

Selenium Interview Questions

Question 1: How do you start Selenium RC from the command line?

Answer: Selenium command line example

```
// Simple way to start Selenium RC is-  
  
java -jar selenium-server.jar  
  
// Run a suite of Selenese scripts in a browser  
  
java -jar selenium-server.jar -htmlSuite
```

Question 2: On my machine port 4444 is not free. How can I use another port?

Answer: Selenium runs on specific port

```
// You can specify port while running the selenium server as -  
  
Java -jar selenium-server.jar -port 5555
```

Question 3: What is Selenium server and how does it differ from Selenium hub?

Answer: Selenium server is a standalone application for using a single server as a test node. Selenium hub acts as a proxy in front of one or more Selenium node instances. A hub + node(s) is called a Selenium grid. Running Selenium server is similar to creating a Selenium grid from a hub and a single node on the same host.

Question 4: How do you connect to Database from selenium?

Answer: Selenium is a Web UI automation tool. It doesn't provide any API to establish a database connection. It's up to the programming language which you are using with Selenium for automation. In the below example, we assume that Java is being used.

A Connection object represents a connection with a database. When we connect to a database by using connection method, we create a Connection Object, which represents the connection to the database. An application may have one or more than one connections with a single database or many connections with different databases.

We can use the Connection object for the following things:

1. It creates the Statement, PreparedStatement and CallableStatement objects for executing the SQL statements.
 2. It helps us to Commit or roll back a JDBC transaction.
 3. If you want to know about the database or data source to which you are connected then the Connection object gathers information about the database or data source by the use of DatabaseMetaData.
 4. It helps us to close the data source. The Connection.isClosed() method returns true only if the Connection.close() has been called. This method is used to close all the connection.
- Firstly we need to establish the connection with the database. This is done by using the method DriverManager.getConnection(). This method takes a string containing a URL. The DriverManager

Selenium Interview Questions

class attempts to locate a driver that can connect to the database represented by the string URL. Whenever the getConnection() method is called the DriverManager class checks the list of all registered Driver classes that can connect to the database specified in the URL.

Syntax:

Selenium database connection example

```
String url = "jdbc: odbc: makeConnection";

Connection con = DriverManager.getConnection(url, "userID",
"password");
```

Question 5: What are the locators available in Selenium RC?

Answer:

1. ID
2. Name
3. CSS (Cascade style sheet)
4. XPATH (Relative XPath and Absolute XPath)
5. Dom

Question 6: How do you verify an object present on multiple pages?

Answer: Check on each page assertTrue(selenium.isElementPresent(locator));

Question 7: What is the difference between a single and double slash used in XPath?

Answer:

If XPath starts selection from the document node, it'll allow you to create 'absolute' path expressions.

e.g. `"/html/body/p"` matches all the paragraph elements

If XPath starts selection matching anywhere in the document, then it'll allow you to create 'relative' path expressions.

e.g. `"//p"` matches all the paragraph elements

Question 8: How will you write a User Extension for Selenium IDE/RC?

Answer:

User extensions are stored in a separate file that Selenium IDE or Selenium RC uses to activate the extensions. It comprises of function definitions which are written in JavaScript.

Because Selenium's core is developed in JavaScript, creating an extension follows the similar standard rules for prototypal languages. To create an extension, we have to write a function in the following design format.

Selenium user extension sample

Selenium Interview Questions

```
// sample
Selenium.prototype.doFunctionName = function() {

}
```

The “do” in front of the function name tells Selenium that this function can be called as a command for a step instead of an internal or private function.

Question 9: How do you verify the presence of an element after the successful page loading?

Answer:

It can be achieved with the following line of code.

Just mention some time value to check the element (in seconds) like:

Java Sample

```
public void waitForElementPresent(String element, int timeout)
    throws Exception {

    for (int second = 0;; second++) {

        if (second >= timeout)

            fail("Timeout. Unable to find the Specified element" + element);

        try {

            if (selenium.isElementPresent(element))
                break;

        } catch (Exception e) {

        }

        Thread.sleep(1000);

    }

}
```

Question 10: What do you know about Selenium Grid? What capabilities does it provide?

