**Introduction to Selenium WebDriver**  
  
i) Introduction to Selenium WebDriver  
  
ii) WebDriver Environment Setup  
  
iii) Create first Selenium Test Case  
-------------------------------------------  
**i) Introduction to Selenium WebDriver**  
Selenium Tool’s Suite  
        Selenium IDE  
        Selenium RC  
        Selenium WebDriver  
        Selenium Grid  
  
> In 2006 Selenium WebDriver was launched at Google.  
  
> In 2008, the whole Selenium team decided to merge Selenium WebDriver with Selenium RC in order to form more powerful tool called Selenium 2.0  
  
Selenium 1.0 + WebDriver = Selenium 2.0  
  
Selenium 1.0  
  
(Selenium IDE + Selenium RC + Selenium Grid)  
  
Selenium 2.0  
  
(Selenium IDE + Selenium RC + Selenium WebDriver + Selenium Grid)  
  
Note: Now Selenium RC is only for maintenance projects.  
----------------------  
> It is a most important tool in Selenium Suite.  
  
> It has Programming interface only, no IDE.  
  
> Selenium WebDriver supports various programming languages to write programs (Test scripts)  
  
Java  
Python  
C#  
Ruby  
Perl  
PHP  
  
> Selenium WebDriver supports various Browsers to create and execute Test cases.  
  
Mozilla Firefox  
  
Google Chrome  
  
IE  
  
Safari  
  
Opera etc...  
  
> Selenium WebDriver supports various Operating environments.  
  
MS Windows  
  
Linux  
  
Macintosh etc...  
  
> Selenium WebDriver supports Data driven Testing and Cross browser testing.  
  
> Selenium WebDriver is faster than other tools of Selenium suite.  
  
> Selenium WebDriver supports Parallel test execution with the help of either JUnit or TestNG.  
-------------------------------  
Drawbacks of selenium Webdriver  
> Selenium WebDriver doesn't have IDE (some difficult to create test cases)  
  
> No Built-in Result reporting facility.  
  
> No other tool Integration for Test management.  
  
> No centralized maintenance of Elements/objects.  
--------------------------------------------------  
How we create Test Cases /Test Scripts / Tests in Selenium WebDriver:  
In UFT:  
  
Using Objects information and Test Methods we create Tests.  
  
In Selenium WebDriver:  
  
Using Element Locators and Webdriver Commands/methods we create Test Cases.  
  
Selenium IDE:  
  
Using Element locators and Selenese / Selenium IDE commands we create Test cases  
-----------------------------------  
Element Locators - to recognize elements/identify elements.  
  
WebDriver commands/methods - to perform operations on elements.  
--------------------------------------  
Java Programming - for enhancing Test Cases  
  
TestNG Framework - for grouping Test cases, batch Testing and generating Test Reports.  
-----------------------------------------------------------  
**ii) Selenium WebDriver Environment Setup.**  
Steps:  
  
1) Download and Install Java (JDK) software -to create programs (Test scripts)  
  
2) Set Environment variable (path variable).  
  
3) Download Eclipse IDE and extract - to write and execute Java programs.  
  
4) Download Webdriver Java language binding (www.seleniumhq.org) and add WebDriver jar files to Java project in Eclipse IDE.  
---------------------------------------  
5) Install Firebug and Firepath plug ins (Mozilla Firefox) for inspecting elements.  
  
6) For Internet Explorer and Google Chrome, no need to install any plug in, they have  
built in developer tools(F12) for inspecting elements.  
  
7) Firefox driver is default driver in Seelenium Webdriver, for IE and Chrome etc... Browsers then we need to download browser drivers.  
--------------------------------------------  
Download Selenium WebDriver Java language binding from www.seleniumhq.org website and extract.  
  
Add WebDriver jar files to Java Project in Eclipse IDE  
Navigation:  
  
Create Java Project  
  
> Select Java project and right click  
  
> Build path  
  
> Configure build path  
  
> Select "Libraries" tab  
  
> Click "Add external Jars"  
  
> Browse path of the WebDriver jars.  
  
> Add  
-----------------------------------------  
Create Selenium WebDriver Test Case  
> Import Webdriver and Firefox/IE/Chrome Libraries. (In Selenium Test Case/Program)  
  
> Using Element locators and Webdriver commands write test steps.  
  
> Insert java programming statements to enhance Test cases.  
-----------  
> Using TestNG Annotations group test cases, execute test batches and generate detailed test reports.  
--------------------------------------------------  
**iii) Write first Selenium Test Case**  
Manual Test Case  
Test Case ID: gcrshop\_admin\_TC001  
  
Test Case Name: Verify Admin Login in GCR shop Web Portal  
  
Test Steps:  
  
1) Launch the Browser and navigate to "www.gcrit.com/build3/admin"  
2) Enter User name  
3) Enter Password  
4) Click Login Button  
  
Input data:  
  
Username = admin  
Password =admin@123  
  
Expected URL: "www.gcrit.com/build3/admin/index.php"  
  
Actual: http://www.gcrit.com/build3/admin/index.php  
  
Test Result: Pass  
-------------------------------------------  
\* Verification point: Capture the Browser URL after submission of Login details and compare with expected URL.  
----------------------------------------------------------------  
Selenium WebDriver Test Case:  
public class AdminLogin {  
  
public static void main(String[] args) {  
WebDriver driver = new FirefoxDriver(); //Launches Firefox Browser with blank url  
driver.get("http://www.gcrit.com/build3/admin/login.php");  
driver.findElement(By.name("username")).sendKeys("admin");  
driver.findElement(By.name("password")).sendKeys("admin@123");  
driver.findElement(By.id("tdb1")).click();  
  
String url = driver.getCurrentUrl();  
if (url.equals("http://www.gcrit.com/build3/admin/index.php")){  
System.out.println("Login Successful -Passed");      
}  
else  
{  
System.out.println("Login Unsuccessful -Failed");          
}  
driver.close(); //Closes the Browser  
}  
}

|  |
| --- |
| ***WebDriver driver = new FireFoxDriver*** |

"*WebDriver is an Interface,and we are defining a reference variable(driver) whose type is an interface. Now any object we assign to it must be an instance of a class(FireFoxDriver) that implements the interface*."