# Selenium WebDriver: Access Form, CheckBox, RadioButton & TextBox

<https://www.guru99.com/accessing-forms-in-webdriver.html>

Forms are the fundamental web elements to receive information from the website visitors. Web forms have different GUI elements like Text boxes, Password fields, Checkboxes, Radio buttons, dropdowns, file inputs, etc.

We will see how to access these different form elements using Selenium Web Driver with Java. **Selenium encapsulates every form element as an object of WebElement.** It provides API to find the elements and take action on them like entering text into text boxes, clicking the buttons, etc. We will see the methods that are available to access each form element.

In this tutorial, we will see how to identify the following form elements

* [Introduction to WebElement, findElement(), findElements()](https://www.guru99.com/accessing-forms-in-webdriver.html#1)
* [Input Box](https://www.guru99.com/accessing-forms-in-webdriver.html#2)
* [Entering Values in Input Boxes](https://www.guru99.com/accessing-forms-in-webdriver.html#3)
* [Deleting Values in Input Boxes](https://www.guru99.com/accessing-forms-in-webdriver.html#4)
* [Buttons](https://www.guru99.com/accessing-forms-in-webdriver.html#5)
* [Submit Buttons](https://www.guru99.com/accessing-forms-in-webdriver.html#6)
* [Radio Button](https://www.guru99.com/accessing-forms-in-webdriver.html#7)
* [Check Box](https://www.guru99.com/accessing-forms-in-webdriver.html#8)
* [Complete Code](https://www.guru99.com/accessing-forms-in-webdriver.html#9)
* [Troubleshooting](https://www.guru99.com/accessing-forms-in-webdriver.html#10)

**Introduction to WebElement, findElement(), findElements()**

Selenium Web Driver encapsulates a simple form element as an object of **WebElement.**

There are various techniques by which the WebDriver identifies the form elements based on the different properties of the Web elements like ID, Name, Class, XPath, Tagname, CSS Selectors, link Text, etc.

Web Driver provides the following two methods to find the elements.

* **findElement()** – finds a single web element and returns as a WebElement object.
* **findElements()** – returns a list of WebElement objects matching the locator criteria.

Let's see the code snippets to get a single element – Text Field in a web page as an object of WebElement using findElement() method. We shall cover the findElements() method of finding multiple elements in subsequent tutorials.

**Step 1:**We need to import this package to create objects of Web Elements

[https://cdn.guru99.com/images/2-2017/072717_0632_SeleniumWeb1.png](https://cdn.guru99.com/images/2-2017/072717_0632_SeleniumWeb1.png)

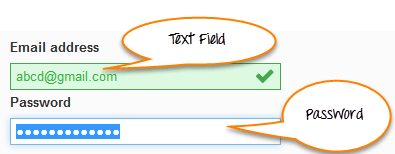
**Step 2:**We need to call the findElement() method available on the WebDriver class and get an object of WebElement.

Refer below to see how it is done.

**Input Box**

Input boxes refer to either of these two types:

1. **Text Fields**- text boxes that accept typed values and show them as they are.
2. **Password Fields**- text boxes that accept typed values but mask them as a series of special characters (commonly dots and asterisks) to avoid sensitive values to be displayed.

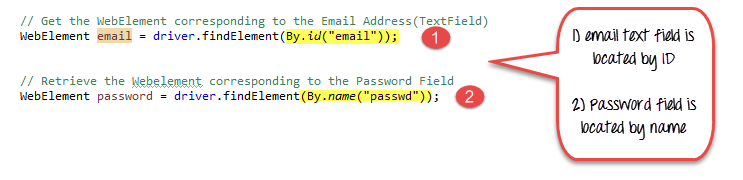
[](https://cdn.guru99.com/images/2-2017/072717_0632_SeleniumWeb2.png)

**Locators**

The method findElement() takes one parameter which is a locator to the element. Different locators like By.id(), By.name(), By.xpath(), By.CSSSelector() etc. locate the elements in the page using their properties like`````` id, name or path, etc.

You can use plugins like Fire path to get help with getting the id, xpath, etc. of the elements.

Using the example site <http://demo.guru99.com/test/login.html> given below is the code to locate the "Email address" text field using the id locator and the "Password "field using the name locator.

