**Agile VS Kanban**

Parameter Agile Kanban

Application Agile is a beneficial method for projects where the final goal is not set. As the project progresses, the development can adapt as per the requirements of the product owner. Reducing waste and removing activities that never add value to the team.

Advantage Breaking the entire project into smaller segments helps the team to focus on high-quality development, testing, and collaboration. Conducting testing after every iteration helps the team to find and resolve bug quickly. Shorter cycle times can deliver features faster.

Focus Agile process focuses on constant communication. Shorter sprint lengths force to breaks up items to fit within sprint boundaries.

Involvement of QA QA has nothing to do at the beginning of a sprint but is overworked at the end. QA is involved in every phase to regularly test the system under development.

Iterative Development Agile process allows Iterative Development. Kanban process does not allow Iterative Development.

Dependency Process depends on Story Boards. Process depends on Kanban Boards.

Visual checking Not providing support for visually checking the work in progress. Visually check the work in progress.

Goal The goal of Agile approach is continuous Integration, development and testing. The goal of the Kanban approach is to improve the team’s process

Planning Sprint planning can consume the scrum teams time for an entire day. Need very less organization set-up changes to get started

Advantage With shorter planning cycles, it’s easy to accommodate changes at any time during the project management. Rapid feedback loops may result in more motivated, empowered and actively performing team members.

**API Mocking**

To install json server: npm install -g json-server

To Start Json Server: json-server --watch db.json

**API Testing Codes**

HTTP defines these standard status codes that can be used to convey the results of a client’s request. The status codes are divided into five categories.

1xx: Informational – Communicates transfer protocol-level information.

2xx: Success – Indicates that the client’s request was accepted successfully.

3xx: Redirection – Indicates that the client must take some additional action in order to complete their request.

4xx: Client Error – This category of error status codes points the finger at clients.

5xx: Server Error – The server takes responsibility for these error status codes.

#3) 2XX Series

200 (OK) : Defines that the request was correct.

201 (Created) : The value wrapped with the request has been created in the database. It is needless to say that the request was correct.

202(Accepted):Indicates that the request has been received but not completed yet. It is typically used in log running requests and batch processing.

204(No Content) : This status code means that the request was correct and received but there is no response to send to the client by the server.

205 (Reset Content):Indicates the client to reset the document which sent this request.

#2) 4XX Series

These are specific to the client-side error.

400 (Bad Request) : A bad request means that the syntax of the request was incorrect. It can happen if you have sent wrong parameters along with the request url or in the body of the request.

401 (Unauthorized): Indicates that the request requires user authentication information. The client MAY repeat the request with a suitable Authorization header field.

402 Payment Required (Experimental).

403(forbidden):Unauthorized request. The client does not have access rights to the content. Unlike 401, the client’s identity is known to the server.

404 (Not Found) : A response code 404 means that the server was connected but it could not find what was requested. You can normally see this status code when you request a web page which is not available.

405 (Method Not Allowed): The request HTTP method is known by the server but has been disabled and cannot be used for that resource.

407 (proxy authentication required):Indicates that the client must first authenticate itself with the proxy.

408 (Request Timeout): Indicates that the server did not receive a complete request from the client within the server’s allotted timeout period.

409: Conflict : The request could not be completed due to a conflict with the current state of the resource.

#3) 5XX Series

These are specific to the server-side error.

500 – Internal Server Error : The server encountered an unexpected condition which prevented it from fulfilling the request.

501 – Not Implemented :The HTTP method is not supported by the server and cannot be handled.

502 – Bad Gateway : The server got an invalid response while working as a gateway to get a response needed to handle the request.

503 – Service Unavailable

504 – Gateway Timeout

505 – HTTP Version Not Supported :The HTTP version used in the request is not supported by the server.

506 – Variant Also Negotiates

507 – Insufficient Storage

508 – Loop Detected

510 – Not Extended

511 – Network Authentication Required: Indicates that the client needs to authenticate to gain network access.

**BDD Cucumber**

Cucumber is an open source tool which aids BDD implementation in the following ways:

Executable Specifications

Automated Tests

Living Documentation

Features of Cucumber

At the core of Cucumber is the human-readable non-technical language called Gherkin which is used to write tests.

It allows the acceptance tests to be hierarchically constructed as features (user stories) -> scenarios (behaviors) -> steps (examples)

Gherkin is the language used to write acceptance tests in feature files as Specifications by Examples using Given-When-Then notation

that:

can be interpreted and executed by Cucumber

is very close to natural human language and can be easily learned by business users

extremely structured that there is no ambiguity for automation test engineers

Hooks in Cucumber

In a test execution, there will be blocks of code that always need to be run to set up (prerequisite tasks) and tear-down (post-run tasks) the related test artifacts like machine settings, browser states, etc.

The Java implementation of Cucumber provides two entities which are called hooks. They are @Before and @After. As the name suggests, Methods/functions/pieces of code defined within:

@Before hooks will be run before the first step of each scenario under the feature

@After hooks will be run after the last step of each scenario under the feature

Hooks are defined within the step definition file only.

Cucumber IQ

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Cucumber

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

1. keywords used inside CucumberOptions

feature, glue, dryrun,tags, monochrome, strict, plugin

There are different type of options present under @CucumberOptions. Lets look at each option one by one to understand it better.

Plugin: plugin Option is used to specify different formatting options for the output reports. Various options that can be used as for-matters are:

Note – Format option is deprecated . Use Plugin in place of that.

A- Pretty: Prints the Gherkin source with additional colors and stack traces for errors.

@RunWith(Cucumber.class)

@CucumberOptions(plugin = {"pretty" ,"html:Folder\_Name" ,

"json:Folder\_Name/cucumber.json" ,

"junit:Folder\_Name/cucumber.xml"})

public class RunYoursTest

{

// This class will be empty

}

DryRun: This option can either set as true or false (default value is false). If it is set as true, it means that Cucumber will only checks that every Step mentioned in the Feature File have corresponding code written in Step Definition file or not. So in case any of the function is missed in the Step Definition for any Step in Feature File, it will give us the message. So If you writing scenarios first and then implementing step definitions then add dryRun = true.

