Stage 1 – Java

Stage 2 – Selenium (Web Automation)

Stage 3 – Framework

Git link – <https://github.com/balaji-githubstore/hybrid-framework-java-allianz1_apr_2024.git>

Hybrid Framework – In Master branch

Selenium – In selenium branch

Java – In java branch

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

* Automate only web application
* Open Source
* Language Independent – Java, C#, Python, Ruby, Javascript

Selenium – A suite of tools

* Selenium IDE
  1. No Programming knowledge is required.
  2. Record and playback feature.
  3. Plugin – chrome, firefox, edge
  4. Use it only for simple scripting or exploratory testing
* Selenium RC – Depreciated
  1. Programming knowledge is required
  2. Language – Java, C#, Python, Ruby, Javascript
  3. Architecture

Source code (Java+Selenium RC) 🡪 RC Server (Turn ON/OFF) 🡪 Browser

* Selenium WebDriver
  1. Programming knowledge is required
  2. Language – Java, C#, Python, Ruby, Javascript
  3. Architecture

Source code (Java+Selenium WebDriver) 🡪 Browser

* Selenium Grid
  1. If you want to scale by distributing and running tests on several machines and manage multiple environments from a central point.

Java –

1. Installation
   1. Install JDK

<https://www.oracle.com/ie/java/technologies/downloads/#jdk17-windows>

* 1. IDE
     1. Eclipse
     2. IntelliJ

1. Architecture

Source code (.java) 🡪 Bytecode (.class) 🡪 O/P

Bytecode – Platform independent

Source code (.java) 🡪 Bytecode (.class) = compile time

Bytecode (.class) 🡪 O/P = Run time

1. UpperCamelCase - MyFirstProject

lowerCamelCase – myFirstProject

1. Structure of Java Program

Workspace

Project - UpperCamelCase

Package 🡪 lower case (com.companyname.purpose)

Class - UpperCamelCase

Methods & Variables 🡪 lowerCamelCase

1. Datatypes
   1. Primitive or Pre-defined datatypes
   2. Non-Primitive or non pre-defined datatypes – collection of primitive datatypes
      1. String
      2. Array
2. Methods/function – Building block of the program
   1. Reusability
   2. Maintenance

* Static Method
  + How to call the static method?
    - Classname.methodname()
* Non-Static Method
  + How to call the non-static method?
    - Create an object
    - Objref.methodname()

1. Object
   1. Declaration
   2. Instantiation (new)
   3. Initialization
2. Access modifier
   1. Private – accessible within the class
   2. Default – accessible within the package
   3. Protected - accessible within the package and also in child class
   4. Public – accessible anywhere
3. Variable
   1. Static Variable
      1. How to call the static variable?
         1. Classname.variablename
   2. Non-Static Variable
      1. How to call the non-static variable?
         1. Create an object
         2. Objref.variablename
4. Class – A class is a template or blueprint or type from which objects are created
5. Object
   1. An object is an instance of class
   2. Every Object has its own state (non-static variable) and behaviour (non-static variable)
6. Constructor – pre-requisite
   1. Class name and constructor name must be same. No return type
   2. It will be called whenever objects are created
   3. There will be always a default constructor present in the class and we can override it by creating our own constructor but we need to call the explicitly created constructor always on object creation.
   4. Can create constructor
      1. With arguments
      2. Without arguments
7. this 🡪
   1. helps to distinguish between the instance variable (non-static variable) and local variables.
   2. this keyword will point to the current object.
8. Constructor Overloading/Static polymorphism / Compile time polymorphism
   1. Can create multiple constructor by change in
      1. Number of arguments
      2. Datatypes of arguments
      3. Sequence of arguments
9. Method Overloading/Static polymorphism / Compile time polymorphism
   1. Can create multiple methods with same name by change in
      1. Number of arguments
      2. Datatypes of arguments
      3. Sequence of arguments

Method to be called is resolved during compile time.

1. Encapsulation
2. Collections
   1. Generic type
   2. Non-Generic type
3. Inheritance
   1. Reuse
   2. Instead of recreating the methods and variable from another class we can reuse it using inheritance.
4. Method Overriding/ Run time polymorphism/ dynamic polymorphism
   1. Method to be called is resolved during runtime.
   2. Object created for child need to be stored in parent reference.
5. Abstraction
   1. Abstract class
      1. We can create a method without definition by providing abstract keyword but the keyword should be added to class as well.
      2. We cannot instantiate the abstract class
      3. When we have some method with definition and some without definition then go for abstract class.
   2. Interface
      1. All method declared is by default public and abstract.
      2. Cannot be instantiated

Selenium WebDriver

* + - 1. Create project and configure the jar
      2. Launch browser
      3. Understand the driver configuration
      4. Click, type, Select
      5. Inspect 🡪 tagname, attribute, text or not
      6. Basic locators
         1. Id
         2. Name
         3. Classname
         4. Tagname
         5. Linktext
         6. Partial linktext

When there are duplicate webelement then findElement will pick the first element.

