**[Walmart](#_Walmart_Interview_Questions:)** [**HCL**](#_HCL_Interview_Question)[**Mphasis**](#_Mphasis_Interview_Questions)[**QSSTech**](#_QSSTech_questions)[**Cigniti**](#_Cigniti_Questions:)

[**Swiggy**](#_Swiggy)[**Credit Karma**](#_Credit_Karma_interview)[**Infosys**](#_Infosys_Interview_Questions)[**Wells Fargo**](#_Wells_Fargo_Interview)

[**RateGain**](#_RateGain_Interview_questions)[**Kapindra**](#_KAPINDRA_INTERVIEW:)[**ATOS**](#_ATOS) [**SMBC**](#_SMBC_interview_questions)

[**Should Know**](#_SHOULD_KNOW)

# **Walmart Interview Questions:**

1. current project and its end to end architecture of automation.
2. [how a cron job is set up for running the automated tests](#_2._How_a)
3. how do we test the mobile application with appium.
4. [4 major components of selenium](#_4._Four_major).
5. [what is the needs of automation when we are doing the integration testing.](#_What_is_the_1)
6. how do we do integration testing
7. how bdd test scenario's work
8. how to set up the pipeline to run the tests.
9. [how much time does it take to run the current project automated test scenarios.](#_How_much_time)
10. is there a way we can reduce the time and run these test multiple times a day.
11. Who owns Github for automation code developer or automation engineer?

**Ans.** Automation Engineer.

1. different types of framework that i have worked on
2. how do automation engineer make use of DB.
3. no sql database use cases as an automation engineer.
4. [Difference between Overloading and Overriding](#_15._Difference_between)
5. Explain your current project Framework?
6. [Explain Implicitly wait() & explicitly wait()](#_17._Explain_Implicitly)
7. [Difference between CSS and xpath?](#_18._Difference_between)
8. [How to access browser?](#_How_to_access)
9. [Why WebDriver driver=new FirefoxDriver () why not WebDriver driver= new WebDriver()?](#_Why_WebDriver_driver=new)
10. [What is POM?](#_21._What_is)
11. [Difference between POM and page factory](#_22._Difference_between). -
12. [Explain Error,bug and defect](#_Difference_between_Error,)
13. [Tell me about TestNG](#_What_is_TestNG?)
14. [Basic Tools for Mobile Automation testing?](#_Tools/software's_to_be)
15. [Explain Hooks in cucumber](#_26._Explain_Hooks).
16. [What is the difference between Background and Scenario](#_27._What_is)
17. [How do you pass data value input to a scenario in feature file](#_28._How_do)
18. [Important parameters of Cucumber Runner class](#_28._Important_parameters)
19. [Java Program to print word count](#_30._Java_Program)
20. [Java program to print Alphabets A to Z](#_31._Java_program)
21. Java program for add 2 number to match the given target value and print the indexes.
22. Program to merge two sorted arrays (should not use any in-built & temporary array)
23. An array which has 1010101010 as elements, program so that array element should have all 1's on right and 0's are left (without using the temporary array)
24. [How hashmap works (internally)](#_How_hashmap_works)
25. program to delete a node from a tree.
26. Explain linked list & how to traverse
27. Puzzles
28. Scenario-based question on e-commerce domain.
29. Design Automation framework and explain why POM

[Top](#_top)

# **HCL Interview Question Bank**

1. Explain the process to push your code into repo?
2. How do you configure Jenkins ?
3. [Overloading and overriding](#_15._Difference_between)
4. If, switch, for and while loops
5. How did you implemented or used oops concept in your current project
6. [TestNG annotations](#_What_are_different)
7. I was asked to open a browser to locate 2 elements.
8. Explain your current project
9. Explain cucumber framework

10.Difference between scenario and scenario outline

11.[Why do we use the Glue keyword in cucumber?](#_28._Important_parameters)

HCL interview questions

\*Tell me about yourself and your current project

\*Rate yourself in java and selinium

\* What framework you are using for automation and explain your framework

\* What Programming language are you using for your framework **Ans.** Java

\* [what is a constructor and interface](#_What_is_a)

\* [Different types of constructors used in your project](#_Different_types_of)

\* Explain diff weights used in your project

\* can we overload a constructor **Ans.** Yes.

\* [Explain abstraction](#_Explain_abstraction)

\* did you use jenkins for running your scripts

\* take one scenario (Ex:login page) and explain the steps for end to end testing

\* if you team have given 500 test cases to run among them you have to run 100 test cases then how will you choose your test cases.

\* different types of exceptions you encountered in your project and what is your approach to rectify them.

\* API experience.

\* ci/cd pipeline

[Top](#_top)

# **Mphasis Interview Questions**

1)[What is method overloading and method overriding.](#_15._Difference_between)

2)can constructor be overloaded in Java? **Ans. Yes**

3)Exception handling

4)[Write a program to check the two strings are anagram without sorting](#_Write_a_program)

5)[Listener and annotation  in testNG](#_What_are_different)

6)Cucumber Framework

7)How to take a screenshot of a failed test case.

8) [What is the StaleElementReferenceException? How to fix it?](#_What_is_the)

9)[What are the challenges with mobile automation?](#_What_are_the_1)

10)what should be covered in mobile automation like it should be responsive

11) did you use something for performance in mobile automation.

12)[Hooks and background](#_26._Explain_Hooks).

[Top](#_top)

# **QSSTech questions**

1) [Appium Architecture](#_Appium_Architecture)

2) About different exceptions that I have encountered till now, while doing programs

3) [Difference between REST API and Restful APIs](#_Difference_between_REST)

4) About the use of constructor

5) [About different Request Methods of APIs](#_About_different_Request)

6) What are the good attributes of a good framework?

7) Program to find largest from the array

8) Program to print reverse of a string

9) How to handle authorizations?

10) Challenges I have faced till now in automation

11) How to handle popup and alerts?

12) How to make a plan based on What should be automated and what should not as you are working in an agile environment?

[Top](#_top)

# **Cigniti Questions:**

What maven plugin to execute all test?

Program to print

\*

\*\*

\*

\*\*

[Program for word count](#_Java_Program_to)

Program for string reverse

[Annotations on TestNG?](#_What_are_different)

How to skip a test in TestNG?

How to find the locators in Android Mobile Automation?

Define static variables and methods

[How to switch apps during the execution of Mobile Automation (while in contacts app, switch to camera app)](#_How_to_switch)

[How to scroll in mobile automation](#_How_to_scroll),

How to handle devices with different dimensions

How to find locators for iOs and Andriod apps

How to choose "About Phone" options under "Settings" menu app screen

[Tools/software's to be installed/required before starting Mobile Automation](#_Tools/software's_to_be)

List of different files in Cucumber framework

Difference between Scenario and Scenario Outline

How/where to define the global variables (like values for Username, Password)

How to handle multiple elements located?

List of main capabilities to be passed to start the application

[Explain the "Background" in Cucumber framework](#_27._What_is)

[Explain Appium Architecture?](#_Appium_Architecture)

How to handle Alerts/Popup window in Mobile Automation?

