**MYSQL – How to create a database**

1. Open command prompt

2.

C:\Users\Shiny Ingrid C>mysql -u root -p

Enter password: \*\*\*\*

3. mysql> create database database\_name;

response: Query OK, 1 row affected (0.01 sec)

4. mysql> show databases;

response : existing databases show up

check if the newly created database is present

5. mysql> use database\_name;

response: Database changed

6. mysql> create table table\_name(id INT NOT NULL AUTO\_INCREMENT,column\_name1 VARCHAR(100)NOT NULL,column\_name2 VARCHAR(100) NOT NULL,column\_name3 VARCHAR(100)NOT NULL,PRIMARY KEY(id));

Query OK, 0 rows affected (0.05 sec)

Eg: mysql> create table qspiders(id INT NOT NULL AUTO\_INCREMENT,first\_name VARCHAR(100)NOT NULL,last\_name VARCHAR(100) NOT NULL,address VARCHAR(100)NOT NULL,PRIMARY KEY(id));

Query OK, 0 rows affected (0.05 sec)

7. mysql> insert into table\_name(column\_name1,column\_name2,column\_name3,.....)values('value1','value2','value3');

Eg:

mysql> insert into test1(first\_name,last\_name,address)values('shiny','ingrid','chennai');

Query OK, 1 row affected (0.01 sec)

Query OK, 1 row affected (0.01 sec)

**ChromeOptions arguments**

**ChromeOptions options = new ChromeOptions();**

**options.addArguments("");**

**WebDriver driver = new ChromeDriver(options);**

**--incognito =>For incognito mode**

**--disable-background-mode => for disable background apps**

**--disable-plugins =>to disable the plugins of the browser**

**--restore-last-session => to restore the immediate previous session**

**--mute-audio =>to mute audio across the browser**

**--disable-notifications =>to handle the notification popup**

**--use-fake-ui-for-media-stream =>to handle allow block popup**

**Handle SSL Certs Issue**

**options=new ChromeOptions();**

**options.setAcceptInsecureCerts(true);**

**WebDriver driver = new ChromeDriver(options);**

**Syntax for setting up browser executables**

**System.setProperty("webdriver.gecko.driver", “exe file path”**

**Ex: "C:/Users/siva/workspace/Selenium/driver/geckodriver.exe");**

**Get parameters in runtime**

import java.util.Scanner;

Scanner sc= new Scanner(System.in)

sc.nextInt();

**JavascriptExecutor**

JavascriptExecutor js = (JavascriptExecutor) driver; **TYPE CASTING**

js.executeScript(Script,Arguments);

**Launch a url** : js.executeScript("window.location = 'https://demo.guru99.com/'");

js.executeScript(“window.location=arguments[0];”,url);

**Refresh a page:** js.executeScript(“location.reload()”);

**Click an element**: js.executeScript(“document.getElementByID(‘element id ’).click()”); (no ; is required after click) js.executeScript("arguments[0].click();", ele);

**Pass value** : js.executeScript(“document.getElementById(‘element id ’).value = ‘xyz’;”)

js.executeScript(“document.getElementById(‘elementID’).setAttribute(‘value’,’hello’)”);

**Interact with checkbox:** js.executeScript(“document.getElementByID(‘element id ’).checked=false;”);

**Get title** : String s=js.executeScript("return document.title;").toString();

**Get url:** String s=js.executeScript("return document.URL;").toString();

**Vertical scroll down by 600 pixels**: js.executeScript("window.scrollBy(0,600)");

**Scrolldown in a page:** js.executeScript("0,document.body.scrollHeight");

**Scrollup in a page:** js.executeScript("0,-document.body.scrollHeight");

**To create an alert popup**: js.executeScript(“alert(‘type the message’);”);

**Call executeAsyncScript() method to wait for 5 seconds:**

js.executeAsyncScript("window.setTimeout(arguments[arguments.length - 1], 5000);");

**To scroll to an element:** js.executeScript("arguments[0].scrollIntoView(true);",ele);

**To Scroll by co-ordinates**: js.executeScript("window.scrollBy("x ","y ")");

[Point point = ele.getLocation(); int x=point.get(X); int y = point.get(Y);]

**WebDriver Interface Methods**

1. get()
2. getCurrentUrl()
3. getPageSource()
4. getTitle()
5. close()
6. quit()
7. manage() |.window().maximize
8. navigate()
9. getWindowHandles()
10. getWindowHandle()
11. switchTo()
12. findElement()
13. findElements()