Strict: if strict option is set to false then at execution time if cucumber encounters any undefined/pending steps then cucumber does not fail the execution and undefined steps are skipped and BUILD is SUCCESSFUL.

and if Strict option is set to true then at execution time if cucumber encounters any undefined/pending steps then cucumber does fails the execution and undefined steps are marked as fail and BUILD is FAILURE. This is what the Console output looks like:

Monochrome: This option can either set as true or false (default value is false). If it is set as true, it means that the console output for the Cucumber test are much more readable. And if it is set as false, then the console output is not as readable as it should be. For practice just add the code ‘monochrome = true‘ in TestRunner class.

Features: Features Options helps Cucumber to locate the Feature file in the project folder structure.All we need to do is to specify the folder path and Cucumber will automatically find all the ‘.features‘ extension files in the folder.

Glue: It is almost the same think as Features Option but the only difference is that it helps Cucumber to locate theStep Definition file. Whenever Cucumber encounters a Step, it looks for a Step Definition inside all the files present in the folder mentioned in Glue Option.

It can be defined like-

Snippet Style: Cucumber generates code snippets in Underscore style by default. If you want to change the format of cucumber snippets then you can set snippet type in your cucumber options. There are two types of snippets,

1- SnippetType.CAMELCASE

2- SnippetType.UNDERSCORE

and if Strict option is set to true then at execution time if cucumber encounters any undefined/pending steps then cucumber does fails the execution and undefined steps are marked as fail and BUILD is FAILURE. This is what the Console output looks like:

2. Difference between DryRun and Strict in cucumberOptions?

dryRun: true. checks if all the steps have the stepdefinition

strict: true: will fail execution if there are indefined or pending tags.

3. And/OR operation in Tags

Separate a list of tags by commas for a Logical OR tag expression.

tags={"@first,@second"}= OR

Specifying multiple tag arguments creates a logical AND between each tag expression.

tags={"@first","@second"}= AND

tags={"~@Regression"}= NOT

4. Annotations in Cucumber

Cucumber hooks are blocks of code that runs before or after each scenario. It can be defined anywhere in project or step definition layers using methods @Before, @After. Cucumber hooks Annotations allow us to manage better code workflow and help in reducing code redundancy. Cucumber hooks are used in a situations where prerequisite steps before testing any test scenario is performed.

1. @Before Hook: It will execute before every scenario.

2. @After Hook: It will execute after every scenario.

5. Use of order function inside cucumber annotations

@Before(order = int) : This runs in increment order, means value 0 would run first and 1 would be after 0.

@After(order = int) : This runs in decrements order, means apposite of @Before. Value 1 would run first and 0 would be after 1.

6. Use of background keyword

Background in Cucumber is used to define a step or series of steps that are common to all the tests in the feature file. It allows you to add some context to the scenarios for a feature where it is defined. A Background is much like a scenario containing a number of steps. But it runs before each and every scenario were for a feature in which it is defined.

7. Keywords used in cucumber

Feature, Scenario, Scenario Outline, Given, When, Then, Examples

8. How to perform singleline and multiline comment in cucumber

single line comment: use #

multi line comment: press Ctrl+/

9. How data is passed in cucumber

Data-Driven Testing in Cucumber

Parameterization without Example Keyword

Scenario: Successful Login with Valid Credentials

Given User is on Home Page

When User Navigate to LogIn Page

And User enters "testuser\_1" and "Test@123"

Then Message displayed Login Successfully

Data-Driven Testing in Cucumber using Scenario Outline

Parameterization with Example Keyword

Scenario Outline: Successful Login with Valid Credentials

Given User is on Home Page

When User Navigate to LogIn Page

And User enters "<username>" and "<password>"

Then Message displayed Login Successfully

Examples:

| username | password |

| testuser\_1 | Test@153 |

| testuser\_2 | Test@153 |

Parameterization using Tables

Scenario: Successful Login with Valid Credentials

Given User is on Home Page

When User Navigate to LogIn Page

And User enters Credentials to LogIn

| testuser\_1 | Test@153 |

Then Message displayed Login Successfully

@When("^User enters Credentials to LogIn$")

public void user\_enters\_testuser\_\_and\_Test(DataTable usercredentials) throws Throwable {

//Write the code to handle Data Table

List<List<String>> data = usercredentials.raw();

//This is to get the first data of the set (First Row + First Column)

driver.findElement(By.id("log")).sendKeys(data.get(0).get(0));

//This is to get the first data of the set (First Row + Second Column)

driver.findElement(By.id("pwd")).sendKeys(data.get(0).get(1));

Maps in Data Tables with Header

In the previous chapter of Data Tables in Cucumber, we pass Username & Password without Header, due to which the test was not much readable. What if there will be many columns. The basic funda of BDD test is to make the Test in Business readable format, so that business users can understand it easily. Setting Header in Test data is not a difficult task in Cucumber. take a look at a below Scenario.

Feature File Scenario

Scenario: Successful Login with Valid Credentials

Given User is on Home Page

When User Navigate to LogIn Page

And User enters Credentials to LogIn

| Username | Password |

| testuser\_1 | Test@153 |

Then Message displayed Login Successfully

Data-Driven Testing in Cucumber using External Files

Parameterization using Excel Files

Parameterization using Json

Parameterization using XML

10. Use of ScenarioOutline and example

Scenario Outline is used to perform same scenario testing on different data set. Scenario Outline must have “Examples” keyword and in example we pass test data. In above feature file login scenario performed for four users (mercury, mercury1, mercury2 and mercury3).