Answer: Selenium Grid is a tool that dramatically speeds up functional testing of web-apps by leveraging your existing computing infrastructure. It allows you to easily run multiple tests in parallel, on multiple machines, in a heterogeneous environment.

Based on the excellent Selenium web testing tool, Selenium Grid allows you to run multiple instances of Selenium Remote Control in parallel. Even better, it makes all these Selenium Remote Controls appear as a single one, so your tests do not have to worry about the actual infrastructure. Selenium Grid cuts down on the time required to run a Selenium test suite to a fraction of the time that a single instance of Selenium instance would take to run.

Selenium Interview Questions

Question 11: How will you start the selenium server from your Java class?

Answer:

Selenium server sample

```
try {  
  
    seleniumServer = new SeleniumServer();  
  
    seleniumServer.start();  
  
} catch (Exception e) {  
  
    e.printStackTrace();  
  
}
```

Question 12: What are the verification points available in Selenium?

Answer:

There are largely three types of verification points available with Selenium –

- Check for page title
- Check for certain text
- Check for certain element (text box, drop down, table etc.)

Question 13: What is XPath? When should you use XPath in Selenium?

Answer:

XPath is a way to navigate in an HTML/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 an element on the page or only a partial part of name/id 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 14: What is CSS locator strategy in Selenium? Explain with example.

Answer:

Selenium Interview Questions

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 symbol)

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 15: There are many locators like id, name, XPath, CSS locator, which one should I use?

Answer:

If there are unique names or identifier available then 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.

Question 16: What is the mechanism to handle multiple popups in selenium?

Answer:

Multiple popups can be handled by using the command `getWindowHandles()`.

Then store all the window names into `Set<String>` variable and transform it into an array.

Next, by using the array index, you can navigate to specific window by using

```
driver.switchTo().window(ArrayIndex);
```

Question 17: How do you handle Ajax controls using selenium?

Answer:

Let's consider an example. Say the google test box which is an ajax control and when we enter some text into it, then it displays the auto-suggested values.

To work with such controls, you need to capture all the suggested values in a string after entering the value in the text box. Then, just split the string and take the values.

Question 18: What are the advantages of Selenium Web driver over Selenium RC?

Answer:

Selenium Interview Questions

- Selenium RC's architecture is quite complicated while WebDriver's architecture is simpler than Selenium RC's.
- Though Selenium RC is slower since it uses an additional JavaScript program called Selenium Core. On the contrary, WebDriver is faster than Selenium RC since it speaks directly to the browser and uses browser's own engine to control it.
- Selenium Core, just like other JavaScript codes, can access disabled elements. Web Driver interacts with page elements in a more realistic way.
- Selenium RC's API set is already evolved but contains redundancies and often confusing commands. WebDriver APIs are simpler and do not contain any redundant or confusing commands.
- Selenium RC cannot support the headless HtmlUnit browser. It needs a real, visible browser to operate on. Web Driver can support the headless HtmlUnit browser.
- Selenium RC has built-in test result generator and it automatically generates an HTML file of test results. Web Driver has no built-in command that automatically generates a Test Results File.

Question 19: State any difference between "GET" and "NAVIGATE"?

Answer:

Get method will get a page to load or get page source or get the text that's all. Whereas navigate will guide through the history like refresh, back, forward. For example, if we want to move forward and do some functionality and back to the home page. This can be achieved through navigate() method only. driver.get() will wait till the whole page gets loaded and driver.navigate() will just redirect to that page and will not wait.

Question 20: How is the implicit Wait different from Explicit wait?

Implicit Wait sets internally a timeout that will be used for all consecutive Web Element searches. It will try lookup the element again and again for the specified amount of time before throwing a NoSuchElementException if the element could not have been found. It does only this and can't be forced into anything else – it waits for elements to show up.

Explicit Wait or just Wait is a one-timer used by you for a particular search. It is more extensible in the means that you can set it up to wait for any condition you might like. Usually, you can use some of the prebuilt Expected Conditions to wait for elements to become clickable, visible, invisible, etc., or just write your own condition that suits your needs.

