Nitin Gupta

**SOFTWARE TESTING**

1. Process of checking the correctness, completeness, security and quality of developed software application.

2. Software testing is a process to verify and validate the requirement and software application in order to find the defects.

Manual Testing - Facebook application Scenario

Using your hand - eye - brain co-ordination.

**Steps to follows**

1. Open browser

2. Enter the url

3. Entering data in text box

4. Click on button

5. Select single option from the multiple options (radio, checkbox)

**What is mean by automation?**

Automation is nothing but performing all above action with some automation testing tool.

**Java Basic revision**.

1. Basic ( variables, operator)

2. Conditions ( if, else, if else, nested if, switch case)

3. Loops ( for, foreach, while, do while)

4. Write function , call the methods

5. Creating object of class and calling the methods via class

6. Creating and using static method members

7. Collections - 1. List, Set, Map

8. Exceptional Handling - throws and try and catch.

Final keyword

Final method

Final class

Why and when to automate

**Disadvantages of manual Testing**

1. Manual is time consuming and tedious.

2. Requiring a heavy investment in Human Resources.

**Why Automation testing**

1. Improve efficiency of testing

2. Reducing Testing cost

3. Replicating testing across different platforms

4. To give consistent and accurate result.

5. Fast

6. Reliable

7. Repeatable

8. Programable

9. Comprehensive

10. Reusable.

**Things we do before automation.**

1. Identify the test cases that covers functionalities of each module.

2. Identify what can be tested and what can’t be tested.

3. Environment setup should be ready, build should be stable.

**AMAZON ( release -1 )**

1. Registration   
2. Login

3. PLP - product listing page

4. PDP - product description page

5. Checkout page. — current release - 1 - type of stories -

6. Thank you

7. Carthage

8. my account

Credit payment. - story ( development sandbox - environment) —> QA sandbox

Paypal payment - story

UPI - story

Paytm - story

1. When to automate

2. Build is stable

3. Testcase that needs to be run for every build

4. Parameterisation. - testcase that’s requires to run multiple time, login

What is selenium ? - community - open source community , develops open sources tools to test web based applications

**What are the selenium components/tools**

1 . Selenium- IDE - Jason Huggins - selenium recorder., play/ record feature, verification.

2. Selenium RC - remote control. Paul hammant 2008

3. Selenium WebDriver - java, python, ruby, Perl, JavaScript’s, c#, php

4. Selenium grid

Chrome, safari, Mozilla, edge

Database testing

Data driven testing

BDD

**Cross browser and parallel - how can I achieved this**

Selenium grid - **Patrick light body**

**Chrome** **Edge** **safari**

**Windows** **linux**. **MacBook slave**

Code - master

**Selenium WebDriver.**

1. A test automation tool to test web based application (Web sites).
2. WebDriver is an **interface** in java.
3. Its an API

**Pre-Requisite.**

1. Window OS
2. Minimum 10
3. Java
4. Minimum version 11 🡪
5. Step to check java version -> open terminal / cmd – java -version.
6. Editor  
   a. Eclipse/ intellij
7. Any updated browser – chrome, safari, ff, edge, opera

**Configuration Part**

1. Create 2 folders ( mostly except C:/)
2. YourName\_SeleniumSessions
3. WebDriver jar files
4. Open <https://selenium.dev>
5. Click on Downloads
6. Click on Latest stable version [4.27.0](https://github.com/SeleniumHQ/selenium/releases/download/selenium-4.27.0/selenium-server-4.27.0.jar) ( Version may vary) ( Don’t’ open this file)
7. Store these files in WebDriver Jar files folder
8. Open Eclipse  
   a. Select the workspace ( The folder where you are going to store the projects)  
    Select the folder created in 1st step
9. Click a lunch button
10. File -> New -> Java project
11. Give Project Name
12. Select Java Version as Java-SE-11
13. Uncheck create module-info.java file checkbox
14. Create a new package inside this project
15. Create a class inside this package
16. Right click on project ( in Package Explorer)
17. Click on BuildPath
18. Click on Configure BuildPath
19. Click on Libraries Tab
20. Click on Class Path
21. Click on Add Externals Jars…
22. Browser and select the File downloaded ( Selenium-Server-xxxx.jar)
23. Click on Open
24. Click on Apply & Close.

