Dropdown values are sorted or not in selenium ?

Table of content

Dropdown values are sorted or not in selenium

Frankly, I do not want to write this kind of article, but I have read it in a blog about the verify whether the option is sorted in the dropdown or not ? And it was utterly wrong. So I wanted to provide the right way(s) to do it.

Let's write a simple way (as mentioned in another blog):

Collections.sort() || Read completely

Open the browser and navigate to the webpage

Find the dropdown using the findElement method in selenium

Create an object to Select class and pass the dropdown element as the parameter to the constructor

```
WebElement element = driver.findElement(By.xpath("//select[@id='animals']"));
Select se = new Select(element);
```

Using the getOptions() method from Select class, you can get all the options from the dropdown in the form of WebElement.

Using the loop, we can retrieve the values from the List of WebElement

Add all the values into a list called `originalList` *that we have already created*

```
List<String> originalList = new ArrayList();
for (WebElement e : se.getOptions()) {
   originalList.add(e.getText());
}
```

The values we retrieved could be sorted or not sorted values [we are not sure, we have to verify this]

Now let's create a temporary list called `tempList` *and get the values from* `originalList`

Now sort the Either `tempList` *or* `originalList` *and compare them, We can sort the list using the Collections.sort(list) method*

```
List<String> tempList= originalList;
Collections.sort(tempList);
```

We can compare the list using the Assertion class in TestNG

Complete program for not working sorting

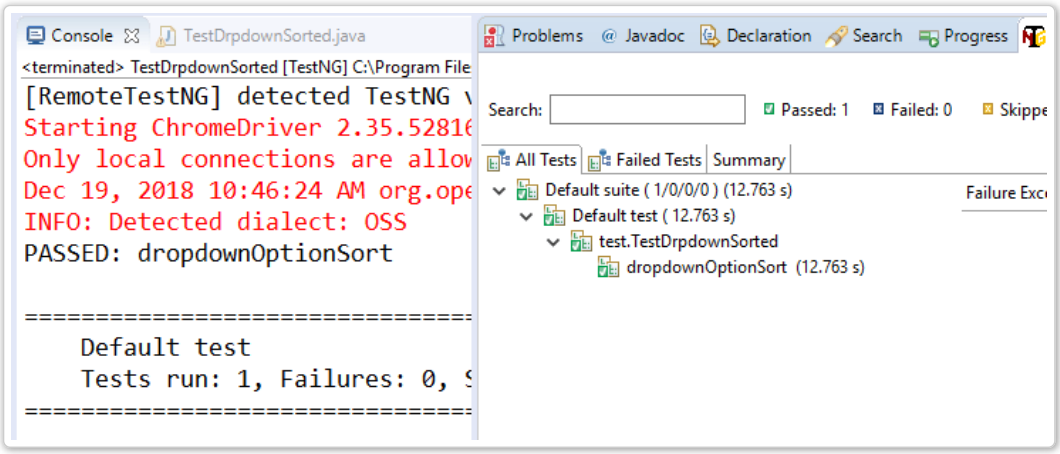
```
public class TestDrpdownSorted {
    @Test
    public void runTestOnDocker() throws Exception {
        String driverPath = "D:\PATH\chromedriver.exe";
        System.setProperty("webdriver.chrome.driver", driverPath);

        WebDriver driver = new ChromeDriver();
        driver.get("https://chercher.tech/practice/practice-dropdowns-selenium-webdriver");

        WebElement element = driver.findElement(By.xpath("//select[@id='animals']"));
        Select se = new Select(element);
```

```
        List<String> originalList = new ArrayList();
        for (WebElement e : se.getOptions()) {
           originalList.add(e.getText());
        }
        //----logic block starts
        List<String> tempList= originalList;
        Collections.sort(tempList);
        Assert.assertEquals(tempList, originalList);
        //----logic ends starts
    }
}
```

