Stage 1 - Java Programming Language

Stage 2 - Web Automation using selenium

Stage 3 - Hybrid Framework

Selenium - <https://www.selenium.dev/>

* Automates only Web Application
* Open Source
* Language independent - Java, C#, Python, Ruby, Javascript, php, perl

Selenium - A Suite of tools

1. Selenium IDE
   1. No programming knowledge is required.
   2. Record and playback feature
   3. Supports - chrome/Firefox.
   4. Simple scripting or exploratory testing.
   5. Plugin

<https://www.selenium.dev/downloads/>

1. Selenium RC - Remote Control - Depreciated
   1. Programming knowledge is must
   2. Java, C#, Python, Ruby, Javascript, php, perl
   3. Architecture

Source code (java+selenium RC) à RC Server (Turn on/off) à Browser

1. Selenium WebDriver
   1. Programming knowledge is must
   2. Java, C#, Python, Ruby, Javascript, php, perl
   3. Architecture

Source code (Java+selenium WebDriver) à Browser

1. Selenium Grid
   1. LAN network
   2. Hub and Node

Selenium IDE

1. Added a plugin
2. Test plan - minimum one verification - expected value (hardcoded value) vs Actual value (runtime)
   1. Verify (Soft Assertion) - on failure, it will complete the remaining step and make the test method failure
   2. Assert (Hard Assertion) - on failure, it will abort the execution and make the test method failure
3. Export to java

Firefox - <https://ftp.mozilla.org/pub/firefox/releases/90.0/win64/en-US/>

Uppercamelcase - MyFirstProject

Lowercamelcase - myFirstProject

Java Programming

1. Install - JDK 8 (Java Development Kit) - (install JRE also)
   1. <https://www.oracle.com/in/java/technologies/javase/javase-jdk8-downloads.html>

C:\Program Files\Java

1. Ide - eclipse, netbean, intellij
2. Java is platform independent
3. Java Architecture

Source code (.java) --> Byte code (.class) --> O/P

Compiler -- Source code (.java) --> Byte code (.class)

1. Eclipse Structure

Workspace – lowercase

Project 1 - UpperCamelCase

package 1 – lowercase - ( com. companyname.purpose or org.companyname.purpose )

class - UpperCamelCase

Methods and variable - lowerCamelCase

package 2

class

Methods and variable

1. Datatypes
   1. Primitive datatypes/pre-defined dataypes -
      1. range,size,default is defined

Byte – 8 bit

1 bit - + or -

7 bit – storing the number

* 1. Non primitive datatypes /non-predefined datatypes - collection of pre-defined datatypes
     1. String
     2. Array
        1. Size is fixed
        2. Zero based index

1. Control Statement
   1. If condition
   2. Switch
2. Relational operators

==, !=, >,<,>=,<=

1. Logical operator – And, or, not
2. Iterative statement
   1. For loop
      1. Start point
      2. End point
      3. Increment
   2. For each --> designed for handling array and collection
   3. While
   4. Do while
3. Debug - executing the code step by step
   1. Add Breakpoint
   2. Then debug as java application

Aug 17, 2021

1. Hands-on on concepts learned – conditional statement/iterative statement

Aug 18, 2021

1. Methods – Building block of the program
   1. Reusability
   2. Static Method
   3. Non-static method

Static Method

* How to create the static method?
* How to call the static method?

classname.methodname()

areaOfCircle -->

accessmodifier static returntype methodname(arguments)

1. Access modifier
   1. Private - accessible only within the current class
   2. Default – accessible within the package
   3. Protected - accessible within the package and also in inherited class (child class)
   4. Public - accessible anywhere
2. Local variable can be used only within the method

Apr 19, 2021

1. Math class document

<https://docs.oracle.com/javase/8/docs/api/java/lang/Math.html>

1. Non-static method
   * How to create the non- static method?
   * How to call the non-static method?
     + Create object
     + Use object reference to call the non-static method
2. Object creation
   1. Declaration
   2. Instantiation --> allocate the memory
   3. Initialization
3. Methods
   1. With argument, with return type
   2. Without argument, with return type
   3. With argument, without return type
   4. Without argument, without return type
4. Importing and using the class from different package.