**WebElement Interface Methods**

1.sendKeys() 10.getLocation()

2.submit() 11.isDisplayed()

3.click() 12.isEnabled()

4.clear() 13.isSelected()

5.getAttribute() 14.findElement()

6.getCssValue() 15. findElements()

7.getTagName()

8. getSize()

9.getText()

**SENDKEYS ========🡺FOR REFERENCE**

sendKeys("sdf",*Keys*.***CONTROL***,"ac");

sendKeys("sdf",*Keys*.***CONTROL***,"v");

sendKeys(Keys.CONTROL,"a");

sendKeys(Keys.CONTROL,"c");

sendKeys(Keys.CONTROL,"v");

**How to handle broken link?**

**Get the link:**

List<WebElements> allLinks=driver.findElements(By.xpath(“//a”));

for(WebElement eachLink : allLinks)

{

String link=eachLink.getAttribute(“href”);

URL url = new URI(link).toURL(); // =new URL() is deprecated

HttpURLConnection huc=(HttpURLConnection) url.openConnection();

int code=huc.getResponseCode();

if(code>400)

{

System.out.println(link+“ is broken”);

}

}

**How to handle broken images?**

**Find the image by xpath:**

WebElement img=driver.findElement(By.xpath(“//img[…]”));

JavascriptExecutor jse = (JavascriptExecutor) driver;

boolean isBroken=(boolean) jse.executeScript(“return arguments[0].naturalWidth===0 && arguments[0].naturalHeight===0”,img);

if(isBroken==true)

{

System.out.println(“Image is broken”);

}

else

{

System.out.println(“Image is not broken”);

}

**How to delete cookies?**

driver.manage.deleteAllCookies();

**How to navigate to previous and back page?**

driver.navigate().back();

driver.navigate().forward();

**How to switch to new window and tab?**

driver.get("https://www.mayoclinic.org");

driver.switchTo().newWindow(*WindowType*.***TAB***).get("http://www.google.in");

driver.switchTo().newWindow(*WindowType*.***WINDOW***).get("https://www.facebook.com/");

**How to find the color of a WebElement?**

String s= **ele**.**getCssValue(“color”);** //returns color in the form of rgba(255, 255, 255, 1)

String hexV=**Color.fromString(s).asHex();** //returns in the form of HexCode #ffffff

**How to get the values of attributes? 🡺**

ele.**getAttribute(“type”);**

**How to switch to window?**

String parent= driver.getWindowHandle();

Set<String> allWin=driver.getWindowHandles();

for(String win: allWind)

{ String title=driver.getTitle();

if(title.contains(“”))

{ driver.switchTo.window(win);

break;

}

}

**ALERTS**

**How to store alert in a variable?**

Alert alert = driver.switchTo().alert();

alert.accept | alert.dismiss()

**How to wait for an alert to be displayed and store alert in a variable?**

Alert al = wait.until(ExpectedConditions.alertIsPresent());

**How to store alert text in a variable?**

String text = alert.getText();

If the default behaviour was set to "ACCEPT", the alert will be closed automatically, and the switch To().alert() wont be able to find it. The solution is to modify the default behaviour of the driver ("IGNORE"), so that it doesn't close the alert:

**DesiredCapabilities dc = new DesiredCapabilities();**

**dc.setCapability(CapabilityType.UNEXPECTED\_ALERT\_BEHAVIOUR, UnexpectedAlertBehaviour.IGNORE);**

**d = new FirefoxDriver(dc);**

LOCATORS

======================

We have 8 types of locators

1.ID

2.Name

3.ClassName

4.LinkText

5.PartialLinkText

6.TagName

7.CssSelector

8.xpath

Priority order has to be followed to selecting locators

--------------------------------------------------------

**Id-->name-->className-->CssSelector-->xpath-->TagName**

While locating Links - priority

------------------------------

**Id-->name-->classname-->cssSelector-->LinkText-->PartialLinkText-->xpath-->tagName**

**Xpath**

**XPath Syntax:**

/ => Immediate Child | // =>any child | /.. => parent | /../.. =>root element

/child:: | /ancestor:: | /following-sibling:: | /preceding-sibling:: | :: Scope Resolution Operator | //\* =>all the tags of html body

**Case 1: Use Attributes**

//tagname[@AN=’AV’]