Question 21: How to Handle Alerts/Popups in Selenium WebDriver?

Answer:

There are two types of alerts which are commonly referred–

1. Windows based alert pop ups
2. Web-based alert pop ups

Web-based alert pop-ups.

1. WebDriver offers the users with a very efficient way to handle these pop-ups using Alert interface.

Selenium Interview Questions

2. void dismiss() – The dismiss() method clicks on the “Cancel” button as soon as the popup window appears.
3. void accept() – The accept() method clicks on the “Ok” button as soon as the popup window appears.
4. String getText() – The getText() method returns the text displayed on the alert box.
5. void sendKeys(String stringToSend) – The sendKeys() method enters the specified string pattern into the alert box.

Windows based alert pop-ups.

Handling window based pop-ups have always been a little tricky as we know Selenium is an automation testing tool which supports only web application testing, that means, it doesn't support windows based applications and window alert is one of them.

1. Robot class is a java based utility which emulates the keyboard and mouse actions and can be effectively used to handling window based pop up with the help of keyboard events.
2. The keyPress and keyRelease methods simulate the user pressing and releasing a certain key on the keyboard respectively.

Question 22: How to take a screenshot with Selenium WebDriver?

Answer:

Selenium capture screenshot

```
Import org.apache.commons.io.FileUtils;  
  
WebDriver driver = new FirefoxDriver();  
  
driver.get("http://www.google.com/");  
  
File scrFile = ((TakesScreenshot)driver).getScreenshotAs(OutputType.  
FILE);  
  
// Now you can do whatever you need to do with it, for example copy  
somewhere  
  
FileUtils.copyFile(scrFile, new File("c:\\tmp\\screenshot.png"));
```

Question 23: How can I address SSL Certificate issue in Firefox with WebDriver (or) how to manage the secured connection error in HTTPS?

Answer:

SSL Certificate issue in Firefox with WebDriver

```
FirefoxProfile profile = new FirefoxProfile();  
  
profile.setAcceptUntrustedCertificates(true);
```

Selenium Interview Questions

```
profile.setAssumeUntrustedCertificateIssuer(false);
```

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

Question 24: How can I fix SSL certification issue in IE?

Answer:

Handle SSL certification issue in IE

```
// Add the below command after opening the browser.
```

```
<strong>driver.navigate().to("javascript:document.getElementById('overridelink').click()");</strong>
```

Question 25: What are the available locators in Selenium WebDriver?

Answer:

1. ID,
2. Name,
3. CSS,
4. XPath,
5. Classname,
6. TagName,
7. LinkText, and
8. Partial Link Text.

Question 26: How can I handle AJAX controls in WebDriver?

Answer:

AJAX stands for Asynchronous JavaScript and XML. It does not rely on the extra overhead of opening and closing tags that is needed to create valid XML. Most of the time WebDriver automatically handle the Ajax controls and calls/ Incase if it is not able to handle, you can follow the below way to handle.

Waiting for Ajax Control

```
//Waiting for Ajax Control
```

```
WebElement AjaxElement = (new WebDriverWait(driver, 10)).until(ExpectedConditions.presenceOfElementLocated(By. &lt;locatorType&gt;("&lt;locator Value&gt;")));
```

Question 27: How to perform Mouse over action on the submenu item of a header menu?

Answer:

With the actions object you should first move the menu title, and then move to the popup menu item and click it. Don't forget to call actions.perform() at the end. Here's some sample Java code:

Mouse over a submenu item of header menu

Selenium Interview Questions

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

WebElement menuHoverLink = driver.findElement(By.linkText("Menu heading"));

actions.moveToElement(menuHoverLink);