**Selenium WebDriver**

1. Launching the browser : Create the object of WebDriver interface.
2. get(): Launch the specified URL
3. driver.manage().window().maximize(); Maximized the browser window
4. close() closed the opened browser.
5. getTitle(): Read the title of page (String)
6. findElement(): Reads the single WebElement/Control or object on page

using any one locator.Always locate first occurance ( webElement).

1. findElements(): Used to read multiple elements on the page (List<WebElement>)

**Web Element**

1. Interface in selenium WebDriver
2. This refers any control or element or object on the page
3. Methods
   1. sendKeys()
   2. Click()
   3. getText() : Used to read the text on the element (String);
   4. isSelected():Check Whether the control (Checkbox/Radio button) is selected or not ( Boolean – true/false)
   5. isEnabled() : check Whether the element is enabled or disabled on any element ( Boolean – true/false)
   6. isDisplayed() : Check Whether the element is visible or not ( Boolean – true/false)
   7. getDomAttribtue() : returns the value of any attributes (String).

**Locators**

These are the ways to find/ locate any Web Element on the page

1. Id
2. Name
3. className
4. CssSelector
5. Xpath
6. LinkText
7. PartialLinktext
8. TagName
9. RelativeLocator

Attribute

<**input** type="text" class="inputtext \_55r1 inputtext \_1kbt inputtext \_1kbt" name="email" id="email" tabindex="0" placeholder="Email address or phone number" value="nitin@gmail.com" autocomplete="username" aria-label="Email address or phone number">

1. What is a Tagname. = Input  
 2. What is an Attribute = class  
 3. What is an value = nitin@gmail.com

tagname

Value

**Common Exception in Selenium WebDriver**

1. InvalidArgumentException: Your url is wrong (URL should always start by https means it should be Absolute URL)
2. SessionNotCreatedException: Your Browser and WebDriver version are mismatch
3. NoSuchElementException: Selenium is unable to locate this control bevause od some reason  
    a. Value of locator is wrong  
    b. Value of locator may be dynamic (Every time it is changing)  
    c. Synchronization issue.
4. Invalid SelectorException: The value of locator is in wrong format.
5. ElementNotInteractableException: You are not able to interact with this control.

CSS Selector

1. Single attribute – tagname[attribute=’value’]
2. Using multiple attribute –

tagname[attribute1=’value1’] [attribute2=’value2]

1. Using special characters
2. ^ - Starts With
3. $ - Ends With
4. \* contains
5. . – stands for class (attributes) -

1 .classattributevalue - .inputtext, .

* + 1. .inputtext.\_55r1.\_6luy
    2. Tagname.classname. - input.inputtext

( if class value holds multiple classname then replace spaces among the class value by

“.”)

1. # - Id attributes. - #idvalue -- #passContainer

Xpath – XMLPath. – Address of locators on Webpage.

Pune -India

World -> Regions -> Country 🡪 States 🡪 City 🡪 Area - > local Area 🡪 house number – address.

World 🡪 Asia 🡪 india – Maharashtra – Pune – Wagholi – XXXXX

1. Absolute Xpath – starts with html – html/<inner-tag>/inner-tag
2. Relative Xpath –
3. //tagname[@attribute=’value’]/tagname
4. //tagname[@attribute=’value’]

Handling Dropdown List

If the control/Element is having <select> tag then only it is treated as dropdown./dropdown list.

Select is the class used to handle dropdown list / list box

Method of the Select Class

1. getFirstSelectedOption(): return the selected option from the list ( web Element)
2. getOptions(): Return list of the option from the dropdown.
3. selectByVisibletext(): Selects the option by using the text on that option.
4. selectByValue(): Selects the option by using its value attributes
5. selectByIndex(): Selects the option by using its index ( Zero Based).
6. deSelectByIndex : deselect the option by using its value attributes
7. deSelectByValue: deselect the option by using its value attributes
8. deSelectByVisibleText(): deSelects the option by using its value attributes
9. Show name of the country that is selected.
10. Show total number of countries
11. Show list of all the countries
12. Select Canada from the list
13. Display selected country.

Difference between dropdown and list box

1. Size of control ( optional)
2. List box allows to select multiple options
3. In List box multiple options are visible whereas in dropdown list only one option is visible

Synchronization ( waits in Selenium)

It is the process of adjusting speed of tool with speed of the application.