//tagname[contains(@AN,’AV’)]

**Case 2: Use Visible Text**

//tagname[text()=’textvalue’]

//tagname[contains(text(),’textvalue’)]

**.**  can be used instead of text() Ex: //tagname[**.**,’textvalue’]

**Case 3: Use both Attributes and Visible Text**

**//tagname[text()=’tv’ and @AN=’AV’] | //tagname[text()=’tv’ and @AN=’AV’ and text()=’textvalue’] | //tagname[text()=’tv’ or @AN=’AV’]**

1. //span[text()='Sign in' **and** @class='action-inner']
2. //span[@type='submit' **or** @name='btnReset']

**Case 4: Use Axes (dependent/independent) : DYNAMIC**

Step 1: Inspect the static or nearest independent element **static 🡺 //tagname[@AN=’AV’]**

Step 2: Find the common parent by traversing => static/../.. until you reach the common ancestor with **id static/ancestor::tn[@AN=’AV’]**

Step 3: Locate the dynamic element by traversing down /\*/\* until you reach the element **static/ancestor::tn[@AN=’AV’]/descendant::tn[text()=’tv’]**

**COMPRESSING XPATH - //Ancestortgn[@AN=’AV’ and contains(.,’tvOfStaticElement’)]**

**How to locate svg elements?**

//\*[name()='svg']

//\*[name()='svg' and @AN='AV']

//tagname[@AN=’AV’]/\*[name()=’svg’]

**Case 5: Web Table :only applicable if we have <table> tag**

**Base===🡺 //tagname[@AN=’AV’] or //tagname[text()=’textvalue’]**

|  |  |
| --- | --- |
| Base/table/tbody/tr[\*] | Captures all the rows in the web table |
| Base/table/tbody/tr[\*]/td[\*] | Captures all the columns of all the rows of the web table |
| Base/table/tbody/tr[last()] | point the last row of the web table |
| Base/table/tbody/tr[\*]/td[last()] | points to the last column of the all rows |
| Base/table/tbody/tr[last()-1]/td[last()-2] | Points to row before last and second to last colum |
| Base/table/tbody/tr[\*]/**.** | **.** refers to all |
| Base/table/tbody/tr[2]/td[3] | **Indexing is allowed in table** |
| Base/table/tbody/tr[contains(.,’wini’)]/td[3] | To find value of a record. Ex: salary of wini -> sal in col 3 |

|  |  |
| --- | --- |
| **SELENIUM 4** | **SELENIUM 3** |
| ChromeOptions opt = new ChromeOptions(); | DesiredCapabilities cap = new DesiredCapabilities(); |
| RelativeLocators |  |
| Selenium Grid -support for docker |  |
| Sel IDE – improved GUI | Supports Chrome as well |  |
| Architecture: W3C protocol | JsonWireProtocol |
| Actions class : click,contextClick and Double Click on WebElements |  |

**POM CLASS**

**ANNOTATIONS**

**============**

**@FindBy** - identifies element using one locator or one condition

**@FindBys ---> AND -** multiple FindBy annoation - so identifying single element with multiple locator**s**

**Considered as true if all conditions are met**

**@FindAll----> OR ->** Identifying single element using multiple locators - any one condition needs to be met

**HOW TO FREEZE A WEBPAGE FOR INSPECTION**

**Approach : 1**

**==============**

Inspect - console - in the last empty line type

**setTimeout(() => { debugger }, 3000)**

keep the element to inspect visible and hit enter. Immediately click on whatever needs to be displayed. Page is held and we can inspect

**Approach 2:**

**===========**

Open DOM - Sources - Right End - Event Listener Breakpoints - control - Check **Blur** -do the action and click F8 - screen will pause

**Approach 3:**

**===========**

Open the webpage - inspect - access source tab - select the element - click function+f8(twice for dell)

**Handling command prompt:**

**cd..** move to previous location in current location : C:\\sddsdf\sdsds cd..|Enter -> c:

**cd locationOfProject**  moves to the location

**cls** clear screen

**Maven Command line Commands:**

mvn clean

mvn validate

mvn compile

mvn test

mvn -Dtest=TestSciptName test

mvn -Dtest=TestScriptName#methodname -Dkey1=value -Dkey2=value test (-Durl=http://loclhost:9090 No””)

mvn test -P profileID

mvn test -DsuiteXmlFile=testng.xml

**How to pass parameters using Jenkins ?**

**=======================================**

**Login to Jenkins - Click on Job - Configure - Build - Under Goals and Options pass the command just like maven command in command prompt**