[What are the few Exceptions in Appium that you have came across?](#_Exceptions_found_during)

[Top](#_top)

# **Swiggy**

Questions: Structure of framework, [How we are testing the APIs](#_How_are_we), asked for [scenarios on example API](#_Scenarios_on_example), Integration test, regression test, functional testing,

Codes:

1- Combination of numbers that would give expected sum

2- Max and min in an array in O(n)

3- Duplicate characters in a string

4- Similar characters on two different strings

**Swiggy Interview Questions:**

1. What is native and hybrid applications ?

2. [What is smoke, sanity and regression testing?](#_What_is_smoke,)

3. Open swiggy app, click on item which you want to buy and explain how will you test the page.

4. Add an item to cart and explain how will you test the screen.

5. Suppose we have release today, how will test the app and ensure that the release is safe.

6. The items which you are seeing in the "Top Picks for You" section what are they generally called in mobile terminology.

7. How you get ios build?

8. How you get android build?

9. In which countries the application you have tested are used?

10. What scenarios will you test in "Top Picks for You" section. Explain them in detail and also why?

11. Suppose there are only 4 items in "Top Picks for You" section. How will you test the scroller in that?

12. Test Make payment page

13. Test Popular Brands page

14. Perform negative testing in Top Offers section

15. Who have better understanding of the application? - developer or tester

16. Perform testing on Tip section and give some out of the box scenarios

17. What is Agile Methodology?

18. [What is API Testing](#_What_is_API)?

19. [What are different error status code?](#_Types_of_API)

20. [How you test API in postman? Please explain.](#_How_you_test)

21. [What are different http methods. Explain them in detail.](#_About_different_Request)

22. [What is the difference in put and patch?](#_About_different_Request)

23. Suppose while ordering any item 10th time it is giving blank screen, how will you report this to developer?

1) Design Patterns

1.1) Page Object Model

1.3) MVC

1.4) Singleton Design pattern

2) Asked about framework

3) If I have given a request then how will I automate it using framework?

4) In which case, we may get Unknown host exception

5) Have I ever faced Connection refused issue or not?

7) What is Searialization?

8) Which build tool we use in our project?

9) How we can create jar through maven?

10) What advantages Maven offers ?

11) Why do we use static class?

12) What I will do If am getting a 500 status issue? What steps would I take to analyze it

13) Asked to write a program for performing reverse using recursion

14) I have ever worked on setting up JDBC connection and written connection script or not?

15) How many times we establish connection while using in the script to get the data

16) How do I apply assertions?

[Top](#_top)

# **Credit Karma interview questions**

**•** [Git merge conflict.](#_Merge_Conflict)

• Different exceptions in selenium

• [Different waits in selenium](#_17._Explain_Thread.sleep,)

• [Why StaleElement exception occur? How to resolve it?](#_What_is_the)

• [Explain BDD](#_Explain_BDD)

• Explain feature file in BDD?

• [Appium architecture](#_Appium_Architecture)

• [How to check when an api crashes?](#_How_to_check)

• [What is adb ? how do you use adb in Appium?](#_What_is_adb)

• Program to Reverse the String without using any stringbuilder.

• How is your release cycle?

• Adb logs

• Interfaces and classes

• Can an abstract method be overloaded? **Ans. Yes**

• How did you implement interface in your project?

[Top](#_top)

# **Infosys Interview Questions**

**​​·** Find repetitive characters in a string.

·  OOPS concept explain

·  Explain your framework.

·  POM explains and gives advantages.

·  How to take a screenshot in selenium

·  Aws

·  [Api testing](#_What_is_API)

·  Have you worked on data driven testing?

·  Parallel testing

·  Problems faced in selenium

·  Different exceptions

·  Jenkins to create a pipeline

·  Git problems, how were they taken care

[Top](#_top)

# **Wells Fargo Interview Questions**

**Q1: Output?**

class TestApp {

public static void main() {

        int odd = 1;

        if (odd) {

            System.out.println("odd");

        } else {

            System.out.println("even");

        }

    }

}

**Ans. Gives compile time error.**

**Q2: Output?**

class TestApp {

    static int index = 0;

    public static void main(String args[]) {

        System.out.println(test());

    }

    int test() {

        int index = 1;

        return index;

    }

}

**Ans. Prints 1**

​

​

Q3.1: Write a cucumber scenario in feature file for an API which return customer name, address and balance

Q3.2: Explain how you will implement this scenario in the framework (library, test runner, stepdefs location and package strategy)

Q4.1: [Difference between Overloading and Overriding](#_15._Difference_between)

Q4.2: Can we override a static method

Q5: What is Super keyword in JAVA

Q6: [Tools for API testing?](#_Tools_for_API)

Q7: Write a generic locator for options under ‘Dresses’ dropdown in the website (<http://automationpractice.com/index.php>)

Q8: Write a maven command to execute a cucumber test with passing the test environment and explain how to handle it in the framework

Q9: How to execute tests in parallel on remote machine

Q10: Where screenshots gets stored on executing the tests on mobile device from jenkins

Q11: Which Java collection can be used to get the data from excel?

Q12: [How to execute any methods before and after any cucumber tests? (cucumber hooks)](#_26._Explain_Hooks)

Q13: Explaining cucumber runner file elements

Q14: Explaining any test automation framework and its components?

Q15: 4 things to keep in mind while creating any test automation framework

Q16: Explain framework that you are working currently

Q17: How to inspect element for web application in mobile browser

Q18: is there any difference on the locator those are present for web application when we inspect in desktop and in mobile browser

Q19: Assume we have a new project then from where we have to start automation

Q20: How to manage test cases result to be published automatically on Jira/Excel to respective test cases using the test automation framework

Q21: Thought over running test execution on virtual machine compare to physical machines

Q22: Benefit of using BDD in test automation

Q23: Key and values(Desired Capabilites) that need to provide while inspecting any android application

Q24: Url for appium server to execute automation on mobile application

Ans.

http://localhost:4723/wd/hub  
  
[Top](#_top)

