.findElement(By.id("addToCartBtn"));
    JavascriptExecutor executor = (JavascriptExecutor) driver;
    executor.executeScript("arguments[0].click();", element);
    log.info("Clicked on add to cart button");

    wait.until(ExpectedConditions.presenceOfElementLocated((By.id("btnShowCart"))));
    driver.findElement(By.id("btnShowCart")).click();
    log.info("Clicked show cart button");
    try {
        WebDriverWait cartListWait = new WebDriverWait(driver, Duration.ofSeconds(5));
        cartListWait.until(ExpectedConditions.presenceOfElementLocated((By.xpath("//div/eb-cart-item-list"))));
        log.info("Test Successful. The Product Has Been Added To The Cart");
    } catch (Exception e) {
        log.warn("Test Failed. The Product Not Found In Cart");
    }

}
```

Şekil 2 @Test Code

@After kısmında ise bütün test bittikten sonra çalışan kod bulunmaktadır. Şeki 3’te görüldüğü üzere gerekli log yazılıp çıkış işlemi yapılmıştır.

```
@After
public void tearDown() {
    log.info("Test Ended");
    driver.quit();
}
```

Şekil 3 @After Code

3.6.2 Log Class

Log class’ta adında anlaşılacağı üzere loglar için oluşturulmuştur. Şekil 4’te kodlar log class’ına ait kodlar gösterilmiştir.

```

1 package patikadev;
2
3 import java.util.logging.*;
4
5
6 public class Log {
7
8     static Logger logger = Logger.getLogger(Log.class.getName());
9
10
11     public void info (String message)
12     {
13
14         logger.info(message);
15
16     }
17
18
19     public void warn (String message)
20     {
21         logger.warning(message);
22
23     }
24 }

```

Şekil 4 Log Class

Aşağıdaki Şekil 5’te test sonrası log çıktısı verilmiştir.

```

<terminated> FirstTest (1) [JUnit] C:\Program Files\Java\jdk-19\bin\javaw.exe (23 Kas 2022 17:45:20 – 17:46:51) [pid: 14388]
Kas 23, 2022 5:45:23 BS patikadev.Log info
INFO: Testing Started
SLF4J: No SLF4J providers were found.
SLF4J: Defaulting to no-operation (NOP) logger implementation
SLF4J: See https://www.slf4j.org/codes.html#noProviders for further details.
Starting ChromeDriver 107.0.5304.62 (1eec40d3a5764881c92085aaee66d25075c159aa-refs/branch-heads/5304@{#942}) on port 44567
Only local connections are allowed.
Please see https://chromedriver.chromium.org/security-considerations for suggestions on keeping ChromeDriver safe.
ChromeDriver was started successfully.
Kas 23, 2022 5:46:10 BS patikadev.Log info
INFO: Homepage Opened
Kas 23, 2022 5:46:13 BS patikadev.Log info
INFO: Clicked on search bar
Kas 23, 2022 5:46:14 BS patikadev.Log info
INFO: Typed in "biberon"
Kas 23, 2022 5:46:14 BS patikadev.Log info
INFO: Word searched
Kas 23, 2022 5:46:41 BS patikadev.Log info
INFO: Clicked on first product
Kas 23, 2022 5:46:42 BS patikadev.Log info
INFO: Clicked on add to cart button
Kas 23, 2022 5:46:45 BS patikadev.Log info
INFO: Clicked show cart button
Kas 23, 2022 5:46:50 BS patikadev.Log info
INFO: Test Successful. The Product Has Been Added To The Cart
Kas 23, 2022 5:46:50 BS patikadev.Log info
INFO: Test Ended

```

Şekil 5 Log Çıktısı

4. Sonuç

Bu rapor sonucunda java ortamında Selenium ile bir test otomasyonu nasıl hazırlanır ve uygulanır öğrenilmiştir. Ayrıca otomasyon ve manuel testler arasında ki farklar daha net bir şekilde incelenmiştir. Otomasyon test çok hızlı olması avantajlarındandır. Yatırım getirisi uzun vadede daha yüksektir. Fakat manuel test ne kadar yavaş ve eski gibi görünse de gerçek kullanıcı deneyimini yansıtmaktadır