WebElement subLink = driver.findElement(By.cssSelector("#headerMenu.subLink"));

actions.moveToElement(subLink);

actions.click();

actions.perform();
```

Question 28: Can you broadly classify and compare the TDD, BDD and DDD frameworks?

Answer:

You would have heard of all these acronyms buzzing all around. Here I'll briefly explain them and tell how exactly they will help in the system test life cycle.

TDD – Test Driven Development.

It's also called test-driven design, is a method of software development in which unit testing is repeatedly done on the source code. Write your tests watch it fails and then refactor it. The concept is we write these tests to check if the code we wrote works fine. After each test, refactoring is done and then the same or a similar test is performed again. The process is iterated as many times as necessary until each unit is functionally working as expected. TDD was introduced first by XP. I believe I have explained enough in simple terms.

BDD – Behavior Driven Development.

The behavior-driven development combines the general techniques and principles of TDD with ideas from domain-driven design. Its purpose is to help the folks devising the system (i.e. the developer) identify appropriate tests to write—that is, tests that reflect the behavior desired by the stakeholders.

DDD-Domain Driven Development.

DDD is about mapping business domain concepts into software artifacts. A DDD framework offers following benefits:

- Helps the team to create a common model, between the business and IT stakeholders
- The model is modular, extensible and easy to maintain as the design reflects the business model.
- It improves the reusability and testability of the business domain objects.

Selenium Interview Questions

Question 29: What is a Data-driven framework & how is it different from the Keyword Driven framework?

Answer:

Data-driven framework.

In this framework, test case logic resides in test Scripts. Test Data is separated and kept outside the Test Scripts. Test Data is read from the external files (Excel File) and are loaded into the variables inside the Test Script. Variables are used both for Input values and for Verification values.

Keyword Driven.

The keyword/table driven framework requires the development of data tables and keywords. They are independent of the test automation tool used to execute them. Tests can be designed with or without the Application. In a keyword-driven test, the functionality of the Application-under-test is documented in a table as well as in step-by-step instructions for each test.

Question 30: Explain the advantages of using TestNG over JUnit framework?

Answer:

Advantages of TestNG over Junit.

1. In JUnit, we have to declare @BeforeClass and @AfterClass. It is a constraint in JUnit whereas in TestNG there is no constraint like this.
2. Additional Levels of setUp/tearDown level are available in TestNG.
 - a. @ Before/AfterSuite
 - b. @Before/AfterTest and
 - c. @Before/AfterGroup
3. There is no need to extend any class in TestNG.
4. There is no method name constraint in TestNG as in JUnit.
5. In TestNG, we can tell the test that one method is dependent on another method whereas in JUnit this is not possible.
6. Grouping of test cases is available in TestNG whereas the same is not available in JUnit. Execution can be done based on Groups. For example, if you have defined many cases and segregated them by defining 2 groups as Sanity and Regression. And if you only want to execute the "Sanity" cases then just tell TestNG to execute the "Sanity". TestNG will automatically execute the cases belonging to the "Sanity" group.
7. Also, TestNG supports parallel test case execution.

Question 31: What is the purpose of TestNG Parameters which are related to @Test annotation?

Answer:

In TestNG, Parameters are keywords that modify the annotation's function. You can read more about parameters from the TestNG tutorials section on our blog.

Question 32: Can we run a group of test cases using TestNG?

Answer: Yes, TestNG framework supports to execute multiple test cases with the help of test groups.

It provides the following options to run test cases from a specific group.

If you want to execute the test cases based on one of the groups like regression test or smoke test-

Selenium Interview Questions

```
@Test(groups = {"regression-tests", "smoke-tests"})
```

Question 33: Which WebDriver implementation claims to be the fastest and why?

Answer: The fastest and a leaner implementation of WebDriver is the HTMLUnitDriver. The simple reason is because the HTMLUnitDriver does not execute tests in the browser. Instead, it uses a plain HTTP request-response mechanism for running test cases. This approach is far quicker than launching a browser for test execution.

Question 34: Is it possible to use Selenium RC API with Selenium 2.0?

Answer:

Yes, you can emulate Selenium 1.0 API (i.e. RC) with Selenium 2.0. But not all of the Selenium 1.0 methods are supported.

To achieve this you need to get Selenium instance from WebDriver and use Selenium methods.

Method executions might also be slower while simulating Selenium 1.0 within Selenium 2.0.

Question 35: 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.