

Lab 11 : Web Testing with Selenium

Usa Sammapun, Kasetsart University

I. ทดสอบการเพิ่มข้อมูล

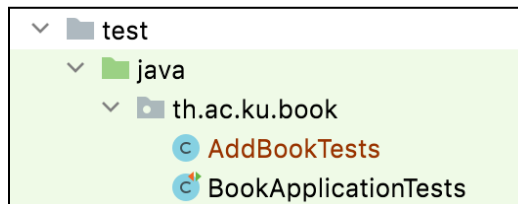
1. Fork จาก repo นี้ ซึ่งเป็น Spring Thymeleaf app เป็นเว็บไซต์ร้านหนังสือ wisdom book ที่สามารถเพิ่มหนังสือ แสดงรายการหนังสือ โดยเก็บข้อมูลใน H2 database

<https://github.com/ladyusa/wisdom-book>

2. Clone จาก repo ของนิสิต แล้วเปิดโปรเจค เพิ่ม selenium dependency ใน pom.xml ดังนี้

```
<dependency>
  <groupId>org.seleniumhq.selenium</groupId>
  <artifactId>selenium-java</artifactId>
  <version>3.141.59</version>
</dependency>
<dependency>
  <groupId>io.github.bonigarcia</groupId>
  <artifactId>webdrivermanager</artifactId>
  <version>5.3.0</version>
</dependency>
```

3. สร้างไฟล์ test ชื่อ AddBookTests โดยใส่ในโฟลเดอร์ test --> java ภายใต้ package th.ac.ku.book



4. ทดสอบโดยใช้ไลบรารีการทดสอบของ Spring Boot และกำหนดให้ใช้ random port

```
@SpringBootTest(webEnvironment = SpringBootTest.WebEnvironment.RANDOM_PORT)
public class AddBookTests {

    @LocalServerPort
    private Integer port;

}
```

5. Setup การทดสอบ โดยการ

a. BeforeAll ก่อนการรันเทสเคส

- i. กำหนดให้ Spring Boot ใช้ random port
- ii. สร้าง webdriver สำหรับ Chrome Browser (สามารถเปลี่ยนเป็น browser อื่นได้ เช่น Firefox, Edge, Opera)
- iii. สร้าง webdriver wait เพื่อรอจนกว่าเว็บจะโหลดเสร็จ ก่อนจะเริ่มทดสอบ (สามารถปรับได้ตามความเร็วการโหลด browser ของเครื่อง)

b. BeforeEach สำหรับแต่ละเทสเคส

- i. ไปที่หน้าที่ต้องการทดสอบ โดยเรียกใช้ driver.get(url)

c. AfterEach สำหรับแต่ละเทสเคส

- i. ให้เทสหยุดการทำงานแป๊บหนึ่ง เราจะได้เห็นผลการทดสอบ ก่อนที่ selenium จะไปทดสอบในเทสเคสถัดไป

d. AfterAll เมื่อทุกเทสเคสรันเสร็จเรียบร้อยแล้ว

- i. ให้ปิดโปรแกรม browser

```
package th.ac.ku.book;

import io.github.bonigarcia.wdm.WebDriverManager;
import org.junit.jupiter.api.AfterAll;
import org.junit.jupiter.api.AfterEach;
import org.junit.jupiter.api.BeforeAll;
import org.junit.jupiter.api.BeforeEach;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;
import org.openqa.selenium.support.ui.WebDriverWait;
import org.springframework.boot.test.context.SpringBootTest;
import org.springframework.boot.web.server.LocalServerPort;

@SpringBootTest(webEnvironment = SpringBootTest.WebEnvironment.RANDOM_PORT)
public class AddBookTests {

    @LocalServerPort
    private Integer port;

    private static WebDriver driver;
    private static WebDriverWait wait;

    @BeforeAll
    public static void beforeAll() {
        WebDriverManager.chromedriver().setup();
        driver = new ChromeDriver();
        wait = new WebDriverWait(driver, 1000);
    }

    @BeforeEach
    public void beforeEach() {
        driver.get("http://localhost:" + port + "/book/add");
    }

    @AfterEach
    public void afterEach() throws InterruptedException {
```

```
        Thread.sleep(3000);  
    }  
  
    @AfterAll  
    public static void afterAll() {  
        driver.quit();  
    }  
}
```

6. ทดสอบการเพิ่มหนังสือ

- a. ทำความเข้าใจการทำงานของ Selenium ก่อนนิดหน่อย ว่ากรอกข้อมูลอัตโนมัติอย่างไร
 - i. หา WebElement สำหรับ input field ในการกรอกข้อมูล โดยหา id ใน .html
 - ii. กรอกข้อมูลใน input field ที่ต้องการ

```
@Test
void testAddBook() {
    WebElement nameField = wait.until(webDriver ->
        webDriver.findElement(By.id("nameInput")));
    WebElement authorField = driver.findElement(By.id("authorInput"));
    WebElement priceField = driver.findElement(By.id("priceInput"));

    nameField.sendKeys("Clean Code");
    authorField.sendKeys("Robert Martin");
    priceField.sendKeys("600");
}
```

