Selenium Easy

For Complete Web Automation Testing Tutorials using Selenium Webdriver

Working with Checkbox using Resuable Methods

Working with checkbox is very simple using webdriver. It is same as click operation. But it is always recommended to check if that is already in selected mode or deselected. Because, if it is already selected and when you click on the same element it will get deselected. We will look into different ways to perform select and de-select operations on checkboxes. We will also use reusable methods to perform the operation in multiple tests.

The below is the simple command to do that:

```
WebElement checkBoxElement=driver.findElement(By.id("persist_box"));
checkBoxElement.click();
```

I will try to explain you both select and de-select with the sample code:

```java
package com.pack.methods;
import java.util.List;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.firefox.FirefoxDriver;
import org.openqa.selenium.support.ui.ExpectedConditions;
import org.openqa.selenium.support.ui.WebDriverWait;
import org.testng.annotations.Test;

public class CheckBoxExample {

        private WebDriver driver;
        private String basePageURL;

        @Test
        public void testCaseToCheck() {
                driver = new FirefoxDriver();
                driver.get(basePageURL);
                WebElement checkBoxElement=driver.findElement(By.id("persist_box"));
                //Wait for the checkbox element to be visible
                new WebDriverWait(driver, 10).until(ExpectedConditions.visibilityOf(checkBoxElement));
                Select_The_Checkbox(checkBoxElement);
        }

        @Test
        public void testCaseToUnCheck() {
                driver.navigate().to(basePageURL);
                WebElement checkBoxElement=driver.findElement(By.id("persist_box"));
                new WebDriverWait(driver, 10).until(ExpectedConditions.visibilityOf(checkBoxElement));
                DeSelect_The_Checkbox(checkBoxElement);
        }

        @Test
        public void testCaseToCheckDesired(){
                driver.navigate().to("someother page");
                WebElement element = driver.findElement(By.cssSelector(".display"));
                Select_The_CheckBox_from_List(element, "soccer");
        }
```

In the above program, we have used reusable methods to select, deselect and select a particular value from multiple checkboxes. It is always better to have resubale to methods, so that we can resuse the same methods in multiple tests instead of writing the same code in multiple tests to perform the same operation.

```java
Select_The_Checkbox(checkBoxElement);
DeSelect_The_Checkbox(checkBoxElement);
Select_The_CheckBox_from_List(element, "soccer");
```

Below are the resuable methods that are used in the above example code. You can write these methods in a separate class called generics or utils class where you can have all the resuable methods like below

Below method is used to Select a Checkbox, if it is not selected already

```java
        public void Select_The_Checkbox(WebElement element) {
                try {
            if (element.isSelected()) {
                System.out.println("Checkbox: " + element + "is already selected");
            } else {
                // Select the checkbox
                element.click();
            }
        } catch (Exception e) {
                System.out.println("Unable to select the checkbox: " + element);
        }


        }
```

Below method is used to De-select a Checkbox, if it is selected already

```java
        public void DeSelect_The_Checkbox(WebElement element) {
                try {
            if (element.isSelected()) {
                //De-select the checkbox
                element.click();
            } else {
                System.out.println("Checkbox: "+element+"is already deselected");
            }
        } catch (Exception e) {
                System.out.println("Unable to deselect checkbox: "+element);
        }
    }
```

Below method is used to select the checkbox with the specified value from multiple checkboxes.

```java
        public void Select_The_CheckBox_from_List(WebElement element, String valueToSelect) {
                List<WebElement> allOptions = element.findElements(By.tagName("input"));
                for (WebElement option : allOptions) {
                        System.out.println("Option value "+option.getText());
                                if (valueToSelect.equals(option.getText())) {
                                    option.click();
                                    break;
                                }
                        }
        }

    }
```

‹ Assertions in TestNG   Working with Select examples ›

Tags:

Checkboxes
Reusables

Selenium Tutorials:

