

# Selenium Questions

## Generic Selenium Questions

### **Question 1:**

#### **What is Selenium?**

Answer:

Selenium is a browser automation tool which lets you automated operations like – type, click, and selection from a drop down of a web page.

### **Question 2:**

#### **How is Selenium different from commercial browser automation tools?**

Answer:

Selenium is a library which is available in a gamut of languages i.e. java, C#, python, ruby, php etc while most commercial tools are limited in their capabilities of being able to use just one language. More over many of those tools have their own proprietary language which is of little use outside the domain of those tools. Most commercial tools focus on record and replay while Selenium emphasis on using Selenium IDE (Selenium record and replay) tool only to get acquainted with Selenium working and then move on to more mature Selenium libraries like Remote control (Selenium 1.0) and Web Driver (Selenium 2.0).

Though most commercial tools have built in capabilities of test reporting, error recovery mechanisms and Selenium does not provide any such features by default. But given the rich set of languages available with Selenium it very easy to emulate such features.

### **Question3:**

#### **What are the set of tools available with Selenium?**

Answer:

Selenium has four set of tools – Selenium IDE, Selenium 1.0 (Selenium RC), Selenium 2.0 (WebDriver) and Selenium Grid. Selenium Core is another tool but since it is available as part of Selenium IDE as well as Selenium 1.0, it is not used in isolation.

**Question 4:****Which Selenium Tool should I use?**

Answer:

It entirely boils down to where you stand today in terms of using Selenium. If you are entirely new to Selenium then you should begin with Selenium IDE to learn Selenium location strategies and then move to Selenium 2 as it is the most stable Selenium library and future of Selenium. Use Selenium Grid when you want to distribute your test across multiple devices. If you are already using Selenium 1.0 then you should begin to migrate your test scripts to Selenium 2.0

**Question 5:****What are the element locators available with Selenium which could be used to locate elements on web page?**

Answer:

- ID
- Name
- ClassName
- XPath
- CSS
- LinkText
- PartialLinkText
- TagName

**Question 6:****What is Selenium Grid?**

Answer:

Selenium grid lets you distribute your tests on multiple machines and all of them at the same time. Hence you can execute test on IE on Windows and Safari on Mac machine using the same test script (well, almost always). This greatly helps in reducing the time of test execution and provides quick feedback to stack holders.

**Question 7:****What is XPath? When would I have to use XPath in Selenium IDE?**

Answer:

XPath is a way to navigate in xml document and this can be used to identify elements in a web page. You may have to use XPath when there is no name/id associated with element on page or only partial part of name/ide is constant.

Direct child is denoted with - /

Relative child is denoted with - //

Id, class, names can also be used with XPath –

- //input[@name='q']
- //input[@id='lst-ib']
- //input[@class='lst']

If only part of id/name/class is constant than “contains” can be used as –

- //input[contains(@id,'lst-ib')]

### **Question 8:**

**What is CSS location strategy in Selenium?**

Answer:

CSS location strategy can be used with Selenium to locate elements, it works using cascade style sheet location methods in which -

Direct child is denoted with – (a space)

Relative child is denoted with - >

Id, class, names can also be used with XPath –

- css=input[name='q']
- css=input[id='lst-ib'] or input#lst-ib
- css=input[class='lst'] or input.lst

If only part of id/name/class is constant than “contains” can be used as –

- css=input[id\*='lst-ib ')]

Element location strategy using inner text

- css = a:contains('log out')

### **Question 9:**

**There is id, name, XPath, CSS locator, which one should I use?**

Answer:

If there are constant name/id available than they should be used instead of XPath and CSS locators. If not then css locators should be given preference as their evaluation is faster than XPath in most modern browsers.

## **Web Driver (Selenium) Questions**

### **Question 1:**

**What is Selenium?**

Answer

Selenium 2.0 is consolidation of two web testing tools – Selenium RC and WebDriver, which claims to give best of both worlds – Selenium and WebDriver. Selenium 2.0 was officially released only of late.

**Question 2:**

**Which web driver implementation is fastest?**

Answer

HTMLUnitDriver. Simple reason is HTMLUnitDriver does not execute tests on browser but plain http request – response which is far quick than launching a browser and executing tests. But then you may like to execute tests on a real browser than something running behind the scenes

**Question 3:**

**How do I click an element in Selenium?**

Answer

You can use “click” method on element to submit form –

```
element.click();
```

Alternatively you can use click method on the element which does form submission.