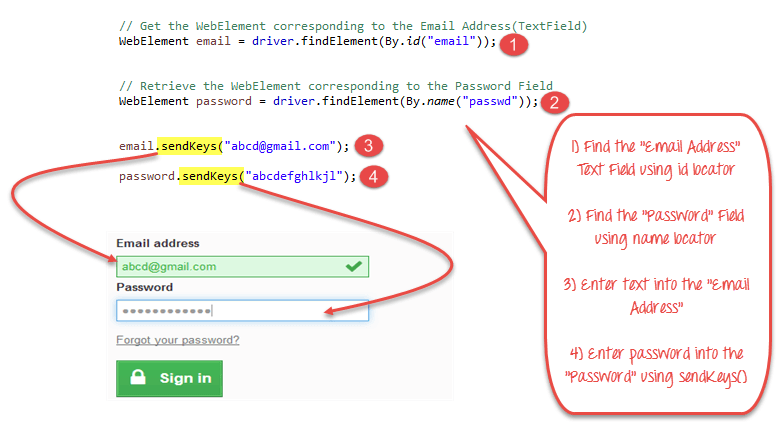
[](https://cdn.guru99.com/images/2-2017/072717_0632_SeleniumWeb3.png)

1. Email text field is located by Id
2. Password field is located by name

**Entering Values in Input Boxes**

To enter text into the Text Fields and Password Fields, sendKeys() is the method available on the WebElement.

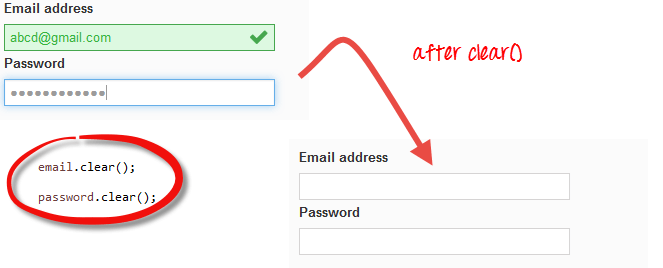
Using the same example of <http://demo.guru99.com/test/login.html> site, here is how we find the Text field and Password fields and enter values into them.

[](https://cdn.guru99.com/images/2-2017/072717_0632_SeleniumWeb4.png)

1. Find the "Email Address" Text field using the id locator.
2. Find the "Password" field using the name locator
3. Enter text into the "Email Address" using the sendKeys() method.
4. Enter a password into the "Password" field using the sendKeys() method.

**Deleting Values in Input Boxes**

The **clear()** method is used to delete the text in an input box. **This method does not need a parameter**. The code snippet below will clear out the text from the Email or Password fields

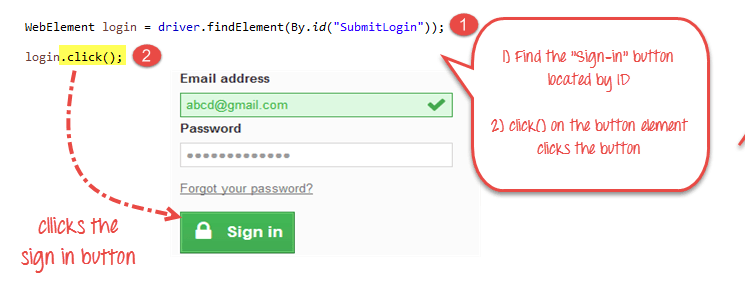
[](https://cdn.guru99.com/images/2-2017/072717_0632_SeleniumWeb5.png)

**Buttons**

The buttons can be accessed using the click() method.

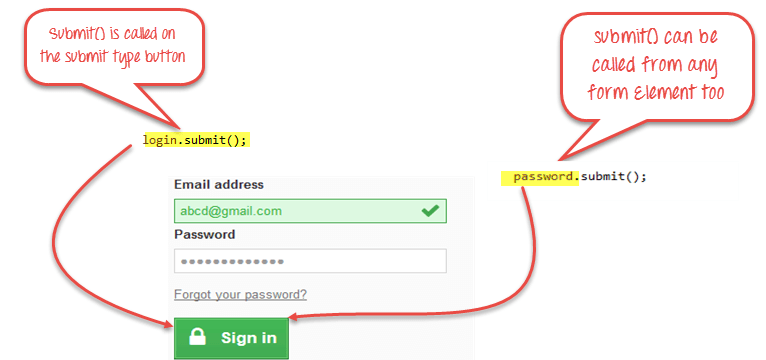
In the example above

1. Find the button to Sign in
2. Click on the "Sign-in" Button in the login page of the site to login to the site.

[](https://cdn.guru99.com/images/2-2017/072717_0632_SeleniumWeb6.png)

**Submit Buttons**

Submit buttons are used to submit the entire form to the server. We can either use the click () method on the web element like a normal button as we have done above or use the submit () method on any web element in the form or on the submit button itself.

