

Golden rule -

For working with any automation. Line 13.

```
1 package seleniumsessions;
2
3 import org.openqa.selenium.WebDriver;
4 import org.openqa.selenium.chrome.ChromeDriver;
5
6 public class WebElementConcept {
7
8     public static void main(String[] args) {
9
10         WebDriver driver = new ChromeDriver();
11         driver.get("https://naveenautomationlabs.com/opencart/index.php?route=account/login");
12
13         //create a webelement(FE) + perform action(click, sendKeys, isDisplayed, getText..)
14     }
```

Enter username and password in login

```
15
16         driver.findElement(By.id("input-email")).sendKeys("tom@gmail.com");
17         driver.findElement(By.id("input-password")).sendKeys("tom@123");
18
```

Second approach to login

Better approach. Store and reuse when needed.

```
21         //2.
22         WebElement email = driver.findElement(By.id("input-email"));
23         WebElement password = driver.findElement(By.id("input-password"));
24
25         email.sendKeys("tom@gmail.com");
26         password.sendKeys("tom@123");
27
```

Third approach to login-

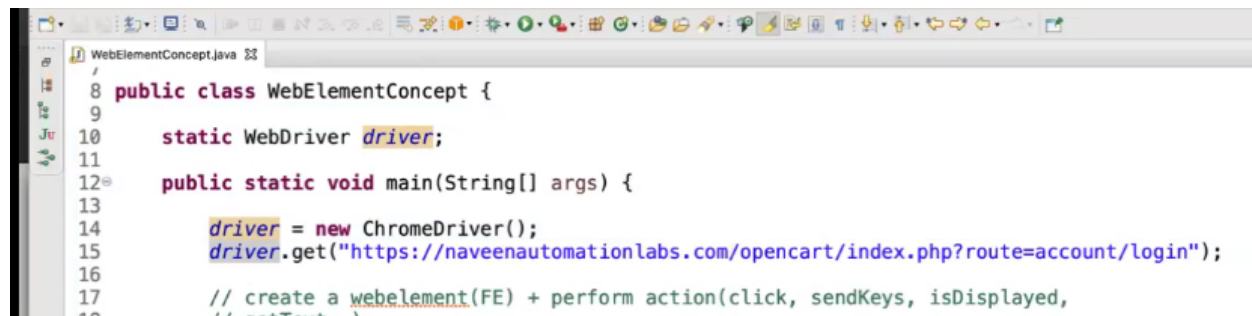
No unnecessary request sent to the server in finding the webelement.

```

28
29 //3. By locator:
30 By emailField = By.id("input-email");
31 By pwdField = By.id("input-password");
32
33 WebElement email = driver.findElement(emailField);
34 WebElement password = driver.findElement(pwdField);
35
36 email.sendKeys("tom@gmail.com");
37 password.sendKeys("tom@123");
38
39
40

```

Created static driver to access by class name  
Made driver global



```

WebElementConcept.java
8 public class WebElementConcept {
9
10     static WebDriver driver;
11
12     public static void main(String[] args) {
13
14         driver = new ChromeDriver();
15         driver.get("https://naveenautomationlabs.com/opencart/index.php?route=account/login");
16
17         // create a webelement(FE) + perform action(click, sendKeys, isDisplayed,
18         // getText

```

Create static method outside class level-  
Fourth method

```

46     }
47
48     public static WebElement getElement(By locator) {
49         return driver.findElement(locator);
50     }
51
52 }

```

Call static method

```

40
41 // 4. By locator + generic function to get the element.
42 By emailField = By.id("input-email");
43 By pwdField = By.id("input-password");
44
45 getElement(emailField).sendKeys("tom@gmail.com");
46 getElement(pwdField).sendKeys("tom@123");
47
48
49 }

```

## Fifth approach-

```

40
47 //5. By locator + generic function to get the element and perform action
48 By emailField = By.id("input-email");
49 By pwdField = By.id("input-password");
50
51 doSendKeys(emailField, "tom@gmail.com");
52 doSendKeys(pwdField, "tom@123");
53
54 }
55
56 public static void doSendKeys(By locator, String value) {
57     getElement(locator).sendKeys(value);
58 }
59
60 public static WebElement getElement(By locator) {
61     return driver.findElement(locator);
62 }
63
64 }
65