Aug 23, 2021 / Aug 24, 2021

1. Variable
   1. Static variable
   2. Non-static variable
2. Static vs non-static
3. Class and Object – definitions
4. Thread.Sleep
5. Collections
6. Constructor
   1. Constructor name and class name will be same
   2. It is kind of method and it will called during object creation
   3. Pre-requisite for the object – you can create constructor
   4. If constructor explicitly created then you need call that only

With parameter

Without parameter

1. Encapsulation – hiding the data member outside the class
2. Method overloading – n number of method with name – compile time polymorphism/static polymorphism
   1. By change in
      1. Number of parameter
      2. Sequence of parameter
      3. Datatype of parameter
   2. Method to be called is resolved during compile polymorphism
3. Inheritance
   1. Reusing the methods from parent class
4. Abstraction
   1. Interface
      1. Set of rules for the child class
5. Method overriding/ runtime polymorphism
   1. Method to be called is resolved during runtime

Generic type vs Non-Generic type

Non-Generic type – not recommeded

* 1. ArrayList
  2. Hashmap

List vs Set

List – can contain duplicates

Set – cannot contain duplicates

Selenium WebDriver

Aug 25, 2021

1. Create a Java Project
2. Download and configure the selenium jar to the project

<https://www.selenium.dev/downloads/>

Configure the jar

Right click on project --> Build path --> Configure Build Path --> libraries --> add external jar

1. Add the chromedriver.exe inside the project home directory

Exception in thread "main" java.lang.IllegalStateException: The path to the driver executable must be set by the webdriver.chrome.driver

To fix the error:

* 1. Goto <https://chromedriver.chromium.org/>
  2. Goto <https://github.com/mozilla/geckodriver/releases/download/v0.29.1/geckodriver-v0.29.1-win64.zip>
  3. Download the driver respective to browser version and place it in the project home directory.

1. get(, gettitle(), getCurrenturl()

Aug 26, 2021

1. Inspect --> tagname, attribute, text or not

<a href="/index.php/auth/requestPasswordResetCode" id=”main” >Forgot your password?</a>

1. Basic locator
   1. id
   2. name
   3. tagname
   4. Classname
   5. linktext
   6. Partial link text

When the locator is duplicate then findelement will pick the first one.

1. Click(), sendkeys(), select

Aug 30, 2021

Hands-on locating element and practice

Aug 31, 2021

1. Exception in thread "main" org.openqa.selenium.NoSuchElementException:

Findelement default nature --> //0.5s to check for the presense of element

1. Synchronization
   1. Unconditional wait
      1. Thread.sleep(5000) --> wait for 5s
   2. Conditional wait
      1. Implicit wait
         1. Default implicit = 0s
         2. Applicable for all findelement and findelements method
         3. Example: Implicit wait = 30s
            1. If element is not present then error will be thrown after 30s.
            2. If element is present then immediately move for the next step.
            3. Polling time – 500ms/0.5s (how frequently checks for the element)
      2. Explicit wait
         1. Exact condition
         2. Polling time – 500ms

Class – WebDriverWait

Non-static – until (condition)

1. Hands-on on Open emr page
2. Hands-on on magento application
3. Hands-on on Mercury tours application
4. Advance locator
   1. Xpath
   2. CSS

Sep 1, 2021

Completed topic - Browser launch, maximize, implicit wait, navigate url, click, sendkeys, gettext

Xpath:-

<https://www.guru99.com/xpath-selenium.html>

1. Hands-on on Medibuddy application
2. Hands-on on Github application
3. Get text from the web page.

Sep 2, 2021

1. Hands-on on oracle login
2. Dropdown
   1. Dropdown with select tag
      1. Selectbyindex
      2. Selectbyvalue
      3. SelectByvisibletext
   2. Dynamic Dropdown

Sep 7, 2021

1. Salesforce signup (hands-on)
2. Frame, Multiple tabs/windows, alert – switchTo()
3. Frame – html embedded into main html
   1. Index
   2. Name or id as String
   3. WebElement

Make sure to come out of frame after handling the elements inside the frame

Sep 8, 2021

1. Exception in thread "main" org.openqa.selenium.ElementClickInterceptedException:
2. Hands-on on <https://www.servicenow.com/>