1. Thread.sleep(); - is used to pause the execution of scripts for specified time duration.(Miliseconds)

a. Application to single statement only  
 b. It takes mandatory delays

2. ImplicitWait - checks whether elements is loaded or not in given time period.

a. It don’t take mandatory delay

b. it is applicable to the entire scripts.

3. ExplicitWait ( WebDriver Wait) – make selenium WebDriver object wait till the certain condition to be made. ( isalertpresent, iselementlocated, iselementclickable, iselementdisplayed)

a.Applicable to single statement only

b. it don’t take mandatory delay.

4. FluentWait - Next version of explicit wait  
 a. Applicable for single statement only

b. it don’t take mandatory delay

c. we can handle exception as well

5. PageLoadFormat.

Implicit wait – 10

webDriverWait – 10 - 2 – 8second

fluent wait - 20 max of time

polling every - 2

**Handling Tables**

1. Display all the columns/headers of the tables
2. Display total number of rows in the table.
3. Read the data from row in the table.

Table > thead > header resides  
 Table > thead > table row > table column  
 Table > tbody > table row > table column  
 th – headers. , td – column , tr - row

**Handling Alerts** ( JavaScripts Alerts)

If the pop-up is not inspect able then only it is an alert

Alert interface is used to handle Alert in selenium.

1. Accepts() ; this method would use to accepts the alert (OK)

What is testNG  
1. TestNG is an open source unit testing framework  
2. Its an test framework also  
3. testNG public in the market under apache user license.  
its an framework which incluse sets of rules/methods classes for test execution that would help automation to control the test executions.

Advantages of testNG  
1. Set priorities for test cases  
2. Combine multiple tests together  
3. Uses annotations  
 1. @test.  
 2. @BeforeTest  
 3. @AfterTest  
 4. @BeforeMethod  
 5. @AfterMethod  
 6. @BeforeSuite  
 7. @AfterSuite  
 8. @BeforeClass  
 9. @AfterClass  
 10. @DataProvider   
4. Implement data driven testing  
5. Get Reports – Normal report | HTML report  
6. Parameterization.  
7. Execute/Skip/multiple test.

@Anotations

1. @Test: This method is to create or treated as testcases.
2. @BeforeTest : This method will get executed once before executing 1st test.
3. @AfterTest: This method will get executed once after executing last test.
4. @BeforeMethod: This method will get executed before each test runs
5. @AfterMethod : This method will get executed after each test runs.
6. @BeforeSuite : this method run once prior all method ( annotations) present in the class
7. @AfterSuite : this method run once after all methods ( annotations) run in class
8. @DataProvider : this annotation is used to perform parameterization.

Registration -> login -> PLP 🡪 PDP – add product -> checkout-> thankyou

Testcases

Tstcases

testcases

**Assertions - what is an defect ?   
 1. Difference between expected and actual result that call defect.  
 2. Assertions – hard assert and soft assert** Assert True and Assert Equals  
 Assert True - > when user needs to validation Boolean conditions.  
 Examples : to check web element on webpages  
   
 Assert Equals = When user needs to validation two values (String)  
 Examples : to check title of pages or some text on the web pages.

**Modular Framework.**

1. Executing or skipping single or multiple testcase
2. Executing testcases via xml files
3. Executing testcases in a groups
4. Executing testcases in parallel
5. Executing testsuites ( more then one testcases)

**Points to be remember while working with XMLS**

1. All the tags are pre-defined.
2. All tags are case sensitive
3. You cannot alter sequence of any tag.

**Hard assert Execution steps**

1. **Launch browser**
2. **Validate the logo on homepage | soft assert soft.assertAll();**
3. **Validate title of the homepage | hard assert**
4. **Do login**

Way of writing the code and executes it

1. Automate one testcase
2. Create on class
3. And write all steps under main method.
4. Keyword driven– how testcases runs with help of keywords
5. Create properties file –
6. Environments 🡪 Development 🡪 QA 🡪 Staging 🡪 Production 🡪 UAT
7. Function – login 🡪 URL , credentials, locators, testdata
8. PageObjectModel
9. LoginPage. – webdriver Object
10. registerationPage - DriverObject
11. PLP - Object
12. PDP – Object
13. How to read and write data in/from excel file.

Reading and Writing Data in/from Excel using POI  
POI Apache - open source libraries to help selenium to automates (read and write the data from/in to files) excels files

1. Properties

2. Excel files

data which require to change frequently - excels - testdata

data which doesnot changes frequently - property file - environment data.