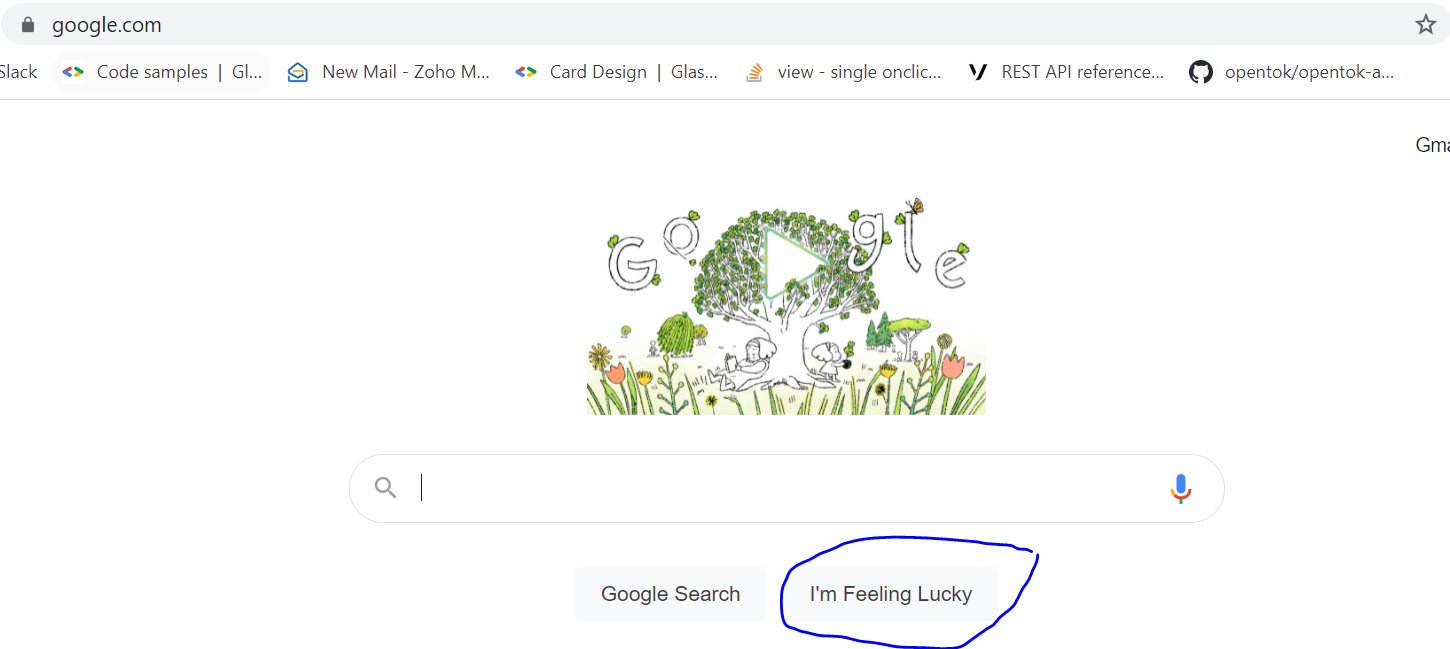
# **RateGain Interview questions**

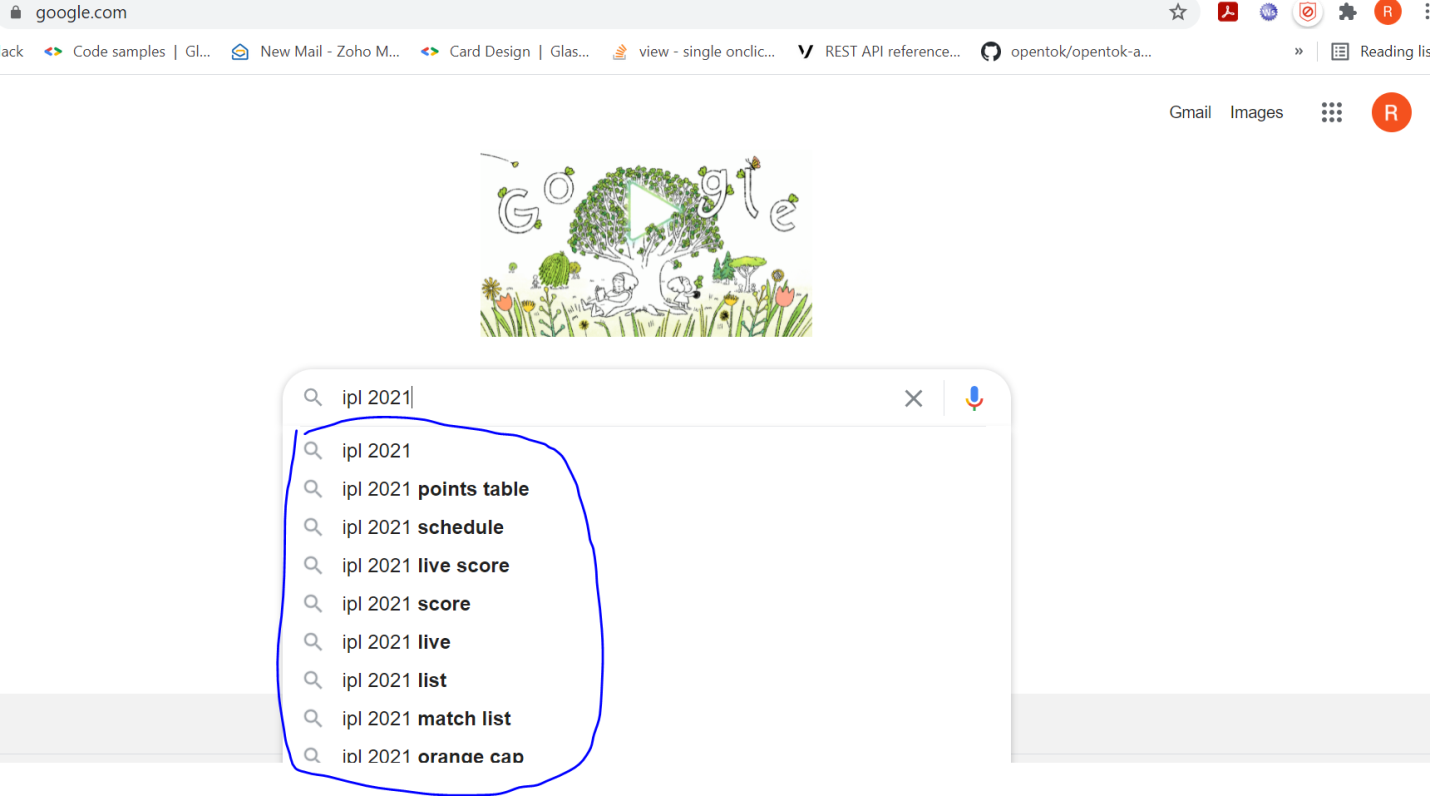
[**API testing**](#_What_is_API)  
[status code](#_Types_of_API)  
Jason and Xml format (have understanding about that)  
Questions: if you know about AWS Simple Cloud Storage (S3), what will be your approach to automate

**Database**Joins (all)2DeleteWhere condition count, average Group

Quistions:   
1)if you have two tables, you need to delete records from second table which belong to particular city of USA  
 2) Select no. of records from others table groupwise

**Selenium:**Basic question of selenium (locaters, xpath, exceptions)

Questions:  
1) Find the xpath of highlighted text in below image   


2.) What will be your approach to automate suggestion appears when you do search any particular things on Google, as you can see in below image  


3) Have you face any challenges earlier while doing automation testing, if yes please explain   
4) Explain about frame work which you are used earlier (overview)

**JAVA**  
String manipulation   
1) Write a program to reverse the string without using method   
2) Basic question of oops (java)

General question   
Situational question ( about time management while you are doing work on multiple project or single project with short deadline )

[Top](#_top)

# **KAPINDRA INTERVIEW:**

1st Round

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

A-Project based

B-[API basic](#_What_is_API)

C- Test scenario to book an hotel for goa though makeMyTrip

D-How to specify load during penetration testing?

E-Define main 3 timers in JMeter(Constant, Gausian and Synchronization)

F-[How to test 1000 load on API, way of putting loads](#_How_to_test)

G-How to doo Parameterization in JMeter

H-What is Throughput?

I-What is ramp-up period?

J- [Issues faced while API testing?](#_Issues_faced_while)

K- [How to use SWAGGER? Can we perform API testing from same?](#_How_to_use)

L- Few more completely sccenario based questions.

M-What if resource is not performing well, will you throwout the team or how you handle?

2nd Round

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

A-Testing type and explanations.

B-Acceptance testing, how to perform?

C-[API testing basic](#_What_is_API)

D-[what is environment in postman?](#_What_is_environment)

E-[How to perform API testing and what should be the standard process to perform the same?](#_How_to_perform)

F-[How to validate the error logs of API?](#_How_to_validate)

G-Are you also responsible for build deployment?