Sep 9, 2021

1. Close vs quit

Close – close the current tab/ current session id

Quit – close the browser and also it will kill the process

1. Multiple tabs/windows
   1. GetWindowHandles – to get all the session ids
2. Hands- on

Sep 10, 2021

1. Alert – javascrtipt handle
2. Explicit wait

Sep 13, 2021

1. Setup driver using System.setproperty
2. Actions class -
   1. Mouse and keyboard activities
   2. Drawback
      1. Do not distrub the mouse/keyboard during runtime
      2. No proper exception will be given.
      3. Headless is not supported.
      4. Perform() is mandatory always and .build() is required when more than one action merged
3. When locator and displayed font is different – go for xpath
4. Keyboard activities
   1. Modifier keys –ctrl, shift, alt - keyUp() and KeyDown()
   2. Sendkeys() for rest of the keys

Sep 14, 2021

1. Hands-on – phptravel.com, <https://www.royalcaribbean.com/>
2. Setup of testng

Sep 15, 2021

Hybrid Framework

* TestNG – Unit Testing Framework - <https://testng.org/doc/index.html>
* Data Driven Framework
* Page object model – keywords for openemr application

Framework – pre-written code which used for creating a software, automating software.

Steps to design the framework

**Setup Eclipse with TestNG plugin (only once for eclipse)**

1. Create java project
2. Create Test method using testng @Test annotations
3. Just mouse over on @Test and Add the testng library
4. Imported the testng package.
5. Test method will be triggered in the ASCII order
6. priority, invocationcount
7. Add selenium jar for getting selenium code in built path.

Sep 16, 2021

1. Configure the driver using System.Setproperty
2. Verification – minimum one assertion should be done in @Test
3. Annotation
   1. @BeforeMethod – launch browser
   2. @AfterMethod – close the browser
4. Browser configuration

Sep 17, 2021

1. Page Object Model – Design pattern
   1. Reusability
   2. Code will be easily maintainable in future

* For each pages, we will create a class in the framework – Page Class
* Actions will happen through methods – Page Methods
* Object repositories (locator) will be kept at class level or separately

Sep 20, 2021

1. Creating TestCase in new test class – hands-on – creating test method for MessagesTest

Sep 21, 2021

1. Data Provider
   1. Create @Test with parameters
   2. Create a method that return two dim array.
   3. Provide @DataProvider annotation to that method
   4. Connect the @DataProvider method to the @Test method
2. Data Provider and Excel