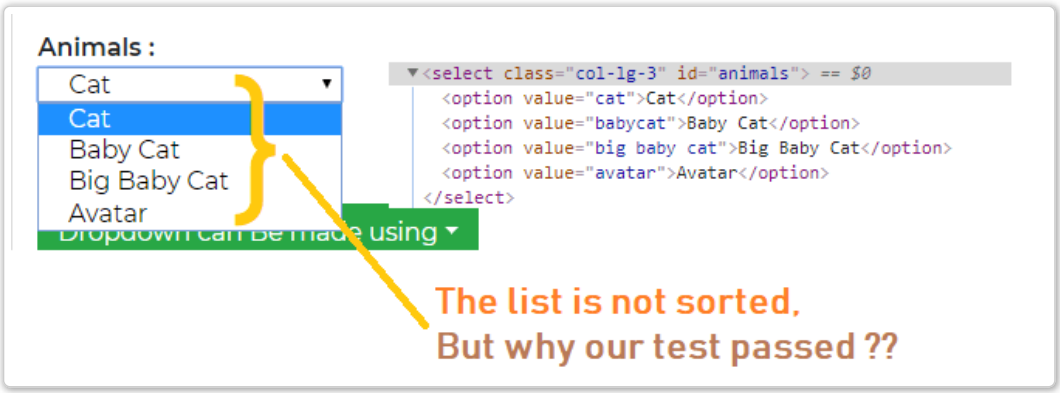
The output of the program, Everything works fine



This was the explanation/solution given in the other blog

Lets bring the twist :

Below is the **screenshot** of the dropdown



Now, you might think, where did the mistake happen, Let's bring some print statements in logic block

```
System.out.println("
 this is originalList before Sorting tempList"+ originalList);
Collections.sort(tempList);
System.out.println("
 this is originalList after sorting tempList"+ originalList);
System.out.println("
 this is tempList"+ tempList);
Assert.assertEquals(tempList, originalList);
```

```
INFO: Detected dialect: OSS
```

The output of the program:

```
 this is originalList before Sorting tempList[Cat, Baby Cat, Big Baby Cat, Avatar]

 this is originalList after sorting tempList[Avatar, Baby Cat, Big Baby Cat, Cat]

 this is tempList[Avatar, Baby Cat, Big Baby Cat, Cat]
PASSED: dropdownOptionSort

===========+====================================
    Default test
    Tests run: 1, Failures: 0, Skips: 0
================================================
```

If you notice above underlines in the screenshot, we see the change in the Original list i.e original list got sorted, but at the same time we have not sorted the original list

So the test gets pass all the time because the sequence in the originalList and tempList is going to be the same.

If you are following the above process, then your test never fails, because when you change one list, it changes the other list as well.

Dropdown Options are sorted or Not ?

Subscribe to my youtube channel :

how do you check the elements in the dropdown are sorted in ascending order or not

We have a few ways to verify whether options are sorted or not in dropdowns with webdriver.

For loop [basic]

addAll()

TreeSet

stream().collect()

1. For loop

Create a List tempList variable

While iterating the option in the dropdown, add values to tempList (along with originalList)

Now sort the tempList, sorting of **`tempList`** *will not affect the* **`originalList`** *because we have created two different objects*

Compare the two Lists

```
WebElement element = driver.findElement(By.xpath("//select[@id='animals']"));
Select se = new Select(element);
List<String> originalList = new ArrayList();
List<String> tempList = new ArrayList();
for (WebElement e : se.getOptions()) {
   originalList.add(e.getText());
   tempList.add(e.getText());
}

System.out.println("
 this is originalList before Sorting tempList"+ originalList);
Collections.sort(tempList);
System.out.println("
 this is originalList after sorting tempList"+ originalList);
System.out.println("
 this is tempList"+ tempList);
Assert.assertEquals(tempList, originalList);
```

Now the test result is a failure because dropdown options are not sorted

```
 this is originalList before Sorting tempList[Cat, Baby Cat, Big Baby Cat, Avatar]

 this is tempList before Sorting[Cat, Baby Cat, Big Baby Cat, Avatar]

 this is originalList after sorting tempList[Cat, Baby Cat, Big Baby Cat, Avatar]

 this is tempList after Sorting[Avatar, Baby Cat, Big Baby Cat, Cat]
FAILED: dropdownOptionSort
java.lang.AssertionError: Lists differ at element [0]: Cat != Avatar expected [Cat] but found
        at org.testng.Assert.fail(Assert.java:96)
        at org.testng.Assert.failNotEquals(Assert.java:776)
        at org.testng.Assert.assertEqualsImpl(Assert.java:137)
        at org.testng.Assert.assertEquals(Assert.java:831)
        at org.testng.Assert.assertEquals(Assert.java:796)
        at test.TestDrpdownSorted.dropdownOptionSort(TestDrpdownSorted.java:45)
```

if you compare originalList and tempList, test fails as expected because dropdown is not sorted