**Ex : test -P Regression -Durl=http://locahost:8080 -Dbrowser=chrome -Dusername:admin -Dpassword:admin**

**Here, instead of http://locahost:8080 in -Durl, they enter the testing env url -Durl=testingEvn.com**

**Save**

**Customers may not be able to add all the parameters technically. So add the parameters**

**In configuration window,**

**-General**

**-This Project is Parameterized**

**-Add Parameter**

**-Choice Parameter - create choice parameter for each command**

**Ex: Name - Suite**

**Choice**

**ST**

**RT (name of the suite IDs in Profiling)**

**Ex2 : Name : Environment**

**Choice**

**http://testenv.com**

**http://sitenev.com**

**http://uatenf.com**

**Ex3: Name : browser**

**Choice**

**chrome**

**edge**

**firefox**

**Once you save the parameters, the Build Now Option changes to Build With Parameters - Click on it - Select the parameters - Click on Build**

**NOTE: IF YOU USE BUILD WITH PARAMETERS, THEN CHANGE THE GOALS AND OPTIONS COMMAND**

**FROM**

**test -P Regression -Durl=http://locahost:8080 -Dbrowser=chrome -Dusername:admin -Dpassword:admin**

**TO**

**test -P $Suite -Durl=$url -Dbrowser=$browser -Dusername:$username -Dpassword:$password**

**(replace the value with $parameterName/title set in the Parameters)**

**TESTNG ANNOTATIONS**

**@Test(dependsOnMethods=”methodName”) | (dependsOnMethods={“mtdName1”,”mtdName2”)**

**@Test(priority=\*)**

**TESTNG**

|  |  |  |
| --- | --- | --- |
| ITestListener | @Listeners(packN.ListImpCN.class)   public class className() { } | public class ListImp implements ITestListener , ISuiteListener   { @Override  public void onTestFailure(ITestResult result) {}  } |
|  |
|  |
| RetryAnalyzer | @Test(retryAnalyzer="packN.RetryAnImpClassN.class) | public class RetryListenerImplementation implements RetryAnalyzer{ @Override public Boolean retry(ITestResult result) {}  } |  |
|  |
| CrossBrowser Execution | @Parameters("ParaNameFromSuiteXML")  @BeforeClass   Public void bc(String v\*)   { String x = v; } | v\* - String variableToStoreThePara |  |
| xml file:  <test>  <parameter name="" value=""/>  <classes> |  |
| Group Execution | @Test(groups="gn") | **xml:**  <suite>  <groups>  <run>  <include name ="gn">   <test>  <Classes> |  |
| @Test(groups ={"gn1","gn2"}) |  |
| Parallel Execution | xml File:  <test name ="test1">  <classes>  <class name=PN.CN/>  <classes/>  </test>  <test name ="test2">  <classes>  <class name=PN.CN/>  <classes/>  </test> | xml File Creation - parallel :tests | thread-count:\* |  |
| Regional Regression Execution | <test>  <classes>  <class name=""> <methods>  <include name="">  </class>  </classes>  </test> | Within class / methods |  |
| Batch Execution | Xml file :  <suite name=”suiteName”>  <test name=”testName”>  <classes>  <class name=”PN.CN”/>  <class name=”PN.CN”/>  </classes>  </test>  </suite> | Add multiple classes |  |

|  |  |  |
| --- | --- | --- |
| DataProvider | public class dataProviderDemo{  @DataProvider   public Object[][] dpdata()   { Object[][] obj = new Object[x][y] ;} | Within the same class   @Test(dataProvider="dpdata")   public void demo(String d1, Stringd2) {} |

**ASSERTION**

**HardAssert** **stops** the flow of execution : throws exception

**SoftAssert** does **not stop** the flow of execution. Captures the exception but throws only when assertAll() is called

**Syntax :**

Assertion hardAssert = new Assertion(); import org.testng.asserts.Assertion;

Assertion softAssert = new SoftAssert();

(Assertion class does not contain assertAll())

Hence object is created as **SoftAssert sa = new SoftAssert();** import org.testng.asserts.SoftAssert;

sa.assertEquals(a,b);

sa.assertAll();

Hard Assert – Object is not required to be created | For Soft Assert – Object needs to be created