**Question 4:**

**Can I simulate pressing key board keys using Selenium?**

Answer

You can use “sendKeys” command to simulate key board keys as –

```
element.sendKeys(" and some", Keys.ARROW_UP);
```

You can also use “sendKeys” to type in text box as –

```
HTMLelement.sendKeys("testData");
```

**Question 5:**

**How can I get the text from the screen?**

Answer

```
HTMLelement.getText();
```

**Question 6:****How can I get the attribute value from the html tag?**

Answer

```
HTMLElement.getAttributeValue("value");
```

**Question 7:****How do I open the developer option in chrome**

Answer

Press F12

**Question 8:****How do I clear content of a text box in Selenium 2.0**

Answer

You can use “clear” method on text box element to clear its content –

```
textBoxElement.clear();
```

**Question 9:****How do I select a drop down value using Selenium?**

Answer

To select a drop down value, you first need to get the select element using one of element locator and then you can select element using visible text –

Select selectElement

```
= new Select(driver.findElement(By.cssSelector("cssSelector")));  
selectElement.selectByVisibleText("India");
```

Different methods

```
selectByValue()
```

```
selectByIndex
```

**Question 10:****How can I get the current window handle?**

Answer

```
driver.getWindowHandle();
```

**Question 11:****What are offering to deal with popup windows while using Selenium?**

Answer

You can use “switchTo” window method to switch to a window using window name. There is also one method “getWindowHandles” which could be used to find all Window handles and subsequently bring control on desired window using window handle –

```
webDriver.switchTo().window("windowName");  
for (String handle : driver.getWindowHandles()) {  
    driver.switchTo().window(handle);  
}
```

### **Question 12:**

#### **How about handling frames using Selenium?**

Answer

You can use “switchTo” frame method to bring control on an HTML frame –

```
driver.switchTo().frame("frameName");
```

You can also use index number to specify a frame –

```
driver.switchTo().frame("parentFrame.4.frameName");
```

This would bring control on frame named – “frameName” of the 4<sup>th</sup> sub frame names “parentFrame”

### **Question 13:**

#### **Can I navigate back and forth in a browser in Selenium ?**

Answer

You can use Navigate interface to go back and forth in a page. Navigate method of WebDriver interface returns instance of Navigation. Navigate interface has methods to move back, forward as well as to refresh a page –

```
driver.navigate().forward();  
driver.navigate().back();  
driver.navigate().refresh();
```

### **Question 14:**

#### **What is the order of fastest browser implementation for WebDriver?**

Answer

HTMLUnitDriver is the fastest browser implementation as it does not involves interaction with a browser, This is followed by Firefox driver and then IE driver which is slower than FF driver and runs only on Windows.

**Question 15:**

**Are there any limitations while injecting capabilities in WebDriver to perform tests on a browser which is not supported by WebDriver?**

Answer

Major limitation of injecting Capabilities is that “fundElement” command may not work as expected. This is because WebDriver uses Selenium Core to make “Capability injection” work which is limited by java script security policies.

**Question 16:**

**Can I change User-Agent while using FF browser? I want to execute my tests with a specific User-Agent setting.**

Answer

You can create FF profile and add additional Preferences to it. Then this profile could be passed to Firefox driver while creating instance of Firefox –

```
FirefoxProfile profile = new FirefoxProfile();
```

```
profile.addAdditionalPreference("general.useragent.override", "User Agent String");
```

```
WebDriver driver = new FirefoxDriver(profile);
```

**Question 17:**

**Is there any difference in XPath implementation in different WebDriver implementations?**

Answer

Since not all browsers (like IE) have support for native XPath, WebDriver provides its own implementation for XPath for such browsers. In case of HTMLUnitDriver and IEDriver, html tags and attributes names are considered lower cased while in case of FF driver they are considered case in-sensitive.