11. What is dataTable

dataTable is used to parameterization in cucumber

12. Can user provide multiple examples for one scenario outline

yes

**Defect Life Cycle**

7. Defect Life Cycle

The number of states that a defect goes through varies from project to project. Below lifecycle diagram, covers all possible states

New: When a new defect is logged and posted for the first time. It is assigned a status as NEW.

Assigned: Once the bug is posted by the tester, the lead of the tester approves the bug and assigns the bug to the developer team

Open: The developer starts analyzing and works on the defect fix

Fixed: When a developer makes a necessary code change and verifies the change, he or she can make bug status as “Fixed.”

Pending retest: Once the defect is fixed the developer gives a particular code for retesting the code to the tester. Since the software testing remains pending from the testers end, the status assigned is “pending retest.”

Retest: Tester does the retesting of the code at this stage to check whether the defect is fixed by the developer or not and changes the status to “Re-test.”

Verified: The tester re-tests the bug after it got fixed by the developer. If there is no bug detected in the software, then the bug is fixed and the status assigned is “verified.”

Reopen: If the bug persists even after the developer has fixed the bug, the tester changes the status to “reopened”. Once again the bug goes through the life cycle.

Closed: If the bug is no longer exists then tester assigns the status “Closed.”

Duplicate: If the defect is repeated twice or the defect corresponds to the same concept of the bug, the status is changed to “duplicate.”

Rejected: If the developer feels the defect is not a genuine defect then it changes the defect to “rejected.”

Deferred: If the present bug is not of a prime priority and if it is expected to get fixed in the next release, then status “Deferred” is assigned to such bugs

Not a bug:If it does not affect the functionality of the application then the status assigned to a bug is “Not a bug”.

8. Severity and Priority

Priority: Low, Medium, High, Blocker

Severity: Low, Medium, High, Critical

**Deloitte IQ**

**1.What is the return type of window handle method?**

**Ans:Set of Strings**

**2.Write a program to print a prime number from 1 to 100 in the reversed string.**

**3.How to use delay in a project.?**

**Thread.sleep(); , implicit wait , explicit wait, fluent wait**

**4.When will you get a null point exception error?**

**NullPointerException is thrown when program attempts to use an object reference that has the null value.**

**5.There are 2 radio buttons on a web page, check if the button is already clicked. If not click on that.**

**Ans: using isSelected we can check the element and click on that.**

**6.Log into a webpage and check for the logged-in user name and verify if they match. (Condition: No access to Dom)**

**7.There are N frames, check for the availability of the given webelement in the frames. (Condition: Appearance of Webelement varies in frames for each login)**

**8.Selenium behavioral and scenario-based questions like when there is a login functionality username and password, and captcha how u will think to write a code using POM using TestNG annotations**

**9.When there is a new application developed, how u will think of preparing automation scripts**

**10.When u r a code reviewer. What are the major areas u think of while reviewing**

**11.What is the approach u follow before you start writing Automation scripts?**

**12.When u will start automation?**

**13.How to handle dynamic web table?**

**WebElement element1 = driver.findElement(By.xpath("//table[@id='users\_table']"));**

**List<WebElement> l = element1.findElements(By.tagName("td"));**

**for (int i = 0; i<l.size(); i++){**

**System.out.println("Total elements present in table are: " +l.size());**

**System.out.println("table contents: "+l.get(i).getText());**

**14.Difference Between Maven Surefire and Failsafe plugin difference?**

**the Failsafe plugin is designed to run integration tests while Surefire to run unit tests. maven-surefire-plugin is designed for running unit tests and if any of the tests fail then it will fail the build immediately.**

**15.How to validate logo in Selenium and a few basic selenium and OOps questions of Java.**

**Ans: using javascript executor.**

**Boolean ImagePresent=(Boolean)js.executeScript("return arguments[0].complete && typeof arguments[0].naturalWidth != \"undefined\" && arguments[0].naturalWidth > 0", image);**

**16.HashMap with excel to extract data**

**17.From a webpage, if we click on a link, then a popup window we have one header with HTML values and two checkboxes with no properties, write dynamic XPath for second check box and select it and validate if it got selected or not.**

**18. Just before 2 days of release if we get one critical defect, what you will do? – Risk Analysis.**

**19.Framework Explanation.**

**20.How you will link the feature file and step definition?**

**You need a cucumber runner file first and inside that you need to glue the step definition file with the cucumber feature. The thing is that in cucumber if you place the feature file, step definition file and runner file in the same package then automatically, it will be able to map the steps from the step definition file with the feature file.**

**21.String reverse without using a reverse function and find duplicate words.**

**Exception.**

**22.Dynamic Overriding used in your framework and he showed Webdriver driver = new ChromeDriver();**

**23.String Building and String Buffer and String — Expected to tell String pool memory.**

**24.Git Commands**

**25.SQL Statement to find in the desired table to find a duplicate.**

**group by having count(\*)>1**

**Roles and responsibilities in the current project**

**26.How to iterate in hashmap**

**1.Iterate through keys of the HashMap using keySet function of map**

**System.out.println("Iterate over HashMap Keys");**

**for(String strKey : employeeMap.keySet() ){**

**System.out.println( "Key (Employee Id) : " + strKey );**

**}**

**2.Iterate through Valuesof the HashMap using valuesfunction of map**

**System.out.println("Iterate over HashMap Values");**

**for(Employee employee : employeeMap.values() ){**

**System.out.println( "Value (Employee) : " + employee );**

**}**

**3) Iterate through keys and values of the HashMap**

**Use this approach if you are interested in retrieving both keys and values stored in the HashMap. Use the entrySet method of the HashMap to get the Set view of stored mappings in the form of Map.Entry object and use a for loop to iterate over the key-values.**

**27.Difference between hash map and hash table**

**1.HashMap is non synchronized. It is not-thread safe and can't be shared between many threads without proper synchronization code. ...**

