API Questions

Notebook: First Notebook

1. What is an API?

API is an acronym and it stands for **A**pplication **P**rogramming **I**nterface. API is a set of routines, protocols, and tools for building Software Applications. APIs specify how one software program should interact with other software program.

In simple words, API stands for **A**pplication **P**rogramming **I**nterface. API acts as an interface between two software applications and allows the two software applications to communicate with each other. API is a collection of software functions which can be executed by another software program.

2. What is API Testing?

API testing is a type of software testing that involves testing APIs directly and also as a part of integration testing to check whether the API meets expectations in terms of functionality, reliability, performance, and security of an application. In API Testing our main focus will be on Business logic layer of the software architecture. API testing can be performed on any software system which contains multiple APIs.

3. What are the common API Testing Types?

API testing typically involves the following practices:

- Unit testing
- Functional testing
- Load testing
- Runtime/ Error Detection
- Security testing
- UI testing
- Interoperability and WS Compliance testing
- Penetration testing
- Fuzz testing

4. Name some of the common protocols used in API Testing?

Some of the protocols using in API Testing are as follows:

- HTTP
- REST
- SOAP
- JMS
- UDDI

5. Difference between API and Web services?

Web services:

1. All web services are APIs
2. All web services need to be exposed over web(HTTP)
3. A Web service uses only three styles of use: SOAP, REST and XML-RPC for communication

4. A Web service always needs a network to operate

APIs:

1. All APIs are not web services
2. All APIs need not be exposed over web(i.e. HTTP)
3. API uses multiple ways for communication e.g. DLL files in C/C++, Jar files/ RMI in java, Interrupts in Linux kernel API etc.

4. APIs don't need a network for operation

6. What is Soap?

SOAP stands for Simple Object Access Protocol. It is an XML based messaging protocol. It helps in exchanging information among computers.

7. What is Rest API?

REST stands for Representational State Transfer. It is a set of functions helping developers in performing requests and receive responses. Interaction is made through HTTP Protocol in REST API.

8. Difference between SOAP and REST?

SOAP:
1. SOAP is a protocol through which two computers communicate by sharing XML document
2. SOAP supports only XML format
3. SOAP does not support caching
4. SOAP is slower than REST
5. SOAP is like custom desktop application, closely conneccted to the server

6. SOAP runs on HTTP but envelopes the message

REST:
1. REST is a service architecture and design for network-based software architecture
2. REST supports different data formats
3. REST supports caching
4. REST is faster than SOAP

5. REST client is just like a browser and uses standard methods An application has to fit inside it

6. REST uses the HTTP headers to hold meta information

9. What are the common tests that are performed on APIs?

Some of the common tests we perform on APIs are as follows.

1. Verify whether the return value is based on input condition. Response of the APIs should be verified based on the request.
2. Verify whether the system is authenticating the outcome when the API is updating any data structure
3. Verify whether the API triggers some other event or request another API

4. Verify the behaviour of the API when there is no return value

10. What are the advantages of API Testing?

- API Testing is time effective when compared to GUI Testing. API test automation requires less code so it can provide faster and better test coverage.
- API Testing helps us to reduce the testing cost. With API Testing we can find minor bugs before the GUI Testing. These minor bugs will become bigger during the GUI Testing. So finding those bugs in the API Testing will be cost effective to the Company.
- API Testing is language independent.
- API Testing is quite helpful in testing Core Functionality. We can test the APIs without a user interface. In GUI Testing, we need to wait until the application is available to test the core functionalities.
- API Testing helps us to reduce the risks.