**Question 18**

**My application uses ajax highly and my tests are suffering from time outs while using Selenium.**

Answer

You can state WebDriver to implicitly wait for presence of Element if they are not available instantly. By default this setting is set to 0. Once set, this value stays till the life span of WebDriver object. Following example would wait for 60 seconds before throwing ElementNotFound exception –

```
WebDriver driver = new FirefoxDriver();
```

```
driver.manage().timeouts().implicitlyWait(60, TimeUnit.SECONDS);
```

```
WebElement element = driver.findElement(By.id("elementID"));
```

**Question 19:**

**What if I don't want to use implicit wait and want to wait only for presence of certain elements?**

Answer

You can use explicit wait in this situation to wait for presence of certain element before continuing with test execution. You can use “WebDriverWait” and “ExpectedCondition” to achieve this –

```
WebDriver driver = new FirefoxDriver();  
WebElement myDynamicElement = (new WebDriverWait(driver,  
60)).until(new ExpectedCondition<WebElement>(){
```

@Override

```
    public WebElement apply(WebDriver d) {  
        return d.findElement(By.id("myDynamicElement"));  
    }  
});
```

This is going to wait up to 60 seconds before throwing ElementNotFoundException exception.

**Question 20:**

**What are the types of waits supported by WebDriver?**

Implicit wait - Implicit wait commands Selenium to wait for a certain amount of time before throwing a “No such element” exception.

```
driver.manage().timeouts().implicitlyWait(TimeOut, TimeUnit.SECONDS);
```

Explicit wait - Explicit wait is used to tell the Web Driver to wait for certain conditions before throwing an "ElementNotVisibleException" exception.

```
WebDriverWait wait = new WebDriverWait(WebDriver Reference, TimeOut);
```



Fluent wait - It is used to tell the web driver to wait for a condition, as well as the frequency with which we want to check the condition before throwing an "ElementNotVisibleException" exception.

```
Wait wait = new FluentWait(WebDriver reference).withTimeout(timeout, SECONDS).pollingEvery(timeout, SECONDS).ignoring(Exception.class);
```

**Question 21: What is the major difference between driver.close() and driver.quit()?**

Answer:

```
driver.close()
```

This command closes the browser's current window. If multiple windows are open, the current window of focus will be closed.

```
driver.quit()
```

When quit() is called on the driver instance and there are one or more browser windows open, it closes all the open browser windows.

**Question 22: When do we use findElement() and findElements()?**

The findElement method returns only the first element from the DOM. Its return type is "WebElement". It throws "NoSuchElementException" if the element is not found. The findElements method returns the List<WebElement> . The list size is 0 in case no element is found.

**Question 23: What is the difference between single and double slash in Xpath?**

Single slash is used to create Xpath with absolute path i.e. the xpath would be created to start selection from the start node.

```
/html/body/div[2]/div[1]/div[1]/a
```

Double slash is used to create Xpath with relative path i.e. the xpath would be created to start selection from anywhere within the document

```
//div[class="qa-logo"]/a
```

#### **Question 24:**

What is RemoteWebDriver? When would I have to use it?

Answer

RemoteWebDriver is needed when you want to use HTMLUnitDriver. Since HTMLUnitDriver runs in memory, you would not see a browser getting launched –

```
// Create HTMLUnitDriver instance
WebDriver driver = new HtmlUnitDriver();

// Launch Yahoo.com
driver.get("http://www.yahoo.com");
```

#### **Question 25: How do I handle java script alert using WebDriver?**

Answer

WebDriver would support handling js alerts using Alert interface.

```
// Bring control on already opened alert
Alert alert = driver.switchTo().alert();

// Get the text of the alert or prompt
alert.getText();

// Click ok on alert
alert.accept();
```

#### **Question 26: Could I safely execute multiple instances of WebDriver implementations?**

Answer

As far as HTMLUnitDriver and FF drivers are concerned, each instance would be independent of other. In case of IE driver there could be only one instance of IE driver

running on Windows. If you want to execute more than one instance of IE driver then you should consider using RemoteWebDriver and virtual machines.

### **Question 27: Is it possible to interact with hidden elements using WebDriver?**

Answer

Since WebDriver tries to exercise browser as closely as real users would, hence simple answer is No, But you can use java script execution capabilities to interact with hidden elements.