**2.HashMap allows one null key and multiple null values. Hashtable doesn't allow any null key or value.**

**3.HashMap is fast and hashtable is slow.**

**28.How to check database value using selenium**

**Connection con = DriverManager.getConnection(dbUrl,username,password);**

**You can use the Statement Object to send queries.**

**Statement stmt = con.createStatement();**

**stmt.executeQuery(select \* from employee;);**

**29.JDBC connection string code**

**//Load mysql jdbc driver**

**Class.forName("com.mysql.jdbc.Driver");**

**//Create Connection to DB**

**Connection con = DriverManager.getConnection(dbUrl,username,password);**

**//Create Statement Object**

**Statement stmt = con.createStatement();**

**// Execute the SQL Query. Store results in ResultSet**

**ResultSet rs= stmt.executeQuery(query);**

**// closing DB Connection**

**con.close();**

**30.How to find Prime number using Java code**

**31.Who will take the final call for priority bugs**

**32.Entry criteria for testing**

**33. Advantages of using maven and Jenkins**

**34.How to schedule a job in Jenkins**

**The steps for schedule jobs in Jenkins:**

**click on "Configure" of the job requirement**

**scroll down to "Build Triggers" - subtitle**

**Click on the checkBox of Build periodically**

**Add time schedule in the Schedule field, for example, @midnight**

**Jenkins uses Cron Expressions. min hr day(month) month day(week): \* \* \* \* \***

**You can simply schedule hourly builds by just typing @hourly. or @monthly, @yearly, @weekly, @daily**

**35.Advantage of pom.xml file**

**36.In the maven project, what is the use of the settings.xml file?**

**Maven provides a settings file – settings.xml. This allows us to specify which local and remote repositories it will use. We can also use it to store settings that we don't want in our source code, such as credentials.**

**There is an example and explanation of all the elements. The settings element in the settings.xml file contains elements used to define values which configure Maven execution in various ways, like the pom.xml. These include values such as the local repository location, alternate remote repository servers, and authentication information.**

**37.In the system, where the jar files will be located while using pom.xml**

**root directory of project**

**38.How to execute the failed test cases again?**

**39.How you used listeners in your project?**

**40.What are all OS you worked on?**

**41.How to use parameters using TestNG?**

**42.Difference between string buffer and string builder?**

**43.Difference between Ant and Maven?**

**Ant and Maven both are build tools provided by Apache. The main purpose of these technologies is to ease the build process of a project.**

**There are many differences between ant and maven that are given below:**

**Ant Maven**

**Ant doesn't has formal conventions, so we need to provide information of the project structure in build.xml file. Maven has a convention to place source code, compiled code etc. So we don't need to provide information about the project structure in pom.xml file.**

**Ant is procedural, you need to provide information about what to do and when to do through code. You need to provide order. Maven is declarative, everything you define in the pom.xml file.**

**There is no life cycle in Ant. There is life cycle in Maven.**

**It is a tool box. It is a framework.**

**It is mainly a build tool. It is mainly a project management tool.**

**The ant scripts are not reusable. The maven plugins are reusable.**

**It is less preferred than Maven. It is more preferred than Ant.**

**44.When to stop testing?**

**45.Difference between interface and abstract? // most important**

**Abstract class and interface both are used to achieve abstraction where we can declare the abstract methods. Abstract class and interface both can't be instantiated.**

**But there are many differences between abstract class and interface that are given below.**

**Abstract class Interface**

**1) Abstract class can have abstract and non-abstract methods. Interface can have only abstract methods. Since Java 8, it can have default and static methods also.**

**2) Abstract class doesn't support multiple inheritance. Interface supports multiple inheritance.**

**3) Abstract class can have final, non-final, static and non-static variables. Interface has only static and final variables.**

**4) Abstract class can provide the implementation of interface. Interface can't provide the implementation of abstract class.**

**5) The abstract keyword is used to declare abstract class. The interface keyword is used to declare interface.**

**6) An abstract class can extend another Java class and implement multiple Java interfaces. An interface can extend another Java interface only.**

**7) An abstract class can be extended using keyword "extends". An interface can be implemented using keyword "implements".**

**8) A Java abstract class can have class members like private, protected, etc. Members of a Java interface are public by default.**

**9)Example:**

**public abstract class Shape{**

**public abstract void draw();**

**} Example:**

**public interface Drawable{**

**void draw();**

**46.How to set up the selenium grid and how to check whether the server is running or not?**

**1. Using Command line**

**On the command prompt, type java -jar selenium-server-standalone-2.30.0.jar -role hub**

**Another way to verify whether the hub is running is by using a browser. Selenium Grid, by default, uses Machine A's port 4444 for its web interface. Simply open up a browser and go to http://localhost:4444/grid/console**

**run the command in node machine:**

**java -Dwebdriver.gecko.driver="C:\geckodriver.exe" -jar selenium-server-standalone-3.4.0.jar -role webdriver -hub http://192.168.1.3:4444/grid/register -port 5566**

**2. Using JSON File**

**There are 2 ways to verify if the hub is running: one was through the command prompt, and the other was through a browser.**

**To run test scripts on the Grid, you should use the DesiredCapabilities and the RemoteWebDriver objects.**

**DesiredCapabilites is used to set the type of browser and OS that we will automate**

**RemoteWebDriver is used to set which node (or machine) that our test will run against.**

**47.What is the page factory?**

**Page Factory is a class provided by Selenium WebDriver to support Page Object Design patterns. In Page Factory, testers use @FindBy annotation. The initElements method is used to initialize web elements.**

**@FindBy: An annotation used in Page Factory to locate and declare web elements using different locators. Below is an example of declaring an element using @FindBy**

**@FindBy(id="elementId") WebElement element;**

**initElements(): initElements is a static method in Page Factory class. Using the initElements method, one can initialize all the web elements located by @FindBy annotation.**