Selenium Tutorials

Selenium Tutorials

- Introduction to selenium
- Basic HTML concepts required for Selenium
- Locators for Selenium
- How to Run Webdriver in chrome browser?
- How to run webdriver in IE browser?
- Selenium webdriver using Eclipse
- Synchronization in Selenium Webdriver
- XPath tutorial for Selenium
- CSS selectors for Selenium with example
- Firefox Not Connected Exception in Selenium
- Handle windows popups using Selenium Webdriver
- Working with Frames in Selenium Webdriver
- How to handle javascript alerts, confirmation and prompts?
- Defining Firefox Profile preferences using Selenium Webdriver
- Why Automation Testing is required?
- Importance of Automation Testing
- Mouse Hover Actions in Selenium Webdriver
- StaleElementReference Exceptions in Selenium Webdriver
- Difference between Webdriver get() and Webdriver navigate()
- Navigation Methods in Webdriver with Examples
- Handling Cookies in Webdriver
- How to addcookie with Selenium Webdriver
- How to delete Cookies in Selenium Webdriver
- WebDriver Methods
- Automation Test Frameworks
- Taking Screenshot using Webdriver
- Webdriver SELECT Methods to work with Dropdowns
- Webdriver Select with Multiple Attribute
- Handling keyboard events and mouse hover events using

Recent Post

Automation Framework

Upload file using AutoIT

Keyword Driven Framework Example

Read data from Properties file using Java Selenium

Tags Cloud

Asserts (1)   AutoComplete (1)   AutoIt (2)   Automation (3)
Build.xml (1)   Checkboxes (2)   Cookies (3)   Css (2)
Css Examples (2)   DataProvider (1)   Date Picker (1)
Drag and Drop (1)   Excel (4)   Exceptions (4)   File Download (3)
Firefox Profile (2)   Framework Frameworks (5)   iFrames (1)
Images (1)   Interview Questions (1)   JavascriptExecutor (3)   JUnit
(2)   Keyword Driven (1)   Listeners (3)   Log4j (4)   log4j.xml (1)
Manual Testing (1)   Maven (5)   Mouse Hover (2)
Page Object Model (2)   Page Scroll (1)   Parallel Test (3)
properties file (1)   Re-Run Failed Tests (2)   Reusables (2)
Screenshot (3)   Select (3)   Selenium (2)   Testing Types (1)
testNG (2)   ToolTip (1)   Upload (5)   Waits (3)   Webdriver (9)
Webdriver Examples (6)   Webdriver methods (2)   Windows (2)
Xpath (3)   Xpath Examples (2)
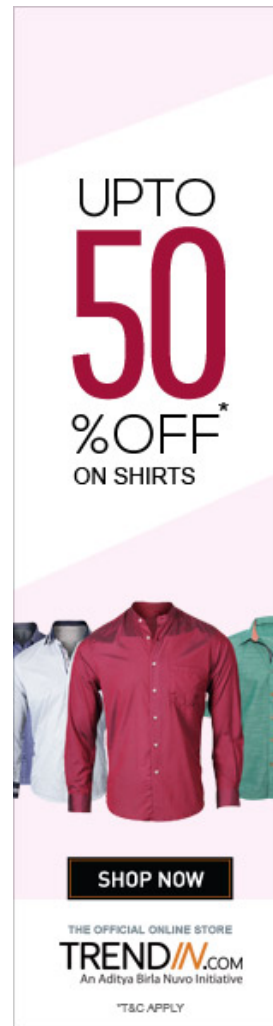
Home

- Home
- Tutorials
- About
- Contact
- Privacy Policy
- Site Map

Tutorial Menu

- Selenium Tutorials
- TestNG Tutorial
- JUnit Tutorial
- JXL Tutorial
- Apache POI
- ANT Tutorial
- Maven Tutorial

Selenium Tutorials

- Basic Webdriver Tutorials
- Advanced Webdriver Tutorials
- Selenium Framework Examples
- JavascriptExecutor Examples

Interview Questions

- Testing Concepts
- Quality Standards
- General Questions
- Roles & Responsibilities
- Technology & Environment

We welcome your questions and requests