

COMP8547 Advanced Computing Concepts – Fall 2023

Assignment 3

Variant 19 - Scrape data from a recipe site, such as recipe names, ingredients, and instructions

Source Code with Explanation

```
// Bhuvan's Package Declarationssss
package recipe_scrapper_Bhuvan;
// Bhuvan's import statemenstts
import org.openqa.selenium.By; // imidbfi
import org.openqa.selenium.JavascriptExecutor; //swlsnunu
import org.openqa.selenium.WebDriver; //selsniunm
import org.openqa.selenium.WebElement; //webelemtn
import org.openqa.selenium.safari.SafariDriver; //safari browserr
import org.openqa.selenium.support.ui.ExpectedConditions; // expect conditionos
import org.openqa.selenium.support.ui.WebDriverWait; // ui

import java.io.FileWriter; //nesesary java
import java.io.IOException; //ioexe
import java.time.Duration; // durations
import java.util.List; //Isitss

public class RecipeScraper_Bhuvan { //declarations of main classs forr the Bhuvan's program
    private static final String CSV_FILE_PATH = "recipes.csv"; // Declaring the file name in which I'm
    gonna save the scraped data

    public static void main(String[] args) { // maaaain funct
        WebDriver driver = new SafariDriver(); // I'm declaring the safari mac os web drive
        driver.manage().window().maximize(); // just for tghe visibility

        try {
            driver.get("https://www.simplyrecipes.com"); // This willl helpp in accessing the current
browserr
            Thread.sleep(2000); //justt declarring some wait timee of 2 secondss

            // Scrollll downn
            JavascriptExecutor js = (JavascriptExecutor) driver;
            js.executeScript("window.scrollTo(0, document.body.scrollHeight)");

            Thread.sleep(2000); // again the samee 2 secondss wait
```

```
WebElement element_Bhuvan_toClick = waitForElement(driver,
By.xpath("//*[@id=\"icon-menu_1-0\"]/li[1]/a")); // This is to activate a click using Selenium
Automation.
```

```
// Clicking on the element
if (element_Bhuvan_toClick != null) { // This is to verify if the Xpath is valid and the click
is valid
    element_Bhuvan_toClick.click(); // Upon successful verification, this command will
make the button click
} else {
    logError("The current Element you tried to click was not found! Try again later!"); // if
the conditions fail, it comes to the else block
    return;
}
```

```
Thread.sleep(2000); // again the same 2 seconds wait
```

```
WebElement element_Bhuvan_toClick2 = waitForElement(driver,
By.xpath("//*[@id=\"mntl-taxonomysc-child-node__more-button_1-0\"]")); // This is to activate
a click using Selenium Automation.
```

```
// Clicking on the element
if (element_Bhuvan_toClick2 != null) { // This is to verify if the Xpath is valid and the click
is valid
    element_Bhuvan_toClick2.click(); // Upon successful verification, this command will
make the button click
} else {
    logError("The current Element you tried to click was not found! Try again later!"); // if
the conditions fail, it comes to the else block
    return;
}
```

```
Thread.sleep(2000); // again the same 2 seconds wait
```

```
WebElement element_Bhuvan_toClick3 = waitForElement(driver,
By.xpath("//*[@id=\"mntl-taxonomysc-child-links__list_3-0-7\"]")); // This is to activate a click
using Selenium Automation.
```

```
// Click on the element
if (element_Bhuvan_toClick3 != null) { // This is to verify if the Xpath is valid and the click
is valid
    element_Bhuvan_toClick3.click(); // Upon successful verification, this command will
make the button click
} else {
```

```
        logError("The current Element you tired to click was not found! Try again later!"); // iff
the conditions fails, it comes to the else block
        return;
    }
}
```

```
Thread.sleep(2000); // again the same 2 secondss wait
```

```
WebElement element_Bhuvan_toClick4 = waitForElement(driver,
By.xpath("//button[text(),'Dismiss']")); // This is to activate a click using Selenium Automation.
```

```
    // Click on the element
    if (element_Bhuvan_toClick4 != null) { // This is too verify if the Xpath is valid and the click
is valid
        element_Bhuvan_toClick4.click(); // Uponn successfull verification, this command will
make the button click
    } else {
        logError("The current Element you tired to click was not found! Try again later!"); // iff
the conditions fails, it comes to the else block
        return;
    }
}
```

```
Thread.sleep(2000); // again the same 2 secondss wait
```

```
// ...
```

```
WebElement element_Bhuvan_toClick5 = waitForElement(driver,
By.xpath("//*[@id=\"mntl-card-list-items_3-2\"]/div[1]/div/img")); // This is to activate a click
using Selenium Automation.
```

```
    // Click on the element using JavascriptExecutor
    if (element_Bhuvan_toClick5 != null) { // This is too verify if the Xpath is valid and the click
is valid
        clickElementWithJs(driver, element_Bhuvan_toClick5); // Uponn successfull verification,
this command will make the button click
    } else {
        logError("The current Element you tired to click was not found! Try again later!"); // iff
the conditions fails, it comes to the else block
        return;
    }
}
```

```
    // Wait for the pop-up button to be clickable (adjust timeout as needed)
    WebElement popupButton_Bhuvan = waitForElement(driver,
By.xpath("//*[@id=\"taxonomyScTemplate_1-0\"]/body/div[2]/div/div[2]/div[2]/button[1]")); //
This is to activate a click using Selenium Automation.
```

```

        // Click on the pop-up button
        if (popupButton_Bhuvan != null) { // This is too verify if the Xpath is valid and the click is
valid
            popupButton_Bhuvan.click(); // Uponn successfull verification, this command will make
the button click
        } else {
            logError("The Popup button is not found! Try again later!"); // iff the conditions fails, it
comes to the else block
            return;
        }

        // Wait for some time after clicking the pop-up button (adjust sleep duration as needed)
        Thread.sleep(5000); // Here I'm making the prograam wait for 5 secondss

        List<WebElement> recipeElements = driver.findElements(By.xpath("//div[@class,'comp
mntl-taxonomysc-article-list mntl-document-card-list mntl-card-list mntl-block']/*"));

        for (int i = 0; i < recipeElements.size(); i++) { // Just passing a loop to get the recipe namess
            WebElement recipeElement = recipeElements.get(i);
            String recipeName_bhuvan = recipeElement.getAttribute("outerHTML");

            // Extract recipe details
            extract_Bhuvan_AndWriteRecipeData_Bhuvan(driver, recipeName_bhuvan, i + 1); //
increamentingg by one
        }

    } catch (InterruptedException e) {
        logError("InterruptedException: " + e.getMessage());
    } finally {
        // Quit the driver
        if (driver != null) {
            try {

                driver.wait(); // if all the work is done, it will quit the
browserr

            } catch (InterruptedException e) {

                e.printStackTrace();
            }
        }
    }
}