**48.How to calculate defect density and test coverage**

**49.Write a feature file for the login scenario.**

**50.Are you working on the creation of a framework from scratch or you just modify the existing framework?**

**51.What is a test runner file?**

**TestRunner is a program used in Cucumber to access Feature file as copied in the picture below. Feature file is something that has user requirement scenarios written in English which gives more readability and understandability of the requirement which is called Gherkin language.**

**52.What is a POM.xml file? Why it is used?**

**53.What is a testNG.xml file? What is the benefit of using a testNG framework?**

**54.What is a hybrid framework? What are the components of a framework?**

**55.What are the benefits of creating a framework? Why POM approach is preferred?**

**56.Have you worked on the data-driven framework? WAP to fetch data from an excel file, just tell me the code.**

**57.From what type of files can we obtain test data?**

**excel, csv, text, json**

**58.How to find broken links in selenium?**

**see the following steps to identify broken links in Selenium**

**Collect all the links present on a web page based on the <a> tag**

**Send HTTP request for each link**

**Verify the HTTP response code**

**Determine if the link is valid or broken based on the HTTP response code**

**Repeat the process for all links captured with the first step**

**59.What are test listeners in selenium? What are the different types of test listeners?**

**60.What is the frame is selenium? Let’s say I have 3 frames, then how I can go to the 3rd frame from the 1st frame? And how to come back to the 1st frame?**

**switch to frames using index, name**

**driver.switchtoframe(); is used to switch b/w frames.**

**61.Have you worked on database testing? How many types of joins are there? What is the use of joins in SQL?**

**inner join, left outer join, right outer join, cross join, self join**

**62.What is the difference between cross join and inner join?**

**63.Suppose I alter the table, then I dropped the table, then I did a rollback, what can be the output?**

**Once you drop the table, rollback doesnot work.**

**64.What is a maven in selenium? How maven is useful?**

**65.What is a build life cycle in Maven?**

**66.What are the qualities of a good build too? On what basis you will select a build tool?**

**67.What is a wrapper class in java?**

**1. Wrapper Class can be used to convert an object into a primitive data type or its vice-versa.**

**2. The use of wrapper classes improves the performance of the program. 3. Wrapper class helps in the serialization of object & its vice versa; It can convert primitive data to objects. Sometimes we need to stream objects, in that case, the wrapper class can convert them into serialization format.**

**3.Subtraction & addition, these types of operations cannot modify the old value of the Wrapper class primitive integer; that’s why the Wrapper class is known as immutable.**

**int i = 100;**

**Integer intVal = new Integer(i);**

**we can say primitive data type is wrapped as an object.**

**68.What is the difference between array and collections in java?**

**69.What is the difference between final, finally, finalize?**

**70.What is the difference between SOAP and REST?**

**71.What are the different methods used in web services?**

**Get, Post, Put, Delete, Patch**

**72.Have you worked on GIT? What are the commands in GIT?**

**73.What is the overall check-in checkout procedure in GIT?**

**74. What are the exceptions that you have encountered in selenium?**

**75. What are the different types of testing?**

**76. Tell me the annotations in testNG by order of execution?**

**77. What are the pop-ups in selenium? How to handle windows based pop-ups?**

**using getwindowhandles and alert class**

**78. Can we handle user sessions in selenium?**

**We can perform session handling with the help of Selenium webdriver with a TestNG framework. To trigger different sessions, we shall use the attribute parallel in the TestNG XML file. A TestNG execution configuration is done in the TestNG XML. To create multiple sessions, we shall add the attributes – parallel and thread-count in the XML file.**

**79. How to capture screenshots in selenium? Tell me the code.**

**File file=(TakesScreenshot)driver).getScreenshotAs(OutputType.FILE);**

**FileUtils.copyFile(file,new File("path"));**

**80. Do you know any scripting language like VB script? Javascript? Can you automate web services using rest assured?**

**javascript,yes**

**DOCKER Commands**

Docker is a tool designed to make it easier to create, run and deploy applications by using containers.

Docker containers are lightweight alternatives to virtual machines and is uses the host OS.

You don't have to pre-allocate any RAM in cantainers.

docker pull <image name:tag>: to pull docker image from docker hub.. tag specifies version of image.

docker run <image-name:tag> or docker run <image-id>: to run the docker image

docker images: to list down all the images in our system

docker ps: to list down all the running containers

docker ps -a :to list down all the containers(whether they are running or not).

docker build -t[image-name]:tag: to build an image from docker file

docker build <path to docker file>: to build image from docker file

docker run --name "container-name" -p<host port>:<container port><image-name:tag>: to run newly built image.

docker start <container-id>: to start container.

docker stop <container-id>: to stop container.

docker rm <container name>: to delete container after stopping

docker rmi <image id>: to delete image

docker login: to login to docker hub repo from cli

docker commit <container id> Dockerhub/image-name: to create and push new docker image on docker hub

docker tab image-name user-hub-id/newimagename

docker push image-name/username: to push docker image to docker hub

docker container logs <container id>: to view container logs

docker container kill <container id>

How do you create a docker container from an image

docker run -it -d<image-name>

docker exec -it<container id>bash: to access a runing container

docker info

docker --help

docker version

Advance docker commands:

docker-compose build : this command is executed in the directory where YAML file is present for building the compose

docker-compose up : to execute docker compose yaml file

Docker Swarm: It is a technique to create and maintain a cluster of docker engines. Service deployed in any node can be acccessed on

other nodes in the same cluster.

this command creates a network of docker engines/hosts to execute container in parallel(for scaling up and high availability)

Features: High availability of services, Auto-load balancing, easy to scale-up deployments

docker swarm init --advertise-addr <ip-address>:to initialize docker swarm, it contains the docker manager

docker swarm join: run this command on other nodes to join containers to docker manager

docker swarm join-token

docker swarm leave: to leave the connection with docker manager

docker swarm leave --force: to exit from docker manager

docker service: this command is used to control any docker existing service(containers/compose/swarm/others). These commands can

only be run on docker manager.

docker service create --name "Ang-App" -p 4200:4200 demoapp1: to deploy application on manager node only.

docker service create --name "Ang-App" -p 4200:4200 --mode global demoapp1: to deploy applicaion an all nodes parallely

docker service ls

docker service ps <service-name>: to check on which nodes , particular service is running

docker service logs

docker service scale

docker service rm <service-name>

docker node ls: to check the running nodes linked to docker manager

docker node update --availability drain <node name>: to drain the node

docker node update --availability active <node name>: to active the node

docker compose is used to run mulitple containers applications

Keyworks used un docker file

FROM:

RUN

Docker is open source and written in Google’s Go language.