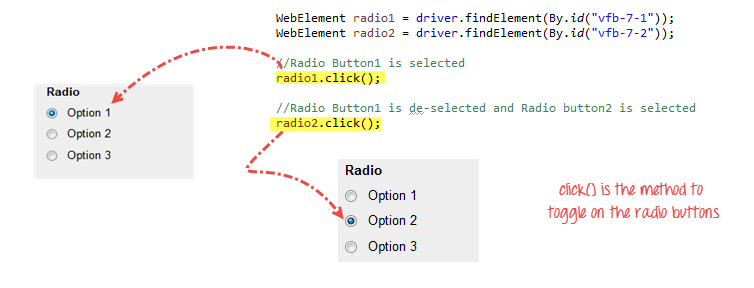
[](https://cdn.guru99.com/images/2-2017/072717_0632_SeleniumWeb7.png)

**When submit() is used, WebDriver will look up the DOM to know which form the element belongs to, and then trigger its submit function.**

**Radio Button**

Radio Buttons too can be toggled on by using the click() method.

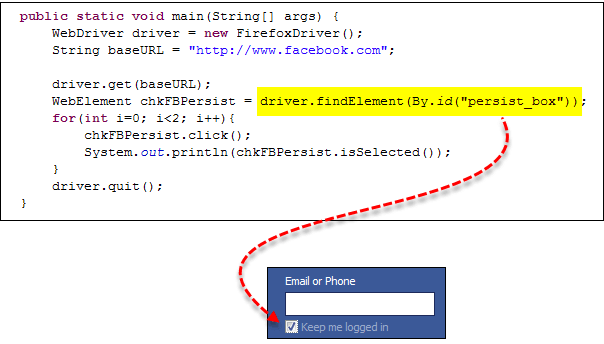
Using <http://demo.guru99.com/test/radio.html> for practise, see that radio1.click() toggles on the "Option1" radio button. radio2.click() toggles on the "Option2" radio button leaving the "Option1" unselected.

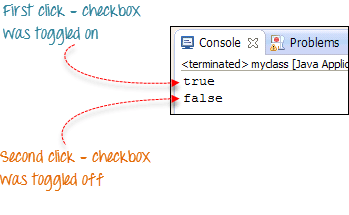
[](https://cdn.guru99.com/images/2-2017/072717_0632_SeleniumWeb8.png)

**Check Box**

Toggling a check box on/off is also done using the **click()** method.

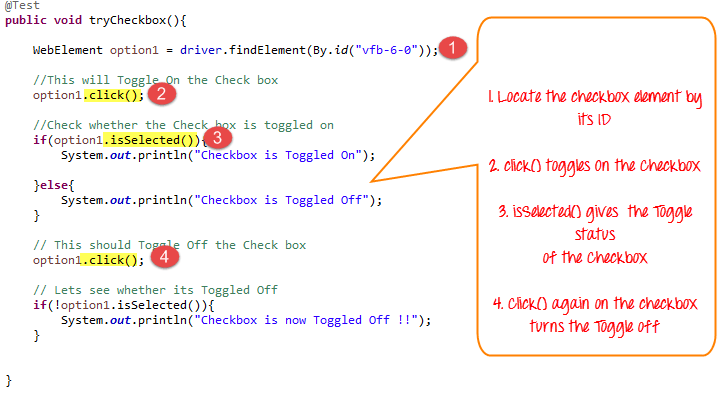
The code below will click on Facebook's "Keep me logged in" check box twice and then output the result as TRUE when it is toggled on, and FALSE if it is toggled off.

[](https://cdn.guru99.com/images/2-2017/072717_0632_SeleniumWeb9.png)

[](https://cdn.guru99.com/images/2-2017/072717_0632_SeleniumWeb10.png)

isSelected() method is used to know whether the Checkbox is toggled on or off.