URL - admin credentails , tile of the pages

1. WorkBook ( Excel files)

collection of sheets

2. Sheets - Collection of Rows & Column

3. Row - Collection of Cells

4. Cell - it contains a data.

5. Handling the Excel Files

1. WorkBook

2. Sheet

3. Row

4. Cell

nitin.xls , nitin.xlsx

office 2003 - .xls - HSSFWorkBook

office 2007 - .xlsx - XSSFWorkBook

1. Create an object of file class

2. Create an object FileInputStream/FileOutputStream

3. Create an object XSSFWorkBook/HSSFWorkBook

4. Create an object of Sheet

5. Create an object of Row

6. Create an object of Cell

**Maven** – Maven is a build tool.

Maven have central repository where companies stores their libraries or jar – selenium, testng etc. its has pom.xml which consider a heart of Maven project.

Creating a maven project - user to supply groupID and artifactID

groupId – package name | artifactid – project name

Main – location where tester or developer can write and keep there methods.  
Test – location where tester can keep and write there testcases.

main/resources – any files or jar, plugins related to methods those are keep in main structure   
test/resources - any files, testate, driver related to testcases those are keep in test structure

Excel file – testdata, properties files

Dynamic locators handling with Xpath.

1. Xxxxxxxxxx12323123123 – starts-with. (// tagname[starts-with(@attribute,’value’)])[1]
2. 123231231231xxxxxxxxxx – contains // tagname[contains(@attribute,’value’)]
3. Xxxxxxxx1231231231xxxxxxx – starts-with or contains

Parent and child

Parent -- would return the immediate tag above current node

//div[@class='\_1ZMrY\_']/parent::div. |. Currentnode/method::tagname  
Child – Would return the immediate tag after the current node

//div[@class='\_1ZMrY\_']/child::div

POI – maven dependencies

<dependency>

<groupId>org.apache.poi</groupId>

<artifactId>poi</artifactId>

<version>5.3.0</version>

</dependency>

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

<dependency>

<groupId>org.apache.poi</groupId>

<artifactId>poi-ooxml</artifactId>

<version>5.3.0</version>

</dependency>

<!-- https://mvnrepository.com/artifact/com.aventstack/extentreports -->

<dependency>

<groupId>com.aventstack</groupId>

<artifactId>extentreports</artifactId>

<version>5.1.2</version>

</dependency>

Cucumber

TDD – approach | test driven development

BDD – Behavior driven development.

**Feature File**

1. Created by QA Team,
2. Gherkin Syntax / Language ( simple plain English language) –
3. Some Keywords - Given, when, then, and, examples, scenarios , scenarios outline, background.
4. Collection of testcases –
5. Team , could have more than 1 feature file.
6. This file of scenarios share with developer, so that developer can generate the code/functionality.
7. Same File is shared with manual tester, so that tester will create testcase
8. Same file is shared with automation test so that automation tester will create the automation scripts.

Create a maven project for cucumber

1. Create maven project – right click -> project 🡪 maven project
2. 🡪 select 3rd checkbox. – click on next button
3. Enter io.cucumber at filter field in eclipse project window.
4. Select the said filter from below list. > next button
5. Supply the package name – group id and artifact Id- project name
6. Click on Finish button
7. Then type y and hit enter button on your keyboard.

Steps definition/ glue code

1. Class that contains the logic/code for automation testcases.
2. Contains junit annotations.

Keyword’s in feature file. Login.feature

1. Feature : the Requirement ( this feature file contains the scenario or testcases

for login page)

1. Scenario: test case objective
2. Scenario outline : when one scenario to be run for multiple times in cucumber using feature files.
3. Given : The pre-re-quisite
4. When : The steps to be performed
5. And: used to combined multiple steps ( when statement)
6. Then: The expected result

Create feature file in eclipse

1. Right click on test/java > new> file
2. Supply the xxxxxx.feature
3. Delete the existing lines in the newly feature file.

Feature file

Test cases – Scenarios 🡪

Feature file – it’s a testcase file

User can multiple testcases.

Feature: Login

Given open google

When Read the title

Then Title should be google

Given open google

When Read the title

Then Title should be google

Feature > Stepdefinition > Runner

Feature > StepDefinition > Page > Runner

LoginPage.feature > LoginSteps > LoginPage > Runner

Feature > stepdefinition > module > page > runner

LoginPage.feature > loginSteps > LoginModule > LoginPage> Runner