Docker is based on containerization (OS-level virtualization), where the physical server and operating system are virtualized to produce multiple instances.

Docker is an application container and runs one service or application in a runtime instance. Though, several applications can run independently in isolated containers

A Docker container is an environment that contains application code, software package, system tools, libraries, services, dependencies etc.

Docker provides a rapid development cycle for developers and sysadmins.

Developers can emphasize on building applications. Sysadmins can emphasize on deployment. While, both can still collaborate their work together on a shared framework.

Some of the benefits of docker are stated below:

Docker Hub provides tools to automatically build and deploy revisions to release latest patches quickly

It offers Secure and centralized repository to maintain application packages, version upgrades and define access to users

Docker helps to reduce the overall cost of enhancements due to simplified maintenance and management

It allows high availability of applications and increases productivity of developers

You can use Docker for packaging applications, scaling web apps, databases and backend services to deploy them automatically.

Docker components:

Docker Engine is a packaging tool which provides a powerful work flow for building and containerizing your applications. It helps to build, ship, run and manage the containers

Docker Hub hosts a registry to upload and manage your apps and offers automated workflows

Docker image (a build component) is a read only template, which is used to create Docker containers. You can reuse existing images, or build new as you require them

Docker registry (a store component) hold images in a public store (called the Docker Hub) or a private store

Docker container (a run component) holds applications along with dependencies/components/libraries. Container are built from Docker images

**GIT Commands**

**git init: to initialize local git repository**

**vim filename: to create any file**

**git status: to check status of your branch**

**git add filename: to add file to the staging area**

**git rm --cached <file>: to unstage**

**git restore <file>..." to discard changes in working directory**

**use "git restore --staged <file>..." to unstage**

**git commit -m "message": to commit the code to local repo**

**now again check git status**

**Now create remote repo on git gub and clone the repo**

**git remote add origin [clone path]: to connect to remote repo**

**git push origin master: to push the code to remote repo**

**git remote --v: to check push and fetch address**

**git pull origin master: to pull the code to local repo**

**git status Check status**

**git add [file-name.txt] Add a file to the staging area**

**git add -A Add all new and changed files to the staging area**

**git commit -m "[commit message]" Commit changes**

**git rm -r [file-name.txt] Remove a file (or folder)**

**Branching & Merging**

**Command Description**

**git branch: List branches (the asterisk denotes the current branch)**

**git branch -a: List all branches (local and remote)**

**git branch [branch name] : Create a new branch**

**git branch -d [branch name]: Delete a branch**

**git push origin --delete [branch name]: Delete a remote branch**

**git checkout -b [branch name]: Create a new branch and switch to it**

**git checkout -b [branch name] origin/[branch name]: Clone a remote branch and switch to it**

**git branch -m [old branch name] [new branch name]: Rename a local branch**

**git checkout [branch name] : Switch to a branch**

**git checkout - :Switch to the branch last checked out**

**git checkout -- [file-name.txt]: Discard changes to a file**

**git merge [branch name] :Merge a branch into the active branch**

**git merge [source branch] [target branch] : Merge a branch into a target branch**

**git stash: Stash changes in a dirty working directory**

**git stash clear: Remove all stashed entries**

**Sharing & Updating Projects**

**Command Description**

**git push origin [branch name]: Push a branch to your remote repository**

**git push -u origin [branch name]: Push changes to remote repository (and remember the branch)**

**git push: Push changes to remote repository (remembered branch)**

**git push origin --delete [branch name]: Delete a remote branch**

**git pull: Update local repository to the newest commit**

**git pull origin [branch name]: Pull changes from remote repository**

**git remote add origin ssh://git@github.com/[username]/[repository-name].git: Add a remote repository**

**git remote set-url origin ssh://git@github.com/[username]/[repository-name].git Set a repository's origin branch to SSH**

**Inspection & Comparison**

**Command Description**

**git log :View changes**

**git log --summary :View changes (detailed)**

**git log --oneline :View changes (briefly)**

**git diff [source branch] [target branch] : Preview changes before merging**

**Git Commands to Resolve Conflicts**

**1. git log --merge**

**The git log --merge command helps to produce the list of commits that are causing the conflict**

**2. git diff**

**The git diff command helps to identify the differences between the states repositories or files.This is useful in predicting and preventing merge conflicts.**

**3. git checkout**

**The git checkout command is used to undo the changes made to the file, or for changing branches**

**4. git reset --mixed**

**The git reset --mixed command is used to undo changes to the working directory and staging area**

**5. git merge --abort**

**The git merge --abort command helps in exiting the merge process and returning back to the state before the merging began**

**6. git reset**

**The git reset command is used at the time of merge conflict to reset the conflicted files to their original state.**

**7. git status**

**The status command is in frequent use when a working with Git and during a merge it will help identify conflicted files.**

**How to resolve merge conflicts using the command line**

**The most direct way to resolve a merge conflict is to edit the conflicted file. Open the merge.txt file in your favorite editor. For our example lets simply remove all the conflict dividers. The modified merge.txt content should then look like:**

**---------------------------------------**

**git pull --rebase origin master**

**git mergetool : to open the megre conflicts nad edit that**

**git rebase --continue**

**Groovy**

Groovy is an object oriented scripting language which is based on Java platform. Since SoapUI is also built on Java platform, you can customize your SoapUI tests to make them more efficient. Since the syntax of Groovy is similar to Java, its very simple to understand for Java developers.

