
TP 8

Selenium web driver

Selenium avec JUnit 5 :

Selenium est une bibliothèque permettant d'automatiser les navigateurs web. JUnit 5 est un framework moderne de test unitaire en Java. Leur combinaison est parfaite pour écrire des tests d'interface utilisateur (UI) automatisés.

Dépendances Maven :

```
<dependencies>

  <dependency>
    <groupId>org.seleniumhq.selenium</groupId>
    <artifactId>selenium-java</artifactId>
    <version>4.31.0</version>
  </dependency>

  <dependency>
    <groupId>io.github.bonigarcia</groupId>
    <artifactId>webdrivermanager</artifactId>
    <version>6.1.0</version>
  </dependency>

  <dependency>
    <groupId>org.junit.jupiter</groupId>
    <artifactId>junit-jupiter</artifactId>
    <version>5.10.0</version>
    <scope>test</scope>
  </dependency>

</dependencies>
```

Exemple :

Voici la correction du TP7, incluant un test JUnit 5 utilisant assertEquals, ainsi qu'une capture d'écran automatique en cas d'échec du test :

```
package TP8.TestSeleniumJuinit5;

import java.io.File;
```

```

import java.io.IOException;
import java.time.Duration;

import org.junit.jupiter.api.AfterEach;
import org.junit.jupiter.api.BeforeEach;
import org.junit.jupiter.api.Test;
import org.openqa.selenium.By;
import org.openqa.selenium.OutputType;
import org.openqa.selenium.TakesScreenshot;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.firefox.FirefoxDriver;
import org.openqa.selenium.io.FileHandler;
import org.openqa.selenium.support.ui.ExpectedConditions;
import org.openqa.selenium.support.ui.WebDriverWait;

import io.github.bonigarcia.wdm.WebDriverManager;

import static org.junit.jupiter.api.Assertions.assertEquals;

public class App
{
    private FirefoxDriver driver = new FirefoxDriver();

    @BeforeEach
    void setUp() {
        WebDriverManager.firefoxdriver().setup();
        driver = new FirefoxDriver();
        driver.get("http://live.techpanda.org/");
        driver.manage().window().maximize();
        driver.manage().timeouts().implicitlyWait(Duration.ofSeconds(10));
    }

    @AfterEach
    void tearDown() {
        driver.close();
    }

    @Test
    void test1() {

        driver.manage().timeouts().implicitlyWait(Duration.ofSeconds(10));

        WebElement mobileButton =
driver.findElement(By.xpath("/html/body/div/div/header/div/div[3]/nav/ol/li[1]/a")
);
        mobileButton.click();

        WebElement addToCardButton =
driver.findElement(By.xpath("/html/body/div/div/div[2]/div/div[2]/div[1]/div[3]/ul/li[1]/div/div[3]/button/span/span"));
        addToCardButton.click();

        WebElement qteEdit =
driver.findElement(By.xpath("/html/body/div/div/div[2]/div/div/div/form/table/tbody/tr/td[4]/input"));
        qteEdit.clear();
    }
}

```

```

qteEdit.sendKeys("1000");

WebDriverWait wait = new WebDriverWait(driver, Duration.ofSeconds(10));
WebElement updateButton =

wait.until(ExpectedConditions.visibilityOfElementLocated(By.xpath("/html/body/div/
div/div[2]/div/div/div/form/table/tbody/tr/td[4]/button/span/span"
)));

updateButton.click();

String message =
driver.findElement(By.xpath("/html/body/div/div/div[2]/div/div/div/form/table/tbod
y/tr/td[2]/p")).getText();

/* The maximum quantity allowed for purchase is 500.
String expectedMessage = "* The maximum quantity allowed for purchase is
500.";

if (!message.equals(expectedMessage)) {
    File srcFile = ((TakesScreenshot)
driver).getScreenshotAs(OutputType.FILE);

    // Enregistrer le fichier
    File destFile = new File("screenshot.png");
    try {
        FileHandler.copy(srcFile, destFile);
    } catch (IOException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
}

assertEquals(expectedMessage , message);
}
}

```

Exercice : Reprendre l'exercice de la séance précédente.

1. Accéder au site : <http://live.techpanda.org/>
2. Aller dans la section Mobile (clic sur le lien "MOBILE" du menu).
3. Dans la liste des mobiles, cliquer sur ADD TO CART pour le mobile Sony Xperia.
4. Dans le panier, modifier la quantité (QTY) successivement avec les valeurs suivantes : **vide, 0, 1, 500, 501.**
5. Cliquer sur le bouton UPDATE après chaque modification.
6. Vérifier le message affiché pour chaque cas afin de valider le comportement attendu (succès ou message d'erreur).

Pour chaque valeur de quantité testée, rédiger un **cas de test JUnit 5**.