H-Have you ever worked on linux, if yes then share few commands?

I-4 people have to cross a bridge in dark night, and there is only 1 tourch and without tourch you cant travel

A will cover in 1 min

B will cover in 2 mins

c will cover in 5 mins

D will cover in 08mins

only two person can go in single time and total time is 15 minutes. [Suggest the case in which everyone can go other side](#_Bridge_Riddle)

within 15 mins.

J-3 boxes are labled wrong, one box has red balls, another has blue and last one has mix, [now tell the way to correct the label of box](#_Wrongly_labelled_Box)

you can only open one box thorough out the process.

K-Current project-based questions

L-Bug life cycle

M- How to perform Load testing, how to use parameters in Jmeter?

N- What are the different different timers in Jmeter.

Priya Interview Questions:

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

1. what is sql and types of SQL commands
2. how to delete data from table using sql query
3. Project related questions
4. What can be the test cases if I search for hotel in london on any website (such as make my trip)
5. [what do you verify in the API response](#_What_do_you)
6. [Types of API response codes](#_Types_of_API)
7. [what is web services](#_What_are_web)
8. what are the major challenges faced by you while automating
9. questions about framework you have worked on
10. DataBase queries on below commands

* WHERE  , INSERT INTO, ALTER DATABASE, DROP

**Shubham Interview questions:**

**Manual Interview Questions:**

1. [What is smoke, sanity and Regression testing?](#_What_is_smoke,)

2. [What is the difference between Smoke and Sanity](#_What_is_smoke,)?

3. How much time do you take to perform smoke, sanity and regression testing?

4. From where do you normally download build in ios devices?

5. In which projects you have worked in Agile Methodology and what is that?

6. What is the test plan?

7. Who knows better about the application- developer or tester? **Ans**. Both in BDD framework.

8. What is the defect life cycle?

9. What is severity and priority?

**SQL Interview Questions:**

1. What is the command used to fetch the first 5 characters of the string?

2. How to fetch common records from two tables?

3. How to fetch alternate records from a table?

4. What is the command used to fetch the first 5 characters of the string?

**Java Interview Questions:**

1. [Write a program to read data from a file](#_Java_program_to_1)

2. Write a program to read data from Xlsx file

3. Write a program to write data in Xlsx file

4. Why is File Class used in read file code?

5. Why XSSFWorkbook Class is used in read file code and for which excel format it is used?

6. Why is XSSFSheet Class used?

7. WAP to read data from H19 cell of an excel sheet

8. What is String? Is it a mutable or immutable class? Explain with examples.

**Selenium Interview Questions:**

1. [What are different annotations used in TestNG?](#_What_are_different)

2. Write a code for opening the browser

3. [What are different types of wait?](#_17._Explain_Implicitly)

4. Apply Explicit wait in your code

5. Explain keyword driven framework

6. [What is Maven?](#_What_is_Maven?)

# **Dhawal’s list of questions**

1. If you get an automation task, what the process you follow

2. What is the definition of done for you?

3. How do you report automation bugs?

4. How do you contribute to a release from the automation side?

5. Why automation is needed?

6. Can you achieve 100% automation in a project? **Ans.** No

7. [What time does your automation take to run 1000 test scenarios? Can you make it faster?](#_What_time_does)

8. Have you implemented parallel automation, if yes how?

9. basics of cURL

10. [what is the git workflow you used in your company](#_Git)

Merge Commit

Rebase

Pull Request

Git fetch --all

Git pull

Git push

11. Test Run, Test Cycles

12. Test Rail

# **ATOS**

Tell me about yourself?

Day to day activities?

What will you consider choosing a right automation tool?

What will be the process of creating framework from scratch?

How will you write your test script effectively?

Assert and verify??? Where will you use?

Default value of implicit wait?

Selenium drag and drop command?

Selenium keyboard actions?

Challenge which you face?

Priority and severity definition with examples?

Sql question

Inner join

Left inner join

Self join

One to many relationships

NEW

Tell me about yourself and your experience of auotmation?

\* Any experience working with Scratch from automation framework? What is your contribution?

\* What Framework do you prefer to work and why?

\* [Diff b/w Implicit and explicit wait? What do you prefer to use?](#_Explain_Thread.sleep,_Implicitly)

\* What will do if you see the test cases fail while running regression suite?

\* How can you reduce page load time?

\* How can you check if textbook is enabled or disabled?

Also She asked about my rating in sql, and I replied her I'm comfortable in using joins and can write queries

Difference b/w priority and severity and example for high priotrity and low severity defect

# **SMBC interview questions**

How you write the test cases

Android architecture

Exceptions u face and how to handle that

Final and finally (difference)

Oops concept in cucumber framework

API’s http-201 and 102

In TestNG if we hv 5 scenario and did not gave priority how it will execute

Cherry picking term in GITHUB

How much time it will take to run 1000 test cases on sauce lab

Challenge u faced in mobile automation

Third round

My yesterday's third round of interview with SMBC was all about the process that is followed in agile methodology :

1)about user stories

2)How much time a user story should be finished.

3)Sprint backlog/product backlog

4) how many user stories you have covered in max international since 2020.

5)Which document are you using for writing the user stories?

7) How did you trace the user stories with the test scenario,which tool is used?

8)entry criteria of SIT

9) How did you perform the UAT.

10)difference between Find Element Find Elements

11)How to perform Hover action

12)As you mentioned in Resume,Managing CICD using SauceLab to run more than 1000.If you have run 600 cases and some blocking issue occurs then what will you do?

13) Which plug in did you use in JIRA?

# **SHOULD KNOW**

1. Understand the process of testing in general and the process of testing we are following in Meta1 Project
2. Understand the different environments on which testing is performed
3. Understand the different types of testing in general and types of testing we are performing in Meta1 Project
4. Understand the levels of testing in general and the which all levels we are performing in Meta1 Project
5. Understand the process of logging defects, defect lifecycle and defect triage
6. Understand the test scenarios/ test cases in general and review the test cases already written in Meta1 Project for Lite wallet web, Android App, iOS App, Mobile Browser, Exchange UI, Explore UI
7. Understand how to use test data effectively to unleash hidden bugs, cover corner cases, and provide maximum coverage in testing
8. Understand different test design techniques in general
9. Understand SDLC in general
10. Understand Agile Project Management, Scrum, Sprints, Epic, User Stories, Story points etc.
11. Understand different meeting types in Agile like sprint planning meetings, backlog grooming, Sprint retrospectives, go-no go meetings etc.
12. Understand release process and where testing is initiated in the whole cycle
13. Understand Test Strategy, Test Planning, Entry and Exit criteria etc

**ANSWERS:**

**SELENIUM**

# **How a cronjob is set up for running the automated tests?**

Ans.

A cronjob is set up using the following syntax:

\* \* \* \* \* command to execute

Here the first \* indicates minutes of the hour (0-59)

the second \* indicates the hour of the day (0-23)

the third \* indicates the day of the month (1-31)

the fourth \* indicates the month of the year (1-12)

and the fifth \* indicates the day of the week(Sun – Sat as 0-7 where both 0 and 7 indicate Sunday)

We create a cron table or crontab using the command crontab -e. For eg. If we want to run the maven test class “FirstTest” at midnight every Monday, the we use the following command:

1. 0 \* \* 1 mvn test -Dtest=FirstTest

[Top](#_top)

# **Four major components of selenium.**

Ans.