* + - Assert.*assertEquals*(a, c);
    - Assert.*assertNotEquals*(a, b);
    - Assert.*assertTrue*(flag);
    - Assert.*assertFalse*(flag2);
    - Assert.*assertNotNull*(c);
    - Assert.*assertNull*(d);
    - Assert.*assertSame*(a, c);
    - Assert.*assertNotSame*(c, d);

**SELENIUM GRID : DEFAULT PORT 4444**

**HUB : SELENIUM 4**

**java -jar selenium-server-<version>.jar hub**

**java -jar selenium-server-<version>.jar node**

**java -jar selenium-server-<version>.jar node --port 5555**

**java -jar selenium-server-<version>.jar node --hub http://localhost:4444/grid/register**

**java -jar selenium-server-<version>.jar node --port 6666**

**HUB : SELENIUM 3**

C>**java -jar D:\softwaredownload\selenium-server-standalone-3.141.59.jar -role hub -port 5555 (to change port)**

C>**java -jar D:\softwaredownload\selenium-server-standalone-3.141.59.jar -role hub**

12:04:34.544 INFO [Hub.start] **- Selenium Grid hub is up and running**

12:04:33.560 INFO [GridLauncherV3.lambda$buildLaunchers$5] **- Launching Selenium Grid hub on port 4444**

12:04:34.544 INFO [Hub.start] **- Nodes should register to http://192.168.91.1:4444/grid/register/**

12:04:34.544 INFO [Hub.start] **- Clients should connect to** [**http://192.168.91.1:4444/wd/hub**](http://192.168.91.1:4444/wd/hub)

**NODE: SELENIUM 3**

>java -Dwebdriver.chome.driver=D&DchromeExeFile -jar D&DstandaloneServer -role node -port \*\*\*\* -hub pasteTheRegisterLink

**STEPS FOR REMOTE EXECUTION**

1. Open command prompt from location
2. Cmd:**java -jar <standalone.jar> hub** |Enter

Find the url from cmd

Started Selenium Hub 4.23.1 (revision 656257d8e9): <http://172.16.16.180:4444>

**DEFAULT PORT OF HUB :4444 (change –port \*\*\*\*)**

1. XPUB and XSUB ========🡺 tcp://ipaddressOfHUB:port ofXPUB and XSUB : Make a note of it
2. NODE : open cmd prompt FOLDER SHOULD CONTAIN DRIVER EXE FILES
3. **java -jar <standalone.jar> node --detect-drivers true --publish-events tcp://172.16.16.180:4442 --subscribe-events tcp://172.16.16.180:4443 --selenium-manager true --port 2222**

4442 – from hub -🡪 XPUB binding to [binding to tcp://\*:4442,

4443 – from hub -🡪 XSUB binding to [binding to tcp://\*:4443,

//172.16.16.180 from HUB 🡺Started Selenium Hub 4.23.1 (revision 656257d8e9): http://172.20.10.3:4444

Eclipse -> Test Script

1.Create Multiple test scripts for parallel execution

2. Create a suite file using those classes (Parallel -tests | Thread-count -5 | Create Test Runners -multiple)

3. In test script

DesiredCapabilities cap = new DesiredCapabilities();

cap.setBrowserName(“chrome”); **// cap.setBrowserName(“edge”);**

cap.setPlatform(Platform.WINDOWS);

URL url = new URI(“<http://172.16.16.180:4444>”).toURL(); **//link from selenium grid hub**

WebDriver driver = new RemoteWebDriver(url,cap);

driver.get(<http://www.google.com>);

<suite parallel="tests" name="Suite">

<test thread-count="5" parallel="tests" name="edge">

<classes>

<class name="SeleniumGRID.GRID\_edge"/>

</classes>

</test> <!-- Test -->

<test thread-count="5" parallel="tests" name="chrome">

<classes>

<class name="SeleniumGRID.GRIDChrome"/>

</classes>

</test> <!-- Test -->

</suite> <!-- Suite -->

EXCEPTIONS

Different Types of selenium Exceptions: (you will get only RunTime Exceptions in Selenium):

1. WebDriverException: The base exception class for all WebDriver-related exceptions.

2. TimeoutException: Thrown when a timeout occurs while waiting for an element or condition.

3. NoSuchElementException: Thrown when an element cannot be found in the DOM.

4. StaleElementReferenceException: Thrown when a previously referenced element is no longer attached to the DOM.

5. ElementNotVisibleException: Thrown when an element is present in the DOM but not visible on the page.

6. ElementNotSelectableException: Thrown when trying to interact with an element that is not selectable, such as a disabled checkbox.

