117. Perform Web Table Sorting using Selenium Java Streams

stream method will take an input of your collection and the stream support multiple methods to manipulate and produce the desired output.

Now, when you apply a map, there is a method called map on the stream.

118. Build Custom Selenium methods using Streams Mapper

How do you get the price of the rice?

 I don't want all the web elements.I need web element which have a text of beans only.

So if you want to apply some condition,then you have to use filter method.

This filter method will help you to filter out and give what you need.

So basically you have to to scan again this first column.

And whenever you encounter Rice, then you have to move one step ahead.

You have to go to its ancestor and grab the price and print it in the output.

in lambda expressions left side argument is like the items present in your stream

or list. And rightside argument after the lambda expression is what action you have to perform on the element that you retrieved from the left side.

So right now we have implemented a code that it says only on the first page.

So let's see how to handle pagination In webtables and scan each and every page using selenium streams

119. Automating Pagination Scenarios to search the data using do while loop

120. Code Download

package tests;

import java.util.List;

import java.util.stream.Collectors;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.chrome.ChromeDriver;

import org.testng.Assert;

public class LiveDemo {

public static void main(String[] args) throws InterruptedException {

// TODO Auto-generated method stub

System.setProperty("webdriver.chrome.driver", "C://chromedriver.exe");

WebDriver driver = new ChromeDriver();

driver.get("https://rahulshettyacademy.com/greenkart/#/offers");

// click on column

driver.findElement(By.xpath("//tr/th[1]")).click();

// capture all webelements into list

List<WebElement> elementsList = driver.findElements(By.xpath("//tr/td[1]"));

// capture text of all webelements into new(original) list

List<String> originalList = elementsList.stream().map(s -> s.getText()).collect(Collectors.toList());

// sort on the original list of step 3 -> sorted list

List<String> sortedList = originalList.stream().sorted().collect(Collectors.toList());

// compare original list vs sorted list

Assert.assertTrue(originalList.equals(sortedList));

List<String> price;

// scan the name column with getText ->Beans->print the price of the Rice

do

{

List<WebElement> rows = driver.findElements(By.xpath("//tr/td[1]"));

price = rows.stream().filter(s -> s.getText().contains("Rice"))

.map(s -> getPriceVeggie(s)).collect(Collectors.toList());

price.forEach(a -> System.out.println(a));

if(price.size()<1)

{

driver.findElement(By.cssSelector("[aria-label='Next']")).click();

}

}while(price.size()<1);

}

private static String getPriceVeggie(WebElement s) {

// TODO Auto-generated method stub

String pricevalue = s.findElement(By.xpath("following-sibling::td[1]")).getText();

return pricevalue;

}

}

121. Filter the web table using Selenium Java streams

New req : if I want to filter it with item called rice, so when you start typing, then you see that there is only one item displaying here. So you should write an automation script to validate that whether whatever you provide in the search is the same vegetable related stuff is coming in the table or not.

122. Code download

package tests;

import java.util.List;

import java.util.stream.Collectors;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.chrome.ChromeDriver;

import org.testng.Assert;

public class filter {

public static void main(String[] args) {

// TODO Auto-generated method stub

System.setProperty("webdriver.chrome.driver", "C://chromedriver.exe");

        WebDriver driver=new ChromeDriver();

        driver.get("https://rahulshettyacademy.com/greenkart/#/offers");

        driver.findElement(By.id("search-field")).sendKeys("Rice");

// get all the first column elements and put it in one webelement list

       List<WebElement> veggies=driver.findElements(By.xpath("//tr/td[1]"));

        //1 results

       List<WebElement> filteredList= veggies.stream().filter(veggie->veggie.getText().contains("Rice")).

        collect(Collectors.toList());

       //1 result

       Assert.assertEquals(veggies.size(), filteredList.size());

}

}