11. What exactly needs to be verified in API Testing?

Basically, on API Testing, we send a request to the API with the known data and we analyse the response.
1. Data accuracy
2. HTTP status codes
3. Response time
3. Error codes in case API returns any errors
4. Authorization checks

5. Non functional testing such as performance testing, security testing

12. Name some tools used for API Testing?

Some of the tools used for API Testing are as follows:

- Postman
- Katalon Studio
- SoapUI
- Assertible
- Tricentis Tosca

- Apigee
- JMeter
- Rest-Assured
- Karate DSL
- API Fortress
- Parasoft
- HP QTP(UFT)
- vREST
- Airborne
- API Science
- APIary Inspector
- Citrus Framework
- Hippie-Swagger
- HttpMaster Express
- Mockbin
- Ping API
- Pyresttest
- Rest Console
- RoboHydra Server
- SOAP Sonar
- Unirest
- WebInject

> *Learn more on API Testing Tools*

13. List some most used templates for API documentation?

Some of the API documentation templates are as follows.

- Swagger
- FlatDoc
- RestDoc
- API blueprint
- Slate
- Miredot
- Web service API Specification.

14. Name some of the API examples which are quite popular.

Some of the popular API examples are

- Google Maps API

- YouTube
- Twitter
- Amazon Advertising API

15. Difference between API testing and Unit Testing?

UNIT TESTING:

- Unit testing is conducted by Development Team
- Unit testing is a form of White box testing
- Unit testing is conducted prior to the process of including the code in the build
- Source code is involved in Unit testing
- In unit testing, the scope of testing is limited, so only basic functionalities are considered for testing

API TESTING:

- API testing is conducted by QA Team
- API testing is a form of Black box testing
- API testing is conducted after the build is ready for testing
- Source code is not involved in API testing
- In API testing, the scope of testing is wide, so all the issues that are functional are considered for testing

16. What are the main challenges faced in API testing?

Some of the challenges we face while doing API testing are as follows

- Selecting proper parameters and its combinations
- Categorizing the parameters properly
- Proper call sequencing is required as this may lead to inadequate coverage in testing
- Verifying and validating the output
- Due to absence of GUI it is quite difficult to provide input values

17. What are the types of bugs we face when performing API testing?

Issues observed when performing API testing are

- Stress, performance, and security issues
- Duplicate or missing functionality
- Reliability issues
- Improper messaging
- Incompatible error handling mechanism
- Multi-threaded issues

- Improper errors

18. How is UI testing is not similar to API testing?

UI (User Interface) testing is to test the graphical interface part of the application. Its main focus is to test the look and feel of an application. On the other hand, API testing enables communication between two different software systems. Its main focus is in business layer of the application.

19. Name some most commonly used HTTP methods?

Some of the HTTP methods are

GET: It enables you to retrieve data from a server
POST: It enables you to add data to an existing file or resource in a server
PUT: It lets you replace an existing file or resource in a server
DELETE: It lets you delete data from a server
PATCH: It is used to apply partial modifications to a resource
OPTIONS: It is used to describe the communication options for the target resource

HEAD: It asks for a response identical to that of a GET request, but without the response body

20. Can you use GET request instead of PUT to create a resource?

No, GET request only allows read only rights. It enables you to retrieve data from a server but not create a resource. PUT or POST methods should be used to create a resource.

21. What is the difference between PUT and POST methods?

PUT and POST methods are sometimes confused in regards to when each should be used. Using POST request, our intent is to create a new object on the server whereas with PUT request, our intent is to replace an object by another object.

POST should be used when the client sends the page to the server and then the server lets the client know where it put it. PUT should be used when the client specifies the location of the page

Postman Interview Questions

What is Postman?

Postman is a rest client software that started as an chrome extension but is now available as native application only. Postman is basically used for API testing in which you can test your APIs with different types of request method types like post, put etc and parameters, headers and cookies. Apart from setting the query parameters and checking the response, postman also let us see different response stats like time, status, headers, cookies etc. with some extra excellent features that can be used with ease.

What is an API?

API stands for ***Application Programming Interface***. Talking in technical terms an API is a set of procedures, functions, and other points of access which an application, an operating system, a library etc., makes available to programmers in order to allow it to interact with other software. It can be considered as the waiter which acts as the middleman between your requests and the chef. Similarly API refers as the middleman between a client and a server.