### **Question 28: How do I execute Java Script in Selenium?**

Answer

You need to use JavaScriptExecutor to execute java script in Selenium 2.0, For example if you want to find tag name of an element using Selenium 2.0 then you can execute java script as following –

```
WebElement element = driver.findElement(By.id("elementLocator"));
String name = (String) ((JavascriptExecutor) driver).executeScript(
    "return arguments[0].tagName", element);
```

### **Question 29: What are the advantages of using Remote WebDriver over WebDriver?**

Answer

You can use Remote WebDriver when –

- When you want to execute tests on a browser not available to you locally
- Introduction to extra latency to tests

But there is one disadvantage of using Remote WebDriver that you would need external servlet container.

### **Question 30: Can you show me code example of using Remote WebDriver?**

Answer

// Any driver could be used for test

```
DesiredCapabilities capabilities = new DesiredCapabilities();
```

```
// Enable javascript support
capabilities.setJavascriptEnabled(true);
```

```
// Get driver handle
```

```
WebDriver driver = new RemoteWebDriver(capabilities);
```

```
// Launch the app
driver.get("http://www.google.com");
```

### Question 31:

**Is it possible to emulate a browser with HTMLUnitDriver?**

Answer

You can emulate browser while using HTMLUnitDriver but it is not recommended as applications are coded irrespective of browser you use. You could emulate Firefox 3 browser with HTMLUnitDriver as –

```
HtmlUnitDriver driver = new HtmlUnitDriver(BrowserVersion.FIREFOX_3);
```

Or you can inject desired capabilities while instantiating HTMLUnitDriver as –

```
HtmlUnitDriver driver = new HtmlUnitDriver(capabilities);
```

### Question 32:

**Can I carry out multiple operations at once while using WebDriver?**

Answer

You can use Builder pattern to achieve this. For example if you want to move an element from one place to another you can use this –

```
Actions builder = new Actions(driver);
```

```
Action dragAndDrop = builder.clickAndHold(element)
    .moveToElement(otherElement)
    .release(otherElement)
    .build();
```

```
dragAndDrop.perform();
```

### Question 33:

**How do I simulate keyboard keys using WebDriver?**

Answer

There is a Keyboard interface which has three methods to support keyboard interaction –

- sendKeys(CharSequence)- Sends character sequence
- pressKey(Keys keyToPress) - Sends a key press without releasing it.
- releaseKey(Keys keyToRelease) - Releases a modifier key

**Question 34:**  
**What about Mouse Interaction?**

Answer

Mouse interface lets you carry out following operations –

- click(WebElement element) – Clicks an element
- doubleClick(WebElement element) - Double-clicks an element.
- void mouseDown(WebElement element) - Holds down the left mouse button on an element.
- mouseUp(WebElement element) - Releases the mouse button on an element.
- mouseMove(WebElement element) - Moves element from current location to another element.
- contextClick(WebElement element) - Performs a context-click (right click) on an element.

**Question 35: What is the use of DesiredCapabilities in Selenium?**

Answer:

The desired capability is a series of key/value pairs that stores the browser properties like browsername, browser version, the path of the browser driver in the system, etc. to determine the behaviour of the browser at run time.

Desired capability can also be used to configure the driver instance of Selenium WebDriver.

More info: <https://www.guru99.com/desired-capabilities-selenium.html>

### **Question 36: What is the architecture of Selenium**

#### **Answer:**

Selenium WebDriver API helps in communication between languages and browsers. Each and every browser has different logic of performing actions on the browser. Below image depicts various components of Selenium WebDriver Architecture.

1. Selenium Client Library
2. JSON WIRE PROTOCOL Over HTTP Client: It is used to transfer data between a server and a client on the web. JSON Wire Protocol is a REST API that transfers the information between HTTP server
3. Browser Drivers: Each browser contains a separate browser driver. Browser drivers communicate with the respective browser without revealing the internal logic of the browser's functionality.
4. Browsers

### **Selenium Tool Implementation Misc Questions**

Question 169:

How do I implement data driven testing using Selenium?

Answer

Selenium, unlike others commercial tools does not have any direct support for data driven testing. Your programming language would help you achieving this. You can use jxl library in case of java to read and write data from excel file. You can also use Data Driven Capabilities of TestNG to do data driven testing.

Question 170:

What is equivalent to test step, test scenario and test suite in Selenium.

Answer

If you are using Java client driver of Selenium then TestNG test method could be considered equivalent to test step, a test tag could be considered as test scenario and a suite tag could be considered equivalent to a test suite.