```

## ocd-

```

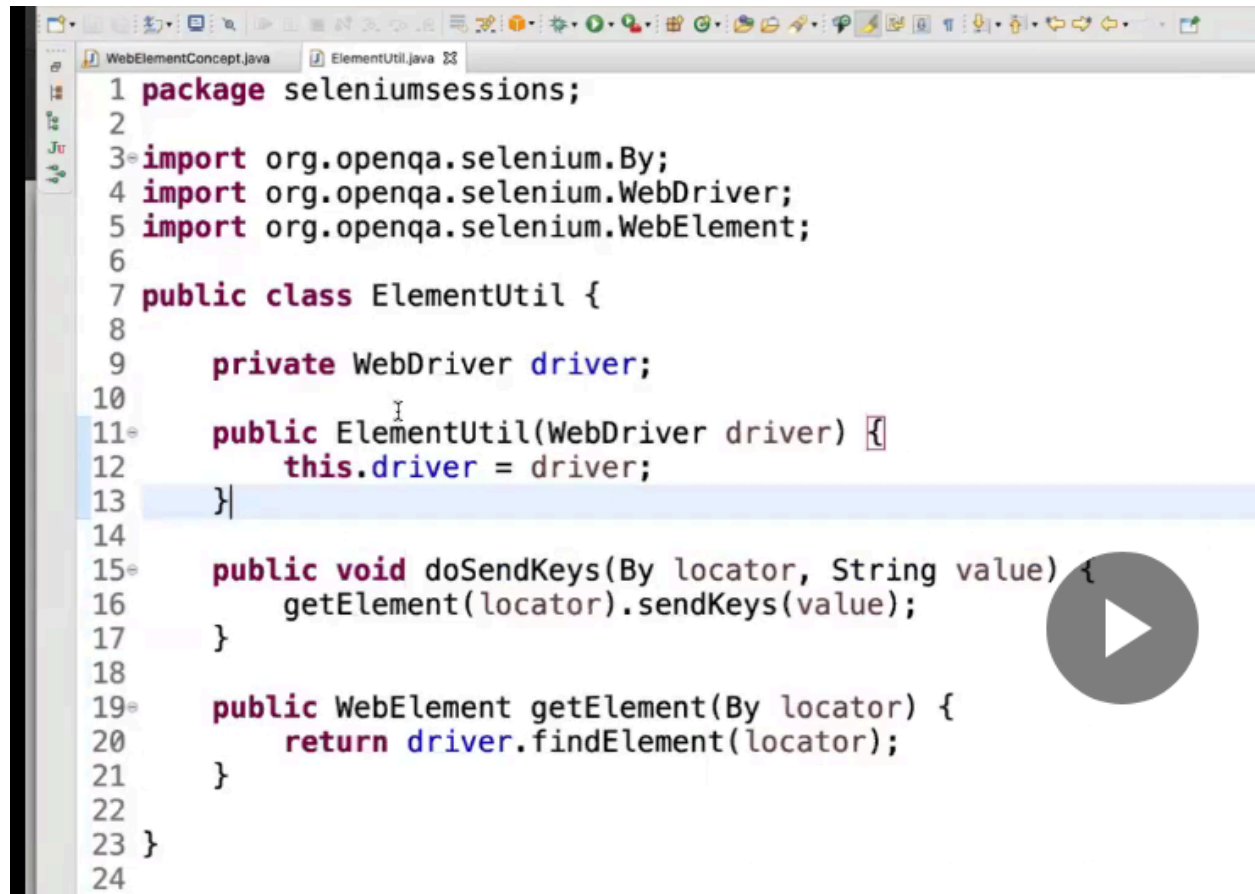
47 //5. By locator + generic function to get the element and perform action
48 By emailField = By.id("input-email");
49 By pwdField = By.id("input-password");
50
51 doSendKeys(emailField, "tom@gmail.com");
52 doSendKeys(pwdField, "tom@123");
53
54 }
55
56 public static void doSendKeys(By locator, String value) {
57     getElement(locator).sendKeys(value);
58 }
59
60 public static WebElement getElement(By locator) {
61     return driver.findElement(locator);
62 }
63
64 }

```

Sixth approach-

Create element util class.

Create constructor to access this class.

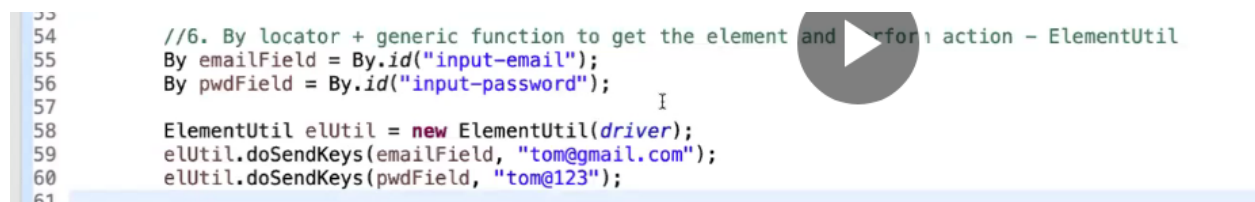


```
1 package seleniumsessions;
2
3 import org.openqa.selenium.By;
4 import org.openqa.selenium.WebDriver;
5 import org.openqa.selenium.WebElement;
6
7 public class ElementUtil {
8
9     private WebDriver driver;
10
11     public ElementUtil(WebDriver driver) {
12         this.driver = driver;
13     }
14
15     public void doSendKeys(By locator, String value) {
16         getElement(locator).sendKeys(value);
17     }
18
19     public WebElement getElement(By locator) {
20         return driver.findElement(locator);
21     }
22 }
23
24
```

Come back to webelement class-

Change code.

Call from element util.



```
54 //6. By locator + generic function to get the element and perform action - ElementUtil
55 By emailField = By.id("input-email");
56 By pwdField = By.id("input-password");
57
58 ElementUtil eUtil = new ElementUtil(driver);
59 eUtil.doSendKeys(emailField, "tom@gmail.com");
60 eUtil.doSendKeys(pwdField, "tom@123");
61
```

## Login page test-

```
1 package seleniumsessions;
2
3 import org.openqa.selenium.By;
4 import org.openqa.selenium.WebDriver;
5
6 public class LoginPageTest {
7
8     public static void main(String[] args) {
9
10         BrowserUtil brUtil = new BrowserUtil();
11         WebDriver driver = brUtil.initDriver("chrome");
12
13         brUtil.launchURL("https://naveenautomationlabs.com/opencart/index.php?route=account/login");
14
15         System.out.println(brUtil.getPageTitle());
16
17         By emailField = By.id("input-email");
18         By pwdField = By.id("input-password");
19
20         ElementUtil eleUtil = new ElementUtil(driver);
21
22         eleUtil.doSendKeys(emailField, "tom@gmail.com");
23         eleUtil.doSendKeys(pwdField, "tom@123");
24
25         //brUtil.quitBrowser();
26
27     }
28 }
```

Web element concept class-  
Seventh approach is

```
60         eleUtil.doSendKeys(pwdField, "tom@123");
61
62         //7. BrUtil + eleUtil --> we+action
63
64     }
```