String x

def o

x = "Hi"

log.info is used to print text data to console in log pane.

log.info(“Print Groovy”)

log.info can accept output of other groovy function calls or properties as argument.

'log.info(testRunner.testCase.name)'

context.fileReader=new BufferedReader(new FileReader("C:\\Users\\sahil.juneja01\\Desktop\\My\\groovy.txt"));

firstLine=context.fileReader.readLine();

String[] value=firstLine.split(",");

testCase.testSteps["Properties"].setPropertyValue("val1",value[0]);

testCase.testSteps["Properties"].setPropertyValue("val1",value[1]);

nextLine=context.fileReader.readLine();

if(nextLine!=null)

{

String[] next=nextLine.split(",");

testRunner.testCase.testSteps["Properties"].setPropertyValue("val1",value[0]);

testRunner.testCase.testSteps["Properties"].setPropertyValue("val1",value[1]);

testRunner.gotoStep(0);

}

**Maven**

Maven is a build automation tool used primarily for Java projects. Maven uses convention over configuration which means developers

are not required to create build process themselves. Maven provides sensible default behavior for projects. When a Maven project is

created, it creates default project structure and developer is only required to place files accordingly.

There are many problems that you may face during the project development which are discussed below:

Adding set of Jars in each project: In case of selenium projects, we need to add multiple jar files in each project.

Dependencies and Versions: Ensuring that the jar files and the required dependencies are added to the project for developing,

compiling and executing the same.

Maven structure

POM in Maven stands for Project Object Model. It is fundamental Unit of Work in Maven. It is an XML file which resides in the base

directory of the project as pom.xml. The POM contains information about the project and various configuration detail used by Maven

to build the projects.

POM also contains the goals and plugins. While executing a task or goal, Maven looks for the POM in the current directory.

It reads the POM, gets the needed configuration information, then executes the goal.

Maven is often considered to be an Intelligent, Comprehensive, Project management and also as a Build tool for the following features:

Intelligent:During the build process, when a certain life-cycle phase is triggered, Maven makes sure to trigger the previous phases

before coming to it.

Comprehensive: A single product is capable of managing a project right from the creation till deployment

Project Management: Manages the dependencies required for a project and also helps in creating project documentation sites to have a well defined collaboration among the team members

Build:Builds a project and derives distribution unit out of it (jar/war/ear)

Maven Plugins and their goals

1. maven-compiler-plugin: compiles the source code of the project. It has two important goals:

compile - is bound to the compile life-cycle phase and compile the main source files.

testCompile - is bound to the compile life-cycle phase and compile the test(junit) source files.

Each of these build lifecycles is defined by a different list of build phases, wherein a build phase represents a stage in the lifecycle.

For example, the default lifecycle comprises of the following phases (for a complete list of the lifecycle phases, refer to the Lifecycle Reference):

validate - validate the project is correct and all necessary information is available

compile - compile the source code of the project

test - test the compiled source code using a suitable unit testing framework. These tests should not require the code be packaged or deployed

package - take the compiled code and package it in its distributable format, such as a JAR.

verify - run any checks on results of integration tests to ensure quality criteria are met

install - install the package into the local repository, for use as a dependency in other projects locally

deploy - done in the build environment, copies the final package to the remote repository for sharing with other developers and projects.

These lifecycle phases (plus the other lifecycle phases not shown here) are executed sequentially to complete the default lifecycle. Given the lifecycle phases above, this means that when the default lifecycle is used, Maven will first validate the project, then will try to compile the sources, run those against the tests, package the binaries (e.g. jar), run integration tests against that package, verify the integration tests, install the verified package to the local repository, then deploy the installed package to a remote repository.

site: generate the project's site documentation

**Selenium Grid**

**Selenium grid is part of the selenium suite which distributes the test across multiple physical or virtual machines, so that the test scripts can be executed in parallel (Simultaneously). It helps us in covering the test process across browsers and across different platforms.**

**Grid makes use of the concept of hub-node where you run the test only on the hub machine, but the execution will be done by different machines called as nodes.**

**The key points about execution of selenium project through Grid:**

**The RemoteWebdriver is used to instantiate the webdriver instance.**

**The DesiredCapabilities object is used to set preferences about the browser.for example: chrome or Firefox**

**The URL contains the node remote URL as well as the port number.**

**The execution which gets triggered in the HUB gets executed through the Node instances.**

**DesiredCapabilities cap = DesiredCapabilities.internetExplorer();**

**cap.setBrowserName("internet explorer");**

**String Node ="http://10.82.189.5:5558/wd/hub";**

**driver = new RemoteWebDriver(new URL(Node), cap);**

**The following things should be inferred from the code above:**

**DesiredCapabilities object is used to set the browser preference. The browsername needs to be set as “internet explorer” as same desired capabilities object is used for edge browser as well.**

**The RemoteWebDriver method signifies that you are trying to execute the script in a remote machine.**

**The RemoteWebDriver accepts two arguments. First one is the node URL, copied from the configuration pane in grid console and the second one is the desired capabilities object. The URL should be appended with the string wd/hub before providing that as an argument for RemoteWebDriver.**

**1. Using Command line**

**On the command prompt, type**

**java -jar selenium-server-standalone-2.30.0.jar -role hub**

**Another way to verify whether the hub is running is by using a browser. Selenium Grid, by default, uses Machine A's port 4444 for its web interface. Simply open up a browser and go to http://localhost:4444/grid/console**

**run the command in node machine:**

**java -Dwebdriver.gecko.driver="C:\geckodriver.exe" -jar selenium-server-standalone-3.4.0.jar -role webdriver -hub http://192.168.1.3:4444/grid/register -port 5566**