Test Framework Host System

WebDriver

Browsers

Selenium Server or Grid

Drivers

The above diagram shows all the components of selenium when the Host System is remotely connected to the WebDriver through the Selenium Server or Grid. When the WebDriver, Drivers and Browsers are part of the same Host System, it is called direct communication. In such a case, Selenium Server or Grid is not part of the system.

Remote communication can be in two ways. It can either be through a Remote WebDriver or through Selenium Server or Grid. If communication is through the Remote WebDriver, then we have a WebDriver component outside the host system and a Remote WebDriver inside the host system with the Drivers and Browsers. While using the Selenium Server, both the WebDriver and the Selenium Server are outside the host system.

The Selenium Grid is used mainly for running tests in parallel in multiple machines. The WebDriver is initiated from the test framework like TestNG or Cucumber and communicates with the server or Grid which in turn connects to the drivers that run the browsers.

[Top](#_top)

# **How much time does it take to run the current project automated test scenarios.**

Ans.

The current project has around 350 test cases which takes around 4 hours to run.

[Top](#_top)

# **How to access browser?**

Ans.

**WebDriver driver;**

//For Chrome Drivers

**System.setProperty(“webdriver.chrome.driver”,”path to chrome driver”);**

**driver = new ChromeDriver();**

//For Firefox Drivers

**System.setProperty(“webdriver.gecko.driver”,”path to gecko driver”);**

**driver = new FirefoxDriver();**

//For IE Drivers

**System.setProperty(“webdriver.ie.driver”,”path to IE driver”);**

**driver = new InternetExplorerDriver();**

# **Why WebDriver driver=new FirefoxDriver () why not WebDriver driver= new WebDriver()?**

Ans.

WebDriver is an interface and not a class. FirefoxDriver() is the class that has implemented the interface WebDriver. The interface is a complete abstract class that groups methods but the methods have no body. The body of the methods in an interface are provided by the class that implements it using the keyword ‘implements’.

[Top](#_top)

**JAVA**

# **Explain abstraction**

Ans.

There are abstract classes and abstract methods. Abstract methods are those methods that are declared with an ‘abstract’ keyword and which has no implementation or body. If a class has an abstract method, then that class should also be declared as abstract.

Abstract classes are also declared with the keyword ‘abstract’ in the declaration. Abstract classes can have both methods with implementation and abstract methods. Abstract classes can also have fields that are not static or final and methods that are public, protected or private. Abstract classes cannot be instantiated but it can be inherited. The subclass has to implement all abstract methods of the abstract class it inherits or else the subclass itself has to be declared ‘abstract’.

[Top](#_top)

# **Difference between Interfaces and Abstract classes**

Ans.

Interfaces are similar to Abstract classes but interfaces are ‘implemented’ and not ‘extended’. Any class can implement multiple interfaces but can only extend one class, whether its abstract or not.

Interfaces can only have fields that are public, static and final and methods that are public. Abstract classes can have any type of fields or concrete methods.

Abstract classes are used by closely related classes wheras interfaces can be implemented by completely unrelated classes.

If any class implements an interface, it should implement all of the interface’s methods, if not, then that class should be declared ‘abstract’.

[Top](#_top)

# **What is a constructor?**

Ans.

A constructor is a special function or method in a class which has the same name as the class it belongs to and it has no return type. A constructor is usually called during runtime while creating a new instance of the class.

[Top](#_top)

# **Different types of constructors used in your project**

Ans.

The different types of constructors used in Java are:

* Default constructor
* Parameterized constructor
* No arg constructor
* Static constructor

Default constructors are created by the compiler when the programmer does not provide a constructor. A default constructor has no parameters.

Parameterized constructors are constructors with at least one parameter.

No arg constructors are similar to default constructors, except that it is created by the programmer.

Static constructors are those that use the object of the same class as a parameter.

[Top](#_top)

# **Difference between Overloading and Overriding**

Ans.

A method is said to be *overloading*, if there is another method with the same name but the input or output parameters of the method is different from each other.

For eg.

public int add(int a, int b);

public double add(double a, double b);

A method is said to be *overriding*, if there is another method with same name as well as same input and output parameters. This usually happens during inheritance when the child has the same function as the parent but different lines of code.

[Top](#_top)

# **Explain Thread.sleep, Implicitly wait(), Explicit wait() & Fluent wait**

Ans.

When loading a webpage, it takes time to load all the web elements based on external circumstances. ‘Waits’ help us to wait for the elements to load before continuing with our program.

Thread.sleep() is an unconditional wait. Thread is a class in java and sleep is a static method which waits fully for the specified time. It waits even if the element is found. This method is not recommended due to the same reason.

Implicit waits and explicit waits are conditional waits that wait for elements to load on a webpage.

Implicit wait –

* Implicit wait waits for a specified amount of time for the web elements to load before it throws an exception.
* Once an implicit wait is provided, it is applicable to all the ‘driver.findElement(s)’ methods in that program till the browser is closed.
* If the element is found before time, then we move to the next line of code.
* Polling is done every 250ms till the time expires to check for the web element. I
* If the element is not found before the specified time, the ‘NoSuchElementException’ exception is thrown.

**Syntax:**

|  |
| --- |
| ***driver.manage().timeouts().implicitlyWait(TimeOut, TimeUnit.SECONDS);*** |

Explicit wait –

* There are two types of Explicit waits. They are

a) WebDriverWait and

b) Fluent wait.

* WebDriverWait waits till a condition occurs before moving to the next line of code.
* It also sets a maximum time after which the exception is thrown.
* WebDriverWait is applicable only to a single element.

**Syntax:**

***WebDriverWait wait = new WebDriverWait (driver, 20);***

***wait.until(ExpectedConditions.VisibilityofElementLocated(By.xpath(""//button[@value='Save Changes']"")));***

Fluent wait –

* Fluent wait is similar to WebDriverWait except that it is more flexible in polling time and ignoring Exceptions.
* Fluent wait tries to find the web element repeatedly at regular intervals of time until the timeout or till the object gets found.

**Syntax:**

***Wait wait = new FluentWait(WebDriver reference)***

***.withTimeout(timeout, SECONDS)***

***.pollingEvery(timeout, SECONDS)***

***.ignoring(Exception.class);***

[Top](#_top)

# **Difference between CSS and xpath?**

Ans.

# **What is POM?**

Ans.

POM refers to Page Object Model. POM is a design pattern commonly used in Selenium for automating test cases. This pattern can be used with any Test Framework like Data Driven, Keyword Driven, Hybrid etc. Here web pages are represented by a corresponding class, the web elements are the variables of the class and the all interactions are provided by the methods of the class

[Top](#_top)

# **Difference between POM and page factory.**

Ans.

POM is a design pattern that organizes page objects into pages such that the pages and the test scripts can be easily differentiated. Whereas Page Factory is a class in Selenium that helps implement the Page Object Model.

In POM, the elements are located using the ‘By’ keyword whereas in Page Factory, elements are located using ‘@FindBy’ annotation.

POM does not handle exceptions well but Page Factory handles it well.

POM does not perform lazy initialization but Page Factory does.

POM uses a cache storage while performing tasks whereas Page Factory does not.

[Top](#_top)

**CUCUMBER**

# **Explain Hooks in cucumber.**

Ans.