> If you are thinking like, why cannot I copy the originalList elements using another loop, then surely you can do, but you will be using two loops in the place of one loop. This method is not recommended

2. addAll()

I want you guys, to give a try to use addAll() on a newly created list and achieve the same above results [leave a comment if you know/do not know the answer]

3. TreeSet

Before we proceed with the process, lets see the trait of Treelist:

The objects of the TreeSet class are stored in ascending order

The TreeSet stores the objects based on the comparator provided; if there is no comparator, then it is stored in ascending order.

We can use the TreeSet for verification dropdown option order: we have to create a TreeSet object using the list as a parameter for its constructor.

Compare the values of the treeSet and the List using **Assert methods**.

> Assert methods accepts Iteratable objects, so now you should stop worrying about how you can compare Set and List

```
WebElement element = driver.findElement(By.xpath("//select[@id='animals']"));
Select se = new Select(element);
List<String> originalList = new ArrayList();
for (WebElement e : se.getOptions()) {
   originalList.add(e.getText());
}

System.out.println("
 this is originalList before Sorting tempList"+ originalList);

Set<String> treeset = new TreeSet(originalList);
System.out.println("
 this is TreeSet "+ treeset);
System.out.println("
 this is originalList after sorting tempList"+ originalList);

Assert.assertEquals(treeset, originalList);
```

The output of the execution

```
 this is originalList before Sorting tempList[Cat, Baby Cat, Big Baby Cat, Avatar]

 this is TreeSet [Avatar, Baby Cat, Big Baby Cat, Cat]
 this is originalList after sorting tempList[Cat, Baby Cat, Big Baby Cat, Avatar]

FAILED: dropdownOptionSort
java.lang.AssertionError: Lists differ at element [0]: Cat != Avatar expected [Cat] but found [Avat
        at org.testng.Assert.fail(Assert.java:96)
        at org.testng.Assert.failNotEquals(Assert.java:776)
        at org.testng.Assert.assertEqualsImpl(Assert.java:137)
        at org.testng.Assert.assertEquals(Assert.java:831)
        at org.testng.Assert.assertEquals(Assert.java:796)
        at test.TestDrpdownSorted.dropdownOptionSort(TestDrpdownSorted.java:45)
```

test fails as options in dropdown are not sorted

4. stream().collect()

We can use the **`stream()`** method along with the **`collect()`** method to create new List object with elements

```
WebElement element = driver.findElement(By.xpath("//select[@id='animals']"));
Select se = new Select(element);
List<String> originalList = new ArrayList();
for (WebElement e : se.getOptions()) {
   originalList.add(e.getText());
}

List<String> tempList = originalList.stream().collect(Collectors.toList());
System.out.println("
 this is originalList before Sorting tempList"+ originalList);
System.out.println("

 this is tempList"+ tempList);
Collections.sort(tempList);
System.out.println("
 this is originalList after Sorting tempList"+ originalList);
System.out.println("
 this is tempList"+ tempList);
Assert.assertEquals(tempList, originalList);
```

The output of the program to verify dropdown option order in **selenium**

```
 this is originalList before Sorting tempList[Cat, Baby Cat, Big Baby Cat, Avatar]

 this is tempList[Cat, Baby Cat, Big Baby Cat, Avatar]

 this is originalList after Sorting tempList[Cat, Baby Cat, Big Baby Cat, Avatar]

 this is tempList[Avatar, Baby Cat, Big Baby Cat, Cat]

FAILED: dropdownOptionSort
java.lang.AssertionError: Lists differ at element [0]: Cat != Avatar expected [Cat] but found
        at org.testng.Assert.fail(Assert.java:96)
        at org.testng.Assert.failNotEquals(Assert.java:776)
        at org.testng.Assert.assertEqualsImpl(Assert.java:137)
        at org.testng.Assert.assertEquals(Assert.java:831)
        at org.testng.Assert.assertEquals(Assert.java:796)
```