Here is another example: <http://demo.guru99.com/test/radio.html>

[](https://cdn.guru99.com/images/2-2017/072717_0632_SeleniumWeb11.png)

**Complete Code**

Here is the complete working code

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.chrome.ChromeDriver;

import org.openqa.selenium.\*;

public class Form {

public static void main(String[] args) {

// declaration and instantiation of objects/variables

System.setProperty("webdriver.chrome.driver","G:\\chromedriver.exe");

WebDriver driver = new ChromeDriver();

String baseUrl = "http://demo.guru99.com/test/login.html";

driver.get(baseUrl);

// Get the WebElement corresponding to the Email Address(TextField)

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

// Get the WebElement corresponding to the Password Field

WebElement password = driver.findElement(By.name("passwd"));

email.sendKeys("[abcd@gmail.com](mailto:abcd@gmail.com)");

password.sendKeys("abcdefghlkjl");

System.out.println("Text Field Set");

// Deleting values in the text box

email.clear();

password.clear();

System.out.println("Text Field Cleared");

// Find the submit button

WebElement login = driver.findElement(By.id("SubmitLogin"));

// Using click method to submit form

email.sendKeys("[abcd@gmail.com](mailto:abcd@gmail.com)");

password.sendKeys("abcdefghlkjl");

login.click();

System.out.println("Login Done with Click");

//using submit method to submit the form. Submit used on password field

driver.get(baseUrl);

driver.findElement(By.id("email")).sendKeys("[abcd@gmail.com](mailto:abcd@gmail.com)");

driver.findElement(By.name("passwd")).sendKeys("abcdefghlkjl");

driver.findElement(By.id("SubmitLogin")).submit();

System.out.println("Login Done with Submit");

driver.get("http://demo.guru99.com/test/radio.html");

WebElement radio1 = driver.findElement(By.id("vfb-7-1"));

WebElement radio2 = driver.findElement(By.id("vfb-7-2"));

//Radio Button1 is selected

radio1.click();

System.out.println("Radio Button Option 1 Selected");

//Radio Button1 is de-selected and Radio Button2 is selected

radio2.click();

System.out.println("Radio Button Option 2 Selected");

// Selecting CheckBox

WebElement option1 = driver.findElement(By.id("vfb-6-0"));

// This will Toggle the Check box

option1.click();

// Check whether the Check box is toggled on

if (option1.isSelected()) {

System.out.println("Checkbox is Toggled On");

} else {

System.out.println("Checkbox is Toggled Off");

}

//Selecting Checkbox and using isSelected Method

driver.get("http://demo.guru99.com/test/facebook.html");

WebElement chkFBPersist = driver.findElement(By.id("persist\_box"));

for (int i=0; i<2; i++) {

chkFBPersist.click ();

System.out.println("Facebook Persists Checkbox Status is - "+chkFBPersist.isSelected());

}

//driver.close();

}

}

**Troubleshooting**

If you encounter NoSuchElementException() while finding elements, it means that the element is not found in the page at the point the Web driver accessed the page.

1. Check your locator again using Firepath or Inspect Element in Chrome.
2. Check whether the value you used in the code is different from the one for the element in Firepath now.
3. Some properties are dynamic for few elements. In case, you find that the value is different and is changing dynamically, consider using By.xpath() or By.cssSelector() which are more reliable but complex ways.
4. Sometimes, it could be a wait issue too i.e., the Web driver executed your code even before the page loaded completely, etc.
5. Add a wait before findElement() using implicit or explicit waits.

**Summary**

* The table below summarizes the commands to access each type of element discussed above

| **Element** | **Command** | **Description** |
| --- | --- | --- |
| **Input Box** | sendKeys() | used to enter values onto text boxes |
| clear() | used to clear text boxes of its current value |
| **Check Box,Radio Button,** | click() | used to toggle the element on/off |
| **Links** | click() | used to click on the link and wait for page load to complete before proceeding to the next command. |
| **Submit Button** | submit() |  |

* WebDriver allows selection of more than one option in a multiple SELECT element.
* You can use the submit() method on any element within the form. WebDriver will automatically trigger the submit function of the form where that element belongs to.