```

```

// Helperrr methodd tto exxxtract annd wriite reciipe detailss
private static void extract_Bhuvan_AndWriteRecipeData_Bhuvan(WebDriver driver, String
recipeName_bhuvan, int counter) {
    try {
        driver.get(recipeName_bhuvan); // Getting the name of the recipess

        // Check if the elements are present before proceeding
        WebDriverWait wait = new WebDriverWait(driver, Duration.ofSeconds(60)); // Passingg
the advancedd Selinimum commands for the browser to waitt.
        wait.until(ExpectedConditions.presenceOfElementLocated(By.xpath("//*[@id='sr-recipe-
callout']/div[4]"))); // thiss command will make the browser waitt till the item is located using
Xpath
        wait.until(ExpectedConditions.presenceOfElementLocated(By.xpath("//*[@id='sr-recipe-
method']/div"))); // thiss command will make the browser waitt till the item is located using Xpath

        WebElement recipeIngredients_Bhuvan = driver.findElement(By.xpath("//*[@id='sr-
recipe-callout']/div[4]")); // Thiss is to find the ingredientss based on id
        WebElement recipeInstructions_Bhuvan = driver.findElement(By.xpath("//*[@id='sr-
recipe-method']/div")); // Thiss is to find the insturctionss based on the id

        String ingredientsText_Bhuvan = recipeIngredients_Bhuvan.getText(); // Thiss will help to
get the contents as texxt format into that string
        String instructionsText_Bhuvan = recipeInstructions_Bhuvan.getText(); // Thiss will help to
get the contents as texxt format into that string

        // Writing to the CSV
        writeToCsv_Bhuvan(recipeName_bhuvan, ingredientsText_Bhuvan,
instructionsText_Bhuvan);

    } catch (Exception e) {
        logError("Exception while processing recipe: " + e.getMessage()); // for error handling
purpose
    }
}

// Helper method to write to CSV
private static void writeToCsv_Bhuvan(String recipeName_bhuvan, String ingredients_bhuvan,
String instructions_bhuvan) {
    try (FileWriter writer = new FileWriter(CSV_FILE_PATH, true)) {
        writer.append(recipeName_bhuvan).append(",").append(ingredients_bhuvan).append(",").app
end(instructions_bhuvan).append("\n"); // Thiss line will append all the texts to the CSV filee
    } catch (IOException e) {

```

```

        logError("IOException while writing to CSV: " + e.getMessage()); // for error handling
    }
}

// Helper method to wait for an element to be present and clickable
private static WebElement waitForElement(WebDriver driver, By locator) {
    try {
        WebDriverWait wait = new WebDriverWait(driver, Duration.ofSeconds(30)); //this is to
        make the driver wait for the particular duration.
        return wait.until(ExpectedConditions.elementToBeClickable(locator)); // making it
        clickable based on the locators
    } catch (Exception e) {
        logError("Exception while waiting for element: " + e.getMessage()); // errorr handlingss
        return null;
    }
}

// Helper method for logging errors
private static void logError(String message) {
    System.err.println("Error: " + message); // For logging errorss
}

// Helper method to click an element using JavascriptExecutor
private static void clickElementWithJs(WebDriver driver, WebElement element) {
    try {
        JavascriptExecutor executor = (JavascriptExecutor) driver; // helps in executing a javascript
        click on the elementss specifiedd
        executor.executeScript("arguments[0].click();", element);
    } catch (Exception e) {
        logError("Exception while clicking element with JavascriptExecutor: " + e.getMessage()); //
        handling errors with care
    }
}
}

```

Explanation:

This code is a Selenium-based web scraper for extracting recipe data from <https://www.simplyrecipes.com> using Java.

1. Setup:

- Uses Selenium WebDriver with Safari to automate browsing.
- Targets the recipe website.

2. Navigation:

- Scrolls down for page coverage.
- Navigates through menu items.

3. Interaction:

- Handles pop-ups, dismissing them.

4. Extraction:

- Identifies recipe elements and captures their outer HTML.

5. Details Extraction:

- Visits each recipe's page.
- Extracts ingredients and instructions.

6. Data Storage:

- Writes data (recipe name, ingredients, instructions) to "recipes.csv."

7. Error Handling:

- Logs errors, covering exceptions like InterruptedException and IOException.

8. Cleanup:

- Maximizes Safari window.

9. Structure:

- Organized into methods for clarity.
- Includes comments for better understanding.

Overall, it automates the process of navigating, extracting, and storing recipe details from the website. CSV file is attached.