Question 171:

How do I get attribute value of an element in Selenium?

Answer

You could use getAttribute method

With Selenium 1.0 –

```
String var = selenium.getAttribute("css=input[name='q']@maxlength");  
System.out.println(var);
```

With Selenium 2.0 –

String var =

```
webdriver.findElement(By.cssSelector("input[name='q']")).getAttribute("maxlength")  
System.out.println(var);
```

Question 172:

How do I do database testing using Selenium?

Answer

Selenium does not support database testing but your language binding does. For example while using java client driver you can use java data base connectivity (jdbc) to establish connection to data base, fetch/write data to data base and doing data comparison with front end.

Question 173:

I completed test execution and now I want to email test report.

Answer

If you are using “ant” build tool then you can use “mail” task to deliver your test results. Similar capabilities are available in Continuous Build Integration tools like – Hudson.

Question 174:

How do I make my tests more comprehensible?

Answer

Selenium tests which are written as –

```
selenium.click("addForm:_ID74:_ID75:0:_ID79:0:box");
```

Make it tough to understand the element which is being exercised upon.

Instead of hard coding element locator in tests you should externalize them. For example with java you can use properties file to contain element locators and then locator reference is given in test script. Following this approach previous test step would look as –

```
selenium.click(PolicyCheckbox);
```

And this is far more comprehensible.

Question 175:

Why should I use Page Object?

Answer

Page object is a design pattern which distinguishes the code carrying out operations on page and code which carries out tests (assertion/verification). While implementing page object you abstract functioning of a page or part of it in a dedicated “Classs” which is then used by test script to perform actions on page and reach a stage when actual test could be performed.

Advantage of using page object is the fact that if application lay out changes then you only need to modify the navigation part and test would function intact.

## **Selenium Grid Questions**

Question 91:

How do I cut down test execution time for my selenium tests? I want to execute my tests on a combination of different machines and browsers.

Answer



Selenium grid is your friend ☺ . Selenium grid lets you distribute tests across browsers and machines of your choice.

Question 92:

How does Selenium grid works?

Answer

Selenium grid uses combination of Selenium RC servers to execute tests in multiple browsers on different machine. Herein one Selenium RC server works as hub while other RC servers work as slaves, which could be controlled by hub. Whenever there is a request for a specific configuration for test execution then hub looks for a free RC slave server and if available then test execution begins on it. Once test execution is over then RC slave server would be available for next set of test execution.

Question 93:

Can you show me one diagram which describes functionality of Selenium grid?

Answer

In the following diagram Selenium hub is controlling three Selenium RC servers which are running for configurations –

- IE on Windows
- FF on Linux
- FF on windows

Question 94:

Which jar files are needed to works with Selenium GRID?

Answer

You need to download and add following jar files to your Selenium set up to be able to work with Selenium. These jar files are –

- selenium-grid-remote-control-standalone-.jar
- selenium-grid-hub-standalone-.jar
- selenium-grid-tools-standalone-.jar

Question 95:

How do I start Selenium Grid hub from my machine?

Answer

You should have “ant” set up on your system to be able to work with Grid. Once you have downloaded Selenium Grid, navigate to its distribution directory and execute following command -

ant launch-hub

This would start grid hub on port 4444 locally. You can verify this by navigating to following URL - <http://localhost:4444/console>

Question 96:

How do I start Selenium Grid Slave Server from my system?

Answer

Navigate to download directory of Selenium grid and execute following command –  
`ant launch-remote-control`

This would launch remote control server at port 555. At this point if you navigate to <http://localhost:4444/console> then you would see this remote control listed under “Available Remote Controls”

Question 97:

How do I start Selenium grid slave on a different port than 5555?

Answer

You can use option “-Dport” followed by port number to start grid slave on a specific port.

`ant -Dport=1111 launch-remote-control`

`ant -Dport=2222 launch-remote-control`

Question 98:

How do I start grid RC slaves on a different system than my local host so that hub could control and contact a specific configuration?

Answer

You should specify following configuration while starting RC slave –

`ant -Dport= -Dhost= -DhubURL= launch-remote-control`

Herein “hostname” is the host where RC slave is running and “hub url” is URL of machine where grid hub is running.