* + - 1. Advance locators
         1. XPath
         2. CSS
      2. To inspect 🡪 f12 or ctrl+shift+c
      3. Synchronization
         1. Unconditional wait (From java)

Thread.sleep(8000) 🡪 wait for 8 sec (Not recommended)

* + - * 1. Conditional wait (from selenium)

Implicit wait

Default implicit wait – 0s

Applicable only for two methods i.e findElement and findElements methods

Example: Implicit wait – 30s

If element is not present, it will check for 30s and then throw exception.

If element is present, it will do the operation immediately.

Polling time – 0.5s (how freq it checks)

Explicit wait

Exact condition

Polling time – 0.5s

Fluent wait

Ignore exception

Modify polling time

* + - 1. Dropdown
         1. With select tag

selectByVisibleText(str)

selectByValue(str)

selectByIndex(int) – starts at 0

* + - * 1. Without select tag

click

* + - 1. ElementInterceptedException – When target element is hidden by some other element
      2. ElementNotInteractable Exception – When element is present but not visible.
      3. Multiple tabs/windows, alert, frame – switchTo()
      4. Multiple tabs/windows
         1. driver.getWindowHandles() 🡪 fetches all session id of the current browser
      5. List vs Set
         1. List – can contains duplicate values
         2. Set – cannot contain duplicate values
      6. Close vs quit
         1. Close – close the current tab/session
         2. Quit – close the current browser/all session and also it will kill the process associated to it.
      7. Alert
         1. Use only for Javascript alert

1. Frame – embedding another html into the current html
   1. Even though the locator is correct, we get NoSuchElementException.
   2. Check for tagname – iframe or frame
   3. Switch to frame (anyone)
      1. Using index (int) – starts at 0
      2. Using name or id (String)
      3. Using xpath (WebElement)
2. Upload
3. Actions – mouse/keyboard
   1. No proper exception will be thrown
   2. May not work in headless mode
   3. Do not disturb the mouse/keyboard

Keyboard

Modifier keys- ctrl, alt, shift – KeyUP/KeyDown method

Rest of the keys – sendKeys()

1. Javascript
   1. Click on hidden element
   2. Type on readonly text box
   3. Scroll to element
   4. Scroll page

Use anyone option

* **How to use javascript to click and type?**

*document.querySelector("#bill-date-long").click()*

*document.querySelector("#bill-date-long").value='20/07/2010'*

JavascriptExecutor js=(JavascriptExecutor) driver;

js.executeScript("document.querySelector('#bill-date-long').value='20/07/2010'");

* How to use javascript and WebElement to click and type?

*WebElement element1= driver.findElement(By.xpath("//input[@id='bill-date-long']"));*

*js.executeScript("arguments[0].value='20/07/2010'",element1);*

1. Selectorhub plugin

Hybrid Framework

* Unit Test Framework – TestNG
* Data Driven Framework – separting the test method and test data in different files (excel, csv, json)
* Page Object Model – Design Pattern for efficiently handling the web elements

Keyword Driven Framework – Create webdriver keywords so it can used in multiple web application automation.

Maven – Build Management Tool

* pom.xml (Project Object Model)
  + It will help us to easily configure the jars.
  + It will automatically download the jar and also dependent jars required.
  + Easily configure the compiler version
  + Easily migrate the project
  + Easily control the project in command line using the maven goals such as clean, compile, test, package and ..

Packages

com.allianz.test – Test class and Test methods

com.allianz.base – Browser and report configuration

com.allianz.utils – Excel, csv file handling code

com.allianz.pages – Page Class and Page Methods

Steps to create a framework

* + - 1. Create maven project
         1. Add groupid and artifactid (projectname)
      2. Add dependency.
         1. TestNG
         2. Selenium
      3. Create a test class and test method
      4. To trigger the @Test method in eclipse, install TestNG for eclipse plugin
      5. TestNG generates report automatically.
         1. Emailable report
         2. Consolidated report (index.html)
      6. @Test method will be triggered based on ASCII keycode order
      7. Assertion – minimum one assertion must be added to the @Test. It decides whether the @test is pass or fail.
      8. Annotation
         1. @BeforeMethod – runs before each @Test
         2. @AfterMethod – runs after each @Test even though @test fails.
      9. DataProvider – To run @Test with multiple set of test data
         1. Create a @Test method with proper arguments.
         2. Create a method that return two-dimensional array and also mark @DataProvider .
         3. Connect @DataProvider with @Test method
      10. Read the excel
          1. Add the dependencies

poi –

poi-ooxml –

Read the excel:

* Location – Read/Write
* Open
* Sheet
* Row
* Cell

.xls – HSSFWorkbook class

.xlsx – XSSFWorkbook class

* + - 1. Connect excel with @DataProvider
      2. Page Object Model
         1. Reusability
         2. Readability
         3. Maintenance

Steps to achieve page object model

For each page in the application, we need to create a class - PageClass

Operations must happen through the methods -Page Methods

Collect the object repository (WebElement/Locators) at class level or different files.

* + - 1. TestNG XML – Suite xml
         1. Control the @Test executions

Run one method

Run all classes

* + - * 1. XML Parameterization for browser config
        2. Group the @Test and run
        3. Parallel execution

At classes

At test tag level

1. Git
2. Extent report