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.

**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 WebElement methods to find the elements.

* **findElement()** – finds a single web element and returns as a WebElement Selenium 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

WebElements in Selenium

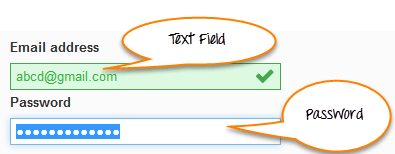
**Step 2:**We need to call the findElement() method available on the WebDriver class and get an object of WebElement.

Refer below to see how it is done.

**Input Box**

Input boxes refer to either of these two types:

1. **Text Fields**– Selenium input 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.

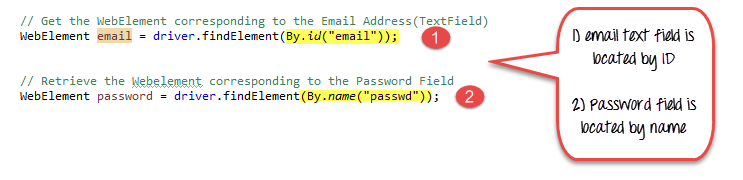


**Locators**

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.



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

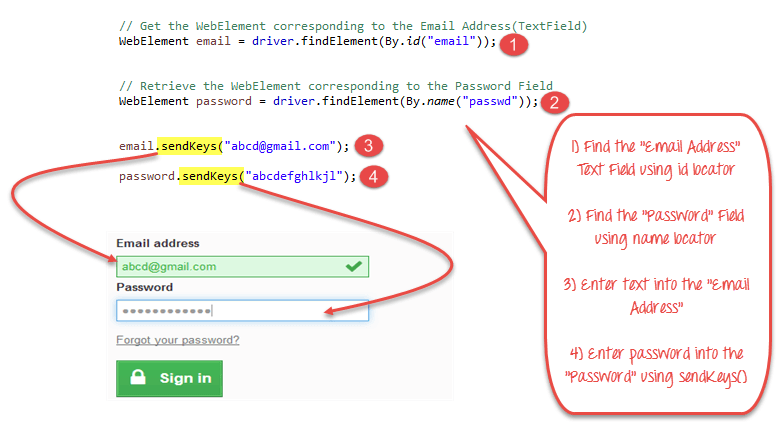
**sendkeys in Selenium**

**sendkeys() in Selenium** is a method used to enter editable content in the text and password fields during test execution. These fields are identified using locators like name, class, id, etc. It is a method available on the web element. Unlike the type method, sendkeys() method does not replace existing text in any text box.

**Entering Values in Input Boxes**

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

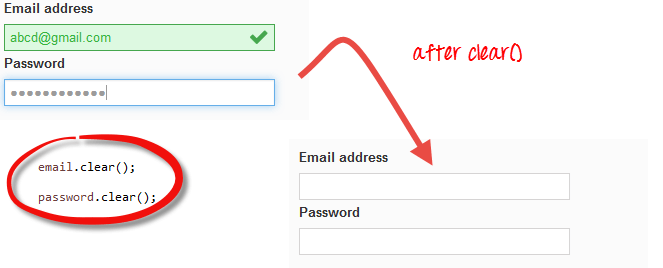
Using the same example of http://demo.guru99.com/test/login.html site, here is how we find the Text field and Password fields and enter text in Selenium.



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 Selenium 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

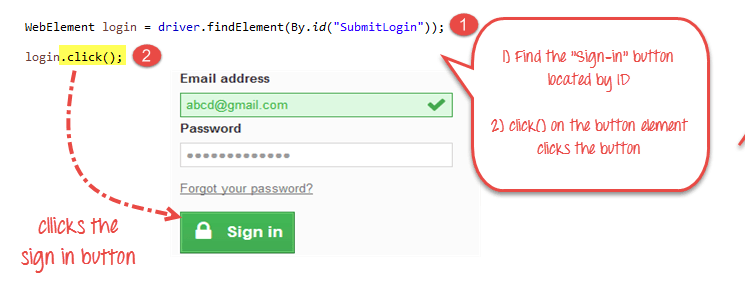


**Buttons**

The Selenium click button can be accessed using the click() method.

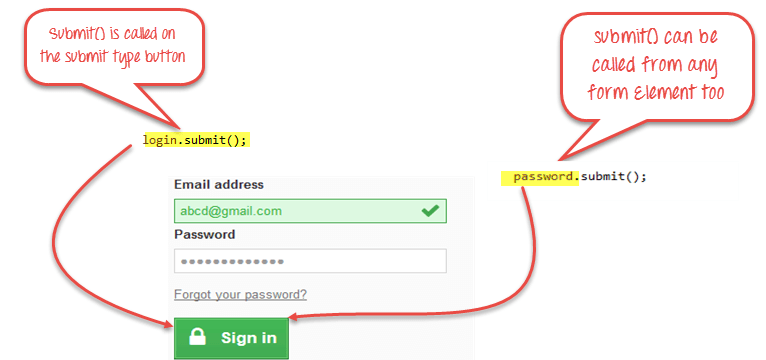
In the example above

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.



**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.



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

**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");

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");

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");

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

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

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

//driver.close();

}

}