7. InvalidElementStateException: Thrown when an element is in an invalid state for the requested operation, e.g., trying to click on a non-clickable element.

8. NoSuchWindowException: Thrown when a window or tab is not found.

9. NoSuchFrameException: Thrown when a frame or iframe is not found.

10. UnhandledAlertException: Thrown when an unexpected alert is present.

11. UnexpectedTagNameException: Thrown when a WebElement does not belong to the specified tag name, e.g., finding a div element when expecting an input element.

12. InvalidSelectorException: Thrown when an invalid selector is used to locate an element.

13. ElementClickInterceptedException: Thrown when another element is covering the target element, preventing it from being clicked.

14. MoveTargetOutOfBoundsException: Thrown when the target of a move action is outside the visible area of the browser viewport.

15. NoSuchAttributeException: Thrown when an element does not have the requested attribute.

16. InvalidCookieDomainException: Thrown when attempting to add a cookie with an invalid domain.

17. InvalidCoordinatesException: Thrown when the coordinates provided to an interaction method are invalid.

18. NoAlertPresentException: Thrown when there is no alert present to switch to.

19. ScriptTimeoutException: Thrown when the script execution exceeds the specified timeout.

20. SessionNotCreatedException: Thrown when a new session cannot be created.

21. UnhandledAlertException: Thrown when an unhandled alert is encountered during the test execution.

22. UnsupportedCommandException: Thrown when an unsupported command is used with the WebDriver.

23. WebDriverTimeoutException: Thrown when a timeout occurs while waiting for a WebDriver command to complete.

24. ElementNotInteractableException: Thrown when an element is present on the page but is not in an interactable state.

25. ImeActivationFailedException: Thrown when activating the input method editor (IME) fails.

26. ScreenshotException: Thrown when an error occurs while capturing a screenshot.

27. ElementNotDisplayedException: Thrown when an element is not displayed on the page.

28. ElementNotEnabledException: Thrown when an element is disabled and cannot be interacted with.

29. InvalidElementSizeException: Thrown when the size of an element is not valid or cannot be determined.

30. InvalidSelectorException: Thrown when an invalid selector is used to locate an element.

31. NoAlertOpenException: Thrown when attempting to interact with an alert that is not open.

32. NoSuchContextException: Thrown when the requested context (e.g., window, frame) does not exist.

33. NoSuchSessionException: Thrown when attempting to perform an action on a nonexistent session.

34. UnhandledWebDriverException: Thrown when an unhandled exception specific to the WebDriver implementation occurs.

|  |  |  |
| --- | --- | --- |
| **Name of the Exception** | **Description** | **Reason** |
| **java.lang.AssertionError:** | Expected vs actual | When we use Hard and Soft assert in test scripts (TestNG Assert) |
| **java.lang.AssertionError:** | 1 expectation failed.  Response body doesn't match expectation.  Expected: The content to match the given JSON schema. | Rest Assured |
| **java.lang.Error**: | Unresolved compilation problem:  Syntax error on token ".", Identifier expected after this token | Java – syntax error |
| **java.lang.IllegalArgumentException:** | The JSON input text should neither be null nor empty. | Check the json body if any parameter is missing |
| **StreamWriteException** | ObjectMapper class – writeValue method | Jackson databind – fasterxml |
| **DatabindException** | Object mapper object | Jackson databind – fasterxml |
| **IOException** | File input output stream | Excel Utility Usage |
| **java.net.ConnectException : Connection Refused : connect** | Rest Assured | Connectivity issue |
| **JsonMappingException** | Rest Assured - ObjectMapper.readTree(jsobBody) | Checked Exception =>to signal exception with mapping of json body contents |
| **JsonProcessingException** | Rest Assured - ObjectMapper.readTree(jsobBody) | Checked Exception =>to signal exception with mapping of json body contents |

Agile Ceremonies:

Sprint Planning meeting

Sprint Review meeting

Sprint Retrospective meeting

Session timeout exception - when the application takes a long time to load

IllegalArgumentStateException – driver is not supported

TimeoutException in Selenium – wait statements (When

No Alert Present Exception

No such element exception

ArgumentMismatchException – Dataprovider array arguments are given incorrectly

ClassCastException – When data