What are some tools used for API Testing?

There are many API testing tools. The following six are the top most according to the users/downloads. These are not the rankings though.

- ***Postman***
- ***SoapUI***
- ***Katalon Studio***
- ***Tricentis Tosca***
- ***Apigee***
- ***Jmeter***

In which type of encoding does postman accept authorization credentials? Why?

Postman accept authorization in Base64 encoding only. This is provided inbuilt in Postman or else you can also refer third party websites to convert the credentials in base64. We use base64 particularly because it transmits the data into textual form and send it in easier form such as HTML form data**.** We use Base64 particularly because we can rely on the same 64 characters in any encoding language that we use. (Refer ***tutorial***)

What is meant by the term Environment with respect to Postman?

An environment in Postman is a set of key-value pairs. You can create multiple env in postman which can be switched quickly with a press of a button. There are two types of environment, global and local. They define the scope of the variable to use it in the requests. Most commonly the variable is defined inside the environment for the same purpose. The most common variable we use is url because url is used in every requests and changing it can be very time consuming. When we create an environment inside Postman, we can change the value of the key value pairs and the changes are reflected in our requests. An environment just provides boundaries to variables. (Refer ***tutorial***)

Can we have two global scope variables with the same name in Postman?

Since global variables are global i.e. without any environment, they cannot have duplicate names as it creates confusion for the software. Local variables can have same name but in different environments. (Refer ***tutorial***)

Which one has the higher priority in Postman? A global variable or a local variable?

In Postman, if two variables have same name (one being local and one being global) then the higher priority is of the local variable. It will overwrite the global variable. (Refer ***tutorial***)

Define Team workspace in Postman

A workspace is a collaborative environment for a group of users to develop and test APIs. A team workspace is a workspace which is shared by the whole team working on same collections of requests. Since it is very time consuming and hard to share the collections through external drives or other sharing means, team workspace synchronises and collaborates all the team's work at one place. (Refer ***tutorial***)

Explain the following piece of test code in Postman

tests["Status Code is 200"] = responseCode.code === 200

- ***tests*** *– variable on one of type array*
- ***Status code is 200*** *– A string or the test name which will be represented in the test result box so that we can know what test was it. It is important as we use many tests on one request.*
- ***responseCode.code*** *= responseCode is used to save all the response that we get from the server. Since we do not need complete response, we need to create one object to extract our required info. code object is then called to output the status code (like 200) from the entire response which we have saved.* (Refer ***tutorial***)

What is the difference between Postman Monitors and Postman Collection Runner?

The postman Monitor is an automated way of running collections. Collections are triggered automatically as per specified parameters whereas Postman Collections require some manual effort to start and monitor the execution. A postman collection runner runs the collection for the iterations you want to. It will be stopped when you stop the software and is not automated. A postman monitor will run your collection at regular user defined intervals till the time you have specified. Your collection will be run even if your system has

shut down as it is connected through postman cloud.
(Refer ***tutorial***)

Can we import local variables in Postman Monitors?

Yes. Postman monitors allows to import local variables but it does not allow to import global variables. I believe it can be imported in json, if yes, pls mention.(Refer ***tutorial***)

Describe any four response things you receive from a response (Correct or Incorrect)

- ***Status Code***
- ***Response Status***
- ***Response time***
- ***Response Size***
- ***Response Headers***
- ***Response Cookies***
- ***Response Date and Time***
- ***Response Session limit***
- ***Response Cookies***
- ***Response Server***
- ***Response type***

(Refer ***Tutorial***)

What is a collection in Postman?

A collection in Postman can be imagined similar to a folder in your system. A collection is the grouping of requests, preferably of the similar types. It can be compared similar to the folder inside your system which has one type of files. It is one of the most vital feature of Postman and it also provides nice features such as running a whole group of request together with just one click. A collection is

also important for sharing many requests at once and contains many more features which can be referred by the given link. (Refer ***tutorial***)

Should we save our work in Postman cloud if we are working in a company? Why?