Question 99:

How do I specify an environment while starting grid slave machine?

Answer

You could specify an environment using “-Denvironment” while starting a slave machine.

`ant -Denvironment="Safari on Mac" launch-remote-control`

Herein Safari on Mac is the environment which would be used to recognize configuration of RC slave.

Question 100:

How do I use machine specific configuration in my Selenium tests?

Answer

You could specify machine specific configuration while instantiating Selenium as –  
`Selenium = new DefaultSelenium("localhost", 4444, '**Safari on Mac**', 'http://yahoo.com');`

And then you use this selenium instance to carryout operation on your web application.

Question 101:

But how does my tests know that ‘Safari on Mac’ mean a safari browser? How does mapping between names like ‘Safari on Mac’ and original browser options available in Selenium work?

Answer

Selenium grid uses a file called “grid\_configuration.yml” which defines configurations of all browsers. You would have to add this in your project. This file looks like –

Question 102:

How does Selenium grid hub keeps in touch with RC slave machine?

Answer

Selenium grid hub keeps polling all RC slaves at predefined time to make sure they are available for testing. If not then Selenium hub disconnect any unavailable RC slaves and makes it clear that any RC slave is not available for testing. The deciding parameter is called – “remoteControlPollingIntervalInSeconds” and is defined in “grid\_configuration.yml” file.

Question 103:

My RC becomes unresponsive at times and my Selenium grid hub keeps waiting for RC slave to respond<sup>Ⓢ</sup>. How do I let hub know to give up on RC slave after a certain time?

Answer

You could state Selenium grid hub to wait for RC slave for a predefined time, and if RC slave does not responds with in this time then hub disconnects test execution on that slave. This parameter is called “sessionMaxIdleTimeInSeconds” and this parameter can be defined in “grid\_configuration.yml” file.

Question 104:

What if my hub goes down while Selenium RC slaves are up and running?

Answer

There is one heart beat mechanism from RC slave to hub which is reciprocal to mechanism used by hub to slave machines. RC slaves use a parameter called “hubPollerIntervalInSeconds” to keep track of running grid hub. This parameter can be defined while starting the hub as –

ant -DhubPollerIntervalInSeconds= launch-hub

if hub does not respond within this time then RC slaves deregister themselves from hub.

Question 105:

Can Selenium grid be used for performance testing?

Answer

Selenium grid is for functional testing of application across different configuration. Performance testing is usually not carried out on actual devices but on a simulated http request/response mechanism. If you want to use Selenium Grid for performance testing then you would have to invest heavily on s/w and h/w infrastructure.

Question 106:

Are there additional logs available while working with Selenium grid?

Answer

You can find Selenium grid hub logs in “log/hub.log” and Remote Control logs in “log/rc-\*.log” folder.

Question 107:

There are various options available while starting a Selenium server, how do I use them with Selenium grid?

Answer

You can use “seleniumArgs” Java property while launching the remote control and specify any of the option which you would with normal Selenium set up. For example you can run Selenium RC slave in single window mode using following command –  
ant -DseleniumArgs="-singleWindow -debug" launch-remote-control

Question 108:

I see duplicate entries in my hub console, same RC slave listed more than once :-O

Answer

This is because you are killing RC slave very ferociously. For example you if you just close the console without actually ending the process in more civil manner. To avoid this you should kill all RC slaves in more civil manner. If you encounter this error then you should restart Selenium hub and RC slaves would then register themselves to hub.

Question 109:

How do I specify my corporate proxy while starting Selenium grid hub or slave machines?

Answer

You could use setting for “http.proxyHost” and “http.proxyPort” while starting hub and remote control machines –

ant -Dhttp.proxyHost= -Dhttp.proxyPort= launch-hub

ant -Dhttp.proxyHost= -Dhttp.proxyPort= launch-remote-control

Question 110:

How do I use Selenium Grid while using Java, .Net or Ruby

Answer

With java you can take advantage of parallel testing capabilities of TestNG to drive your Selenium grid tests

With .Net you can use “Gallio” to execute your tests in parallel

With Ruby you can use “DeepTest” to distribute your tests

Question 111:

How about the test report when I use Selenium grid for test execution?

Answer

This entirely boils down to framework you use to write your tests. For example in case of java, TestNG reports should suffice.