Cucumber uses Feature files as the starting point of their framework. Each feature file can have any number of Scenarios. It is possible that these scenarios can have some common lines or blocks of code which need to be implemented before or after each scenario. Such lines of code can be grouped together and can form hooks. There are four types of hooks. @Before, @After, @BeforeStep and @AfterStep.

@Before hook contains blocks of code that should be executed before each Scenario is executed. @After hook contains code that should be executed after each Scenario. @BeforeStep is executed before each step in all scenarios and @AfterStep is executed after each step.

Hooks can also be combined with tags such that those blocks of code will be executed only for those Scenarios that have been similarly tagged.

Hooks are written in a separate class file and is not visible in the feature file.

[Top](#_top)

# **What is the difference between Background and Scenario**

Ans.

Scenario is a list of one or more steps in the Feature file in the Cucumber framework that represents a test case.

Background is a list of one or more steps in a Feature file that is common to all the Scenarios and which is to be executed before each Scenario.

[Top](#_top)

# **How do you pass data value input to a scenario in feature file**

Ans.

There are three ways in which to add data value input into a Scenario in a Feature file.

The first way is to directly hard code the data into the step of the Scenario. Changes are made to the step definition file to accept the data.

The second way is to change the keyword Scenario to Scenario Outline. The steps in the Scenario Outline and followed by the keyword Examples where the data table is provided that contains two or more sets of data with column headers. These headers are mentioned in the Scenario Outline steps where data needs to be sent.

The third way is to retain the Scenario keyword and provide the data as a table after each step that accepts the data.

[Top](#_top)

# **Important parameters of Cucumber Runner class**

Ans.

There are several parameters in the Cucumber Runner class mentioned as @CucumberOptions. Out of them, two are the most important and needs to be present in the file. They are ‘features’ and ‘glue’. The ‘features’ options gives us the location of the cucumber feature files that need to be run. The ‘glue’ option specifies the location of the step definition files.

There are other parameters like ‘tags’. The ‘tags’ option specifies the tags of the scenarios that need to be run. If no ‘tags’ are specified, it will run all the scenarios of the feature files mentioned in the ‘features’ option.

[Top](#_top)

**JAVA PROGRAMS**

# **Java Program to print word count**

Ans.

public class WordCountInString {

public static void main(String[] args) {

int count = 0;

String sentence;

int strLength;

Scanner scan = new Scanner(System.***in***);

System.***out***.println("Enter the string for counting words:");

sentence = scan.nextLine();

scan.close();

strLength = sentence.length();

for (int i = 0; i < strLength; i++) {

if ((sentence.charAt(i) == ' ') && (sentence.charAt(i + 1) != ' ')) {

count++;

}

}

if (strLength > 1) {

count++;

}

System.***out***.println("The number of words in the string is: " + count);

}

}

[Top](#_top)

# **Java program to print Alphabets A to Z**

Ans.

public class PrintAtoZ {

public static void main(String[] args) {

// Prints a to z

for (int i = 97; i < 123; i++) {

System.*out*.println((char) (i));

}

// Prints A to Z

for (int i = 65; i < 91; i++) {

System.*out*.println((char) (i));

}

}

}

[Top](#_top)

# **Java program to read from file**

package dataStructures.General;

import java.io.FileReader;

import java.io.IOException;

public class ReadFromFile {

public static void main(String[] args) throws IOException {

FileReader fRead = new FileReader("resources/permit.sh");

int i;

while ((i = fRead.read()) != -1) {

System.***out***.print((char) i);

}

fRead.close();

}

[Top](#_top)

# **Write a program to check the two strings are anagram without sorting**

Ans.

} /\* Check if two strings are anagrams.

That is, they contain the same characters in another order. \*/

public class StringAnagram {

public static void main(String[] args) {

Scanner scan = new Scanner(System.***in***);

String str1, str2;

char[] charStr1, charStr2;

Map<Character, Integer> hmapStr1 = new HashMap<Character, Integer>();

Map<Character, Integer> hmapStr2 = new HashMap<Character, Integer>();

System.***out***.println("Enter first string:");

str1 = scan.next();

System.***out***.println("Enter second string:");

str2 = scan.next();

scan.close();

if (str1.length() != str2.length()) {

System.***out***.println("The strings are not anagrams.");

return;

}

charStr1 = str1.toCharArray();

charStr2 = str2.toCharArray();

for (char c : charStr1) {

if (hmapStr1.containsKey(c)) {

hmapStr1.put(c, hmapStr1.get(c) + 1);

} else {

hmapStr1.put(c, 1);

}

}

for (char c : charStr2) {

if (hmapStr2.containsKey(c)) {

hmapStr2.put(c, hmapStr2.get(c) + 1);

} else {

hmapStr2.put(c, 1);

}

}

System.***out***.println(Arrays.*asList*(hmapStr1));

System.***out***.println(Arrays.*asList*(hmapStr2));

if (hmapStr1.entrySet().equals(hmapStr2.entrySet())) {

System.***out***.println("The strings are anagrams.");

} else {

System.***out***.println("The strings are not anagrams.");

}

[Top](#_top)

}

# **What is the StaleElementReferenceException? How to fix it?**

Ans.

A WebElement is an interface which means that when it is declared, a pointer or reference is created to an instance of the class it is instantiated with. StaleElementReferenceException occurs when the reference to an element that we have found is no longer fresh. Whenever a DOM is refreshed, it is rebuilt. Which means the reference to its elements is no longer valid. The DOM of a full page can be refreshed or part of the page can be refreshed by Ajax calls.

If our element has been refreshed or rebuilt, then any old reference to it becomes stale. We have to reference it again.

The best way to fix the issue is to declare and instantiate the web element in the same place that it is going to be used in, instead of in the beginning of the class.

[Top](#_top)

# **Explain BDD**

Ans.

* A BDD workflow starts when a Business Owner tells a Business Analyst or a Product Owner about his requiresments for the features he needs for his product or application.
* The Business Analyst documents these requirements and creates some User Stories and Examples.
* The Business Analyst then meets with the representatives of the Development Team and the Testing Team where he discusses the User Stories and examples. This meeting is called the Three Amigoes meeting.
* Anyone in the team can raise questions, offer opinions or make suggestions and come up with some agreed User Stories and Examples. This stage is called Discovery.
* In the next stage, these agreed User Stories and Examples are documented using a specific language called Gherkin, so that it can be understood by everyone from the Business Owner, to the last member of the Technical Team. This can also be used further on for development and automation. This phase is called the Formulation phase.
* The next stage is called Automation, where the automation test cases are created, and the product is developed based on the documentation created in Formulation stage.
* In the last stage, the test execution is done. If bugs are found, they are fixed and the retests or regression is done. Finally the product is released.

[Top](#_top)

# **Exceptions found during your project:**

* + - 1. NoSuchElementFound
      2. ElementNotVisibleException

**APPIUM**

# **Appium Architecture**

Appium has a client-server architecture.

*IOS Flow*

Developed by Facebook

WebDriver Agency

CLIENT

A

XCUITest Driver

App

B

APPIUM

SERVER

Bootstrap.js

UI Automation

Module

Device/Simulator

Deprecated

**CLIENT**-> In the IOS flow, we begin with the client which can be an IDE like eclipse that run test scripts that use the Appium client libraries to send HTTP requests to the Appium Server.