Sep 22, 2021

1. Excel - <https://poi.apache.org/\>

Download jar and add it to build path

<https://dlcdn.apache.org/poi/release/bin/poi-bin-5.0.0-20210120.zip>

Excel

.xlsx --> XSSFWorkbook

.xls --> HSSFWorkbook

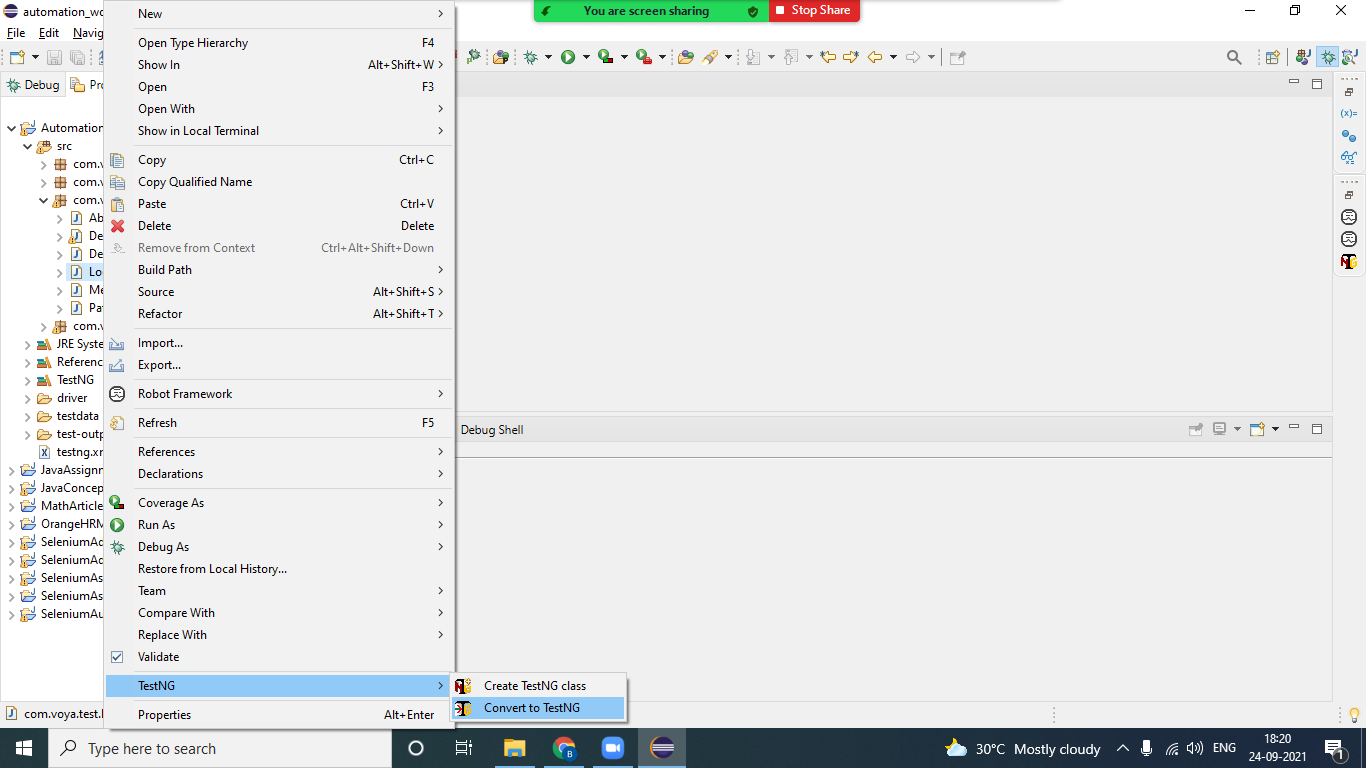
Steps to read:-

* 1. Location – Read/Write
  2. Open
  3. Sheet
  4. Row
  5. Cell

Cell data will be convert to string.

Sep 23, 2021

1. Excel to String[][]
2. Dataprovider and excel method
   1. Create @Test with parameters
   2. Create @Dataprovider method for the @Test methods
   3. Create a sheet with @Test metho name in excel and provide the data as per the parameter sequence.
   4. Connect the @DataProvider method to the @Test method
3. TO remove unused package --> ctrl+shift+i+o
4. Create Testng xml suite



1. XML Parameterization --> sending data from testng.xml

Sep 24, 2021

Build Management tool

Maven Project -

POM.xml --> Project Object Model (will have all configuration detail of the maven project)

Auto configure the jars and even dependent jars will also be added. (network to connect)

1. Create Maven project
2. Group id --> com.voya
3. Artifact id --> project name
4. Version --> 1.0
5. Goto Pom.xml --> add the jar dependency detail from <https://mvnrepository.com/>

Testng - <https://mvnrepository.com/artifact/org.testng/testng/7.4.0>

Selenium - <https://mvnrepository.com/artifact/org.seleniumhq.selenium/selenium-java/3.141.59>

Excel

Poi - <https://mvnrepository.com/artifact/org.apache.poi/poi/5.0.0>

Poi-ooxml - <https://mvnrepository.com/artifact/org.apache.poi/poi-ooxml/5.0.0>

1. Maven goals
2. Maven settings file- C:\Users\JiDi\.m2

Class Assignment -

Framework

1 GB – 1024 MB

1 MB – 1024KB

1KB – 1024 B

1B – 8 bit

Class Assignments:

|  |  |
| --- | --- |
| **PERCENTAGE** | **GRADE** |
| 90 and above | A |
| 80 to 89 | B |
| 60 to 79 | C |
| 45 to 59 | D |
| below 45 | F |

int[] numbers= {45,65,78,98,71,25,33,55,41};

Picked up JAVA\_TOOL\_OPTIONS: -agentlib:jvmhook

Picked up \_JAVA\_OPTIONS: -Xrunjvmhook -Xbootclasspath/a:"C:\Program Files (x86)\Micro Focus\Unified Functional Testing\bin\java\_shared\classes\jasmine.jar"

int[] numbers = {45,88,1002,65,89,650};

String[] colors=new String[3];

colors[0]="red";

colors[1]="yellow";

colors[2]="green";

System.out.println(colors[0]);

System.out.println(colors[1]);