- b. ลองรันเทส จะเห็นว่า (1) มีการรัน server ก่อน (2) จากนั้น จะเปิด Chrome (3) ไปที่หน้า add book และกรอกข้อมูล name, author และ price ตามที่ได้ระบุ
- c. เขียนเทสให้สมบูรณ์ โดยให้กดปุ่ม submit และหลัง submit ให้ assert ว่า ได้หนังสือที่เพิ่มไปหรือไม่ โดยดูจาก td ว่ามีชื่อหนังสือหรือไม่

```
@Test
void testAddBook() {
    WebElement nameField = wait.until(webDriver ->
        webDriver.findElement(By.id("nameInput")));
    WebElement authorField = driver.findElement(By.id("authorInput"));
    WebElement priceField = driver.findElement(By.id("priceInput"));

    WebElement submitButton = driver.findElement(By.id("submitButton"));

    nameField.sendKeys("Clean Code");
    authorField.sendKeys("Robert Martin");
    priceField.sendKeys("600");

    submitButton.click();

    WebElement firstTd = wait.until(webDriver ->
        webDriver.findElement(By.tagName("td")));
    assertEquals("Clean Code", firstTd.getText());
}
```

- d. ลองรันเทส จะเห็นการกรอกข้อมูลหนังสือ จากนั้นมีการกดปุ่มและไปที่หน้ารายการหนังสือ

7. หากต้องการตรวจสอบข้อมูลในตารางทั้งหมด สามารถใช้ xpath ช่วยดังนี้

```
@Test
void testAddBook() {
    WebElement nameField = wait.until(webDriver ->
        webDriver.findElement(By.id("nameInput")));
}
```

```

WebElement authorField = driver.findElement(By.id("authorInput"));
WebElement priceField = driver.findElement(By.id("priceInput"));
WebElement submitButton = driver.findElement(By.id("submitButton"));

nameField.sendKeys("Clean Code");
authorField.sendKeys("Robert Martin");
priceField.sendKeys("600");

submitButton.click();

WebElement name = wait.until(webDriver -> webDriver
    .findElement(By.xpath("//table/tbody/tr[1]/td[1]")));
WebElement author = driver
    .findElement(By.xpath("//table/tbody/tr[1]/td[2]"));
WebElement price = driver
    .findElement(By.xpath("//table/tbody/tr[1]/td[3]"));

assertEquals("Clean Code", name.getText());
assertEquals("Robert Martin", author.getText());
assertEquals("600.00", price.getText());
}

```

8. ลองรันเทส ว่ายังใช้ได้หรือไม่

9. Refactor เพิ่มเติม โดยใช้ @FindBy แทนการเรียกเมทอด webDriver.findElement() เพื่อให้โค้ดสั้นลง จากนั้น ลองรันเทส ว่ายังใช้ได้หรือไม่

```

@SpringBootTest(webEnvironment = SpringBootTest.WebEnvironment.RANDOM_PORT)
public class AddBookTests {

    @LocalServerPort
    private Integer port;

    private static WebDriver driver;
    private static WebDriverWait wait;

    @FindBy(id = "nameInput")
    private WebElement nameField;

    @FindBy(id = "authorInput")
    private WebElement authorField;

    @FindBy(id = "priceInput")
    private WebElement priceField;

    @FindBy(id = "submitButton")
    private WebElement submitButton;

    // ... setup and teardown ...

    @BeforeEach

```

```

public void beforeEach() {
    driver.get("http://localhost:" + port + "/book/add");
    PageFactory.initElements(driver, this);
}

// ... setup and teardown ...

@Test
void testAddBook() {
    WebElement nameField = wait.until(webDriver ->
        webDriver.findElement(By.id("nameInput")));
    WebElement authorField = driver.findElement(By.id("authorInput"));
    WebElement priceField = driver.findElement(By.id("priceInput"));
    WebElement submitButton = driver.findElement(By.id("submitButton"));

    nameField.sendKeys("Clean Code");
    authorField.sendKeys("Robert Martin");
    priceField.sendKeys("600");

    submitButton.click();

    WebElement name = wait.until(webDriver -> webDriver
        .findElement(By.xpath("//table/tbody/tr[1]/td[1]")));
    WebElement author = driver
        .findElement(By.xpath("//table/tbody/tr[1]/td[2]"));
    WebElement price = driver
        .findElement(By.xpath("//table/tbody/tr[1]/td[3]"));

    assertEquals("Clean Code", name.getText());
    assertEquals("Robert Martin", author.getText());
    assertEquals("600.00", price.getText());
}
}

```