**Steps to set up selenium hub**

**1. Run command to start hub on your local**

**java -jar selenium-server-standalone-3.141.59.jar -role hub**

**2.Hit the link in browser to check hub is up and running**

**http://192.168.1.5:4444/grid/console**

**3. Now go to node machine and run the below command to register node to hub**

**java -jar selenium-server-standalone-3.141.59.jar -role node -hub "Hob node address"**

**java -jar selenium-server-standalone-3.141.59.jar -role node -hub " http://192.168.1.5:4444/grid/register/"**

**java -Dwebdriver.chrome.driver="C:\\Users\\sahil.juneja01\\Desktop\\chromedriver\_win32 (7)\\chromedriver.exe" -jar selenium-server-standalone-3.141.59.jar -role node -hub " http://192.168.1.5:4444/grid/register/"**

**java -Dwebdriver.edge.driver="D:\chromedriver\msedgedriver.exe" -jar selenium-server-standalone-3.141.59.jar -role node -hub " http://192.168.1.5:4444/grid/register/"**

**String, Stringbuilder**

Java provides three classes to represent a sequence of characters: String, StringBuffer, and StringBuilder. The String class is an immutable class whereas StringBuffer and StringBuilder classes are mutable. There are many differences between StringBuffer and StringBuilder. The StringBuilder class is introduced since JDK 1.5.

A list of differences between StringBuffer and StringBuilder are given below:

No. StringBuffer StringBuilder

1) StringBuffer is synchronized i.e. thread safe. It means two threads can't call the methods of StringBuffer simultaneously. StringBuilder is non-synchronized i.e. not thread safe. It means two threads can call the methods of StringBuilder simultaneously.

2) StringBuffer is less efficient than StringBuilder. StringBuilder is more efficient than StringBuffer.

**XPATH**

Relative XPath in Selenium

Starts from the middle of the HTML DOM.

Starts with a double slash “//” which means it can start to search anywhere in the DOM structure.

Shorter than Absolute XPath.

Less fragile.

Example://span[text()=’About Software Test Academy’]

RelativeXpath Syntax : //tag[@attribute='value']

Example: //img[@alt='abc']

//input[@type='send text']

//label[@id='clkBtn']

//input[@value='SEND']

//\*[@class='swtestacademy'] --> "\*" means, search "swtestacademy" class for all tags.

//a[@href='http://www.swtestacademy.com/']

//img[@src='cdn.medianova.com/images/img\_59c4334feaa6d.png']

Contains XPath in Selenium

It is a very handy XPath Selenium locator and sometimes it saves the life of a test automation engineer. When an attribute of an element is dynamic, then you can use contains() for the constant part of the web element but also you can use contains() in any condition when you need.

Syntax: //tag[contains(@attribute, ‘value‘)]

Example: //img[contains(@alt,’Baskirt’)]

Starts-with Xpath Examples

This method checks the starting text of an attribute. It is very handy to use when the attribute value changes dynamically but also you can use this method for non-changing attribute values.

Syntax: //tag[starts-with(@attribute, ‘value‘)]

Example: //img[starts-with(@alt,’Onur Ba’)]

How to find XPath by Text

We can find an element with its exact text.

Syntax: //tag[text()=’text value‘]

Example://span[text()=’About Software Test Academy’]

Xpath=//tagname[@attribute='value']

// : Select current node.

Tagname: Tagname of the particular node.

@: Select attribute.

Attribute: Attribute name of the node.

Value: Value of the attribute.

Contains:

//tag-name{contains(@attribute,'value)]

Starts-with

//tag-name[starts-with(@attribute,'value')]

How to find XPath by Text

text()

//tag-name[text()='text value'

Example://span[text()=’About Software Test Academy’]

XPATH AXES

Now, there can be scenarios that, even after using the above functions, sometimes the user is not able to locate the web elements uniquely.

In those scenarios, the XPath axes come as a rescue and help locate a web element with the help of its relationship with other web elements

on the web page. Let's understand what XPath provides all different types of axes in Selenium.

What are the XPath Axes in Selenium?

We read that XPath in Selenium uses the Absolute and Relative paths to locate the web elements. Additionally, all the web elements in the

XML DOM are related to each other via a hierarchical structure. XPath provides specific attributes that are called "XPath Axis", and

these use the relationship between various nodes to locate those nodes in the DOM structure. The following table shows a few of those Axis, which can locate the elements on a web page using XPath in Selenium.

Axis Description

ancestor This axis locates the ancestors of the current node, which includes the parents up to the root node.

ancestor-or-self This axis locates the current node as well as its ancestors.

attribute This axis specifies the attributes of the current node.

child This axis locates the children of the current node

descendant This axis locates the descendants of the current node, i.e., the node's children up to the leaf node.

descendant-or-self This axis locates the current node and its descendants.

following This axis locates all nodes that come after the current node.

following-sibling This axis locates the below siblings of the context node. Siblings are at the same level as the current node and share its parent.

parent This axis locates the parent of the current node.

preceding This axis locates all nodes that come before the current node.

self This axis locates the current node.

TESTNG

TestNG (NG stands for Next Generation) is an open-source automation testing framework. It is inspired from testing frameworks Junit and NUnit (unit testing frameworks for Java and C# respectively). TestNG framework helps in unit, functional, end to end and integration testing etc.

Major features of TestNG:

Annotations helps in easier differentiation of methods

Helps in prioritizing and grouping of tests

Supports data-driven testing (@Data Provider)

Parallel test execution is possible

Supports parameterizing the test cases

Supported by multiple tools and plugins

The annotations available in TestNG are listed below:

https://lex.infosysapps.com/en/viewer/web-module/lex\_auth\_01297633668276224011808?collectionId=lex\_24050900789800050000&collectionType=Course