**A**-> HTTP requests are sent as JSON Wire Protocols or W3 WebDriver JSON.

**SERVER**-> The server is a node server ie. it runs on node.js.

**B**-> If the IOS version is 9.3 or above then the request is forwarded to the XCUITest driver. IOS version < 9.3 uses the UIAutomation module.

**XCUITest**-> This driver sends the request to the WebDriver Agency, developed by Facebook, which in turn processes the requests and work with the App on the physical IOS device or the simulator.

**UIAutomation**-> This module sends processes the requests and uses the bootstrap.js to implement them on the IOS device or simulator. This module is still available for use with older IOS version but mostly deprecated.

*ANDROID Flow*

CLIENT

Appium Android Bootstrap.jar

A

UIAutomator2

App

B

APPIUM

SERVER

UIAutomator

Deprecated

Device/Simulator

**CLIENT**-> In the IOS flow, we begin with the client which can be an IDE like eclipse that run test scripts that use the Appium client libraries to send HTTP requests to the Appium Server.

**A**-> HTTP requests are sent as JSON Wire Protocols or W3 WebDriver JSON.

**SERVER**-> The server is a node server ie. it runs on node.js.

**B**-> If the Android version is 5 and above, then the request is forwarded to the UIAutomator2. Android version < 5 uses the UIAutomator. Both drivers are developed by Google.

**UIAutomator2**-> This driver uses the appium android bootstrap.jar to process the requests and work with the App on the physical Android device or the emulator.

**UIAutomator**-> This module has deprecated but is used with older android versions. This module processes the requests and implement them on the Android device or emulator.

**APPIUM ECOSYSTEM**

TIZEN driver

SAMSUNG Apps

WebDriver Agency

XCUITest driver

Test Scripts

IOS Apps

UIAutomation

Appium Client Libraries

ESPRESSO modules

ESPRESSO server

ESPRESSO driver

ANDROID Apps

APPIUM SERVER

UIAutomator2

WINDOW Apps

WIN APP driver

WINDOW driver

MAC Apps

APPIUM MAC

MAC driver

YOU I enabled Apps

YOU I driver

[Top](#_top)

# 

# **What is adb ? how do you use adb in Appium?**

Ans.

Android Debug Bridge(adb) is a command line tool that can connect an Android device to a computer to perform tasks that require root. For normal users to perform system-protected tasks like removing unwanted pre-installed software they need advanced techniques like altering the operating system files through rooting.

adb reboot,

adb install <path>,

adb shell dumpsys iphonesysinfo

(adb shell "service call iphonesubinfo 4 | cut -c 52-66 | tr -d '.[:space:]'")

adb shell “cmd package list packages -3”

adb reboot recovery

**Adb script to grant access permissions to media files.**

adb shell

pm grant com.liqvd.digibox.test android.permission.READ\_EXTERNAL\_STORAGE

pm grant com.liqvd.digibox.test android.permission.WRITE\_EXTERNAL\_STORAGE

[Top](#_top)

# **How to switch apps during the execution of Mobile Automation (while in contacts app, switch to camera app)**

Ans.

We can use the following code:

// launch settings App

        Activity activity = **new** Activity(settingsAppPackageName, settingsAppActivityName);

        activity.setStopApp(**false**);

        ((AndroidDriver<MobileElement>) driver).startActivity(activity);

[Top](#_top)

# **How to scroll in mobile automation**

Ans.

TouchActions action = **new** TouchActions(driver);

action.scroll(element, 10, 100);

action.perform();

[Top](#_top)

# **What are the challenges with mobile automation?**

Ans.

Main challenges faced by a mobile app automator are

* Too many devices – different types of devices in this world
* Device fragmentation – no. of devices running an app at the same time is huge.
* Different screen sizes
* Different types of mobile apps – native, web and hybrid
* Changing user expectations
* Security issue with cache and encryption
* Heavy battery usage
* Too many app testing tools.
* Flakiness
* Synchronisation issues
* App state
* Resource hungry

[Top](#_top)

**MAVEN**

# **What is Maven?**

Ans.

Maven is a build automation tool only for Java. It is used to create a build in the form of jar or war file. It executes your code, compile code and generate artifacts in the form of jar or war file. Similar build automation tools are Ant and Gradle.

[Top](#_top)

**API QUESTIONS**

# **What are web services?**

Ans.

A Web Service is a collection of open protocols and standards which are used for communication between systems or applications. Web services are API’s that are communicated over the internet through HTTP protocols.

# **What is API Testing?**

Ans.

API testing is a type of software testing which determines if the developed API meets expectations regarding the functionality, reliability, performance and security of the application.

[Top](#_top)

# **How to perform API testing and what should be the standard process to perform the same?**

Ans.

# **How to validate the error logs of API?**

Ans.

# **Comparison between SOAP and REST APIs**

Ans.

SOAP – Simple Object Access Protocol

REST – REpresentational State Transfer

* Soap is a protocol while Rest is an architectural concept.
* Soap cannot use Rest, since it’s a protocol but Rest can use Soap.
* Soap uses service interfaces to expose business logic whereas Rest uses URI.
* JAX-WS is the security Java API for SOAP web services whereas JAX-RS is the security Java API for RESTful web services.
* SOAP defines standards to be strictly followed whereas REST does not define too many standards.
* SOAP requires more bandwidth and resources than REST.
* SOAP defines its own security whereas RESTful web services inherit the security measures from the underlying transport.
* SOAP permits XML format only whereas REST permits different data formats like plain text, HTML,XML,JSON etc.
* SOAP is less preferred to REST.

[Top](#_top)

# **Difference between REST API and Restful APIs**

Ans.

REST API is a set of constraints or an architectural design.

RESTful API are APIs that follow those constraints.

[Top](#_top)

# **About different Request Methods of APIs**

Ans.

# **How are we testing the APIs?**

Ans.

# **Scenarios on example API**

Ans.

# **How you test API in postman? Please explain.**

Ans.

# **How to test 1000 load on API, way of putting loads**

Ans.

# **How to check when an API crashes?**

Ans.

# **Tools for API testing?**

Ans.

# **What do you verify in the API response?**

Ans.

# **Types of API response codes**

Ans.

The most used API status response codes are:

100 Series – Informational.

200 Series – Successful response

300 Series – Response related to URL relocation

400 Series – Error on client side.

500 Series – Error on server side

200 – Ok

201 – Created successfully

204 - Deleted successfully

301 – Relocated successfully

400 – Bad request (missing something or error in request payload)

401 – Not authorized

402 – Payment required

403 – Server refuses request

404 – Not found

405 – Method not allowed

500 – Internal Error/Server issues

[Top](#_top)

# **Issues faced while API testing?**

Ans.

# **How to use SWAGGER? Can we perform API testing from same?**

Ans.

# **What is environment in postman?**

Ans.

# **Tools/software's to be installed/required before starting Mobile Automation**

Ans.

The following tools are required before we start Mobile Automation.

* We need a test script. An IDE like Eclipse to run it.
* We need Appium Server.
* We need an emulator or a real connected device.

**URL(Endpoint URL) + (API)Service URL = URI**

**? Query parameter** – filtered result

**/ Path parameter** – specific result

[Top](#_top)

**TESTNG**

# **What is TestNG?**

Ans.

TestNG is an open-source automation testing framework that is inspired by JUnit which uses annotations. TestNG overcomes the disadvantages of JUnit and is used for all kinds of testing like unit testing, integration testing etc.

There are three major advantages of TestNG over JUnit:

* Annotations are easier to understand
* Test cases can be grouped more easily
* Parallel testing is possible

Other features of TestNG are:

* XML based test configuration
* Reporting
* Data driven testing
* Dependent groups and methods
* Listeners
* Open API

[Top](#_top)

# **What are different annotations used in TestNG?**

Ans.

Annotations are lines of code written with an ‘@’ prefix that controls the execution of the methods below it. The different annotations in TestNG are:

* **@BeforeSuite**: The annotated method will be run before all tests in this suite have run.
* **@AfterSuite**: The annotated method will be run after all tests in this suite have run.
* **@BeforeTest**: The annotated method will be run before any test method belonging to the classes inside the tag is run.
* **@AfterTest**: The annotated method will be run after all the test methods belonging to the classes inside the tag have run.
* **@BeforeGroups**: The list of groups that this configuration method will run before. This method is guaranteed to run shortly before the first test method that belongs to any of these groups is invoked.
* **@AfterGroups**: The list of groups that this configuration method will run after. This method is guaranteed to run shortly after the last test method that belongs to any of these groups is invoked.
* **@BeforeClass**: The annotated method will be run before the first test method in the current class is invoked.
* **@AfterClass**: The annotated method will be run after all the test methods in the current class have been run.
* **@BeforeMethod**: The annotated method will be run before each test method.
* **@AfterMethod**: The annotated method will be run after each test method.
* **@Test**: The annotated method is a part of a test case.

[Top](#_top)

# **TestNG Listeners**

Ans.

**MANUAL TESTING**

# **Difference between Error, Defect and Bug**

Ans.

An error is a mistake made during coding. When a developer cannot successfully compile or run a program due to a coding mistake, it is called an error.

A defect is when the application does not work as expected. This is found by the programmer inside the code during development.

A bug is a defect that is found by the testing team that is reproduced and recorded.

[Top](#_top)

# **What is smoke, sanity and Regression testing?**

Ans.

Smoke Testing –

* This is performed to cover critical functionalities of the application.
* Its objective is to verify stability of the product before further tests are done on the product. It checks that the product is good enough to run the first test case.
* They are part of regression tests.
* Failing in these tests result in instant rejection of the product.
* Also known as Build Verification tests.
* These tests are performed by both QA and developers.

Sanity Testing –

* This is performed to check if new module additions are stable enough to pass to the next level of testing.
* Its objective is to verify stability of new functionality or changes made to the build.
* They are also part of the regression tests.
* These tests check modules where changes have been made.
* Also known as Surface Level Testing.
* These tests are only performed by QA.

Regression Testing –

* This is performed to check the detailed functionality of the product.
* These tests check whether changes in one module has affected all other modules of the product.
* These tests need to be well documented.
* Smoke and Sanity are part of Regression Testing.
* These tests are only done by the QA team and mostly by automation.

Note: Smoke and Sanity can have some common test cases.

[Top](#_top)

# **How much time do you take to perform smoke, sanity and regression testing?**

Ans.

[Top](#_top)

# **Integration Testing**

Ans.

Integration Testing is a type of testing where software modules are integrated logically and tested together as a group. The purpose of this testing is to detect defects in the interaction between these modules. It focuses on checking data communication between these modules.

[Top](#_top)

# **What is the needs of automation when we are doing the integration testing?**

Ans.

[Top](#_top)

# **How do we do integration testing?**

Ans.

[Top](#_top)

# **What is the defect life cycle?**

Ans.

A defect life cycle is a set of states that a defect goes through from the time its raised to the time it’s completely closed. The following are the different states:

* New
* Assigned
* Duplicate, Deferred, Rejected, Not a Bug
* Open / In Progress
* Fixed
* Pending Retest
* Retest
* Reopened
* Verified
* Closed

The defect life cycle explained is as below:

* Tester finds and raises the bug. Status of bug is now ‘New’.
* The defect is forwarded to the Project Manager for analysis.
* If the Project Manager finds the bug invalid, he changes the status to ‘Rejected’.
* If not rejected, he checks the scope. If it’s not within the functionality or scope, he changes status to ‘Deferred’.
* If the bug is similar to another one raised earlier, he changes the status to ‘Duplicate’.
* If the defect state is still ‘New’, then the defect is assigned to a developer. The status changes to ‘Assigned’.
* When developer starts working on the defect the status changes to ‘In Progress’.
* Once the code is fixed, status changes to ‘Fixed’.
* The tester will retest the code, if the test passes, status changes to ‘Closed’.
* If test fails, the status changes to ‘Reopened’ and is assigned to the developer.
* In cases where the defect is seen in future builds, a closed defect can also be reopened.

[Top](#_top)

# **What time does your automation take to run 1000 test scenarios? Can you make it faster?**

Ans.

1000 test cases take around 10hrs to execute. Few of the ways to speed up automation of test cases are given below:

* Parallel Testing – test cases are split among devices
* Smaller test cases – less than 20 steps per test case
* Refining test scripts - no duplicates and only relevant methods are called.
* Using Explicit waits – eliminates waste of time.
* Avoid x-paths where possible

[Top](#_top)

**VERSION CONTROL**

# **Git**

Ans.

Git is a distributed version control system which lets you store files in a repository and as you make changes to the file, we can save it each time such that it is easy to track each version of each file of a project. It is also easy to revert back to a previous version in case of errors.

[Top](#_top)

# **Merge Conflict**

Merge commit occurs in a team environment when two commits edit the same line of code or if one commit deletes a file that was edited by another commit. To solve a conflict

* First identify the file and location of conflict.
* Make the necessary changes to resolve the conflict.
* Add the file to the staging area using ‘git add’ command.
* Then commit the file using the ‘git commit’ command.
* Now try to merge again.

[Top](#_top)

**COLLECTIONS**

# **How hashmap works (internally)**

Ans.

* HashMap uses its static inner class Node<K,V> for storing map entries.
* HashMap uses multiple buckets and each bucket points to a Singly Linked List where the entries are stored
* HashMap uses a hashCode of the key Object which is used by the hash function to find the index of the bucket where the new entry is to be added.
* Once the bucket is identified, the hashCode is used to check if the key with the same hashCode is present in the bucket.
* If the key exists then the equals() method is used on the key. If the method returns true, then the value against that key is overwritten in the node else a new node is created and added to this Singly Linked List of that bucket.
* If there is no such key with the same hashCode in the bucket then a new node is created and added into the bucket.
* If the number of nodes in a bucket exceeds 8 then the Singly Linked List changes into a self-balancing Binary Search Tree(BST).

[Top](#_top)

**Answers to Riddles.**

# **Bridge Riddle**

First A & B cross over, then A returns. Next C and D cross over. B returns. The A and B cross again.

# **Wrongly labelled Box**

Open any box. If it has for eg. Red balls but labelled blue, we know that the box labelled red does not have red balls or blue balls. Otherwise, the third box will be labelled right. So, the box labelled blue should be labelled Red, box labelled Red should be labelled Mix and the box labelled Mix should be labelled Blue.

[Top](#_top)

**Should Know**

# **Understand different test design techniques in general**

Ans.

There are a lot of techniques but the following five are the most popular.

* Equivalent class partitioning
* Boundary value analysis
* State transition
* Pairwise testing
* Error guessing

Equivalent Class Partioning -