A Postman cloud is Postman company's repository like Microsoft has One Drive etc. In Postman cloud you can save your work instantly after logging in and also retrieve it from anywhere you want. It is not preferred to save your work in Postman cloud as company's work is often confidential and should not be leaked out. Postman cloud needs signing in and hence security can be compromised, therefore team workspace is preferred instead of saving the work in Postman Cloud. (Refer ***tutorial***)

State any 5 types of Request Method types.

- ***Get***
- ***Post***
- ***Put***
- ***Delete***
- ***Patch***
- ***Head***
- ***Delete***

(Refer ***Tutorial***)

Please define status code 401. Also, a situation in which we can incur such status code.

Status code 401 is referred for an unauthorized request. An unauthorized request is a request for which you are not authorized.

We can incur such a status code when you are not authorized to access the server or you have entered wrong credentials.

Other status codes which are seen commonly are

- ***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.*
- ***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.*
- ***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.*
- ***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.*

(Refer ***tutorial***)

What are different types by which we can see response body in Postman. Explain.

In Postman, a response body can be seen by three different types

- ***Pretty***
- ***Raw***
- ***Preview***

Although all the three have their own importance and value in Postman, the most commonly used is Pretty as it shows the response code in different format and colours which is easy to read and analyse the response. It is just like any good text editor used for coding. (Refer ***tutorial***)

What is "Bulk Edit" feature of Postman used for?

Bulk Edit feature of Postman is used for the convenience of adding parameters to a new request from the previous request. Since a request can have many parameters and it is very difficult to copy and paste one by one, bulk edit feature helps us copy all the keys and their respective values at once and paste them. (Refer ***tutorial***)

What is "x-www-urlencoded" in Post method in Postman?

Form data and x-www-form-urlencoded are very similar. They both are used for almost the same purposes. But the difference between the form data and x-www-form-urlencoded is that the url will be encoded when sent through x-www-form-urlencoded. Encoded means the data which is sent will be encoded to different characters so that it is unrecognizable even if it is under attack.
(Refer ***tutorial***)

What is binary in Post method in Postman?

Binary form in Postman is designed to send the information in a format that cannot be entered manually. Since everything in a computer is converted to binary, we use these options which cannot be written manually such as an image, a file etc. (Refer ***tutorial***)

What is Pre-Request Script in Postman?

A pre request script is a script that runs before the execution of a request. (Refer ***tutorial***)

What is the difference between authorization and authentication?

Authentication is a process of presenting your credentials to the system and the system validating your credentials. These credentials tell the system about who you are. Authorization is a process of allowing or denying someone from accessing something, once authentication is done. (Refer ***tutorial***)

What are the different scopes of an environment variable in Postman?

A scope of a variable is defined as the boundaries through which it can accessed. They are

- ***Local Scope*** *: Can be accessed only in the environment in which it was created*
- ***Global Scope****: Can be accessed globally in any environment or no environment.*

(Refer ***tutorial***)

Why do we group requests under collections when collection is already a grouping of requests?

A collection may have hundreds of requests under it. We need to sub categorize the requests according to a more specific category so that it is easier for us to find them, edit them or modify them. For this we use folders in collections. A collection may have many folders inside it and a folder may have many requests. This way we can generalise the types of requests to a deeper level than the collections which is already generalised. *For easiness, a collection can be considered a folder "Movies" in your system which has all the movies. A folder can be considered as different folders inside "Movies" like Hollywood, Bollywood etc which have respective types of movies.*

(Refer ***tutorial***)

What are the two ways in which tests can be written in Postman?

In postman we can write tests in either Javascript method or Functional method. Although functional method also uses javascript but the syntax is different. Functional method is officially recommended and used method in Postman. It can also be noticed that all the snippets inside Postman are in functional methods only. It also has an inbuilt library which is Chai. Chai also uses functional method in a very beautiful way to make it more readable and shorter.(Refer ***tutorial***)

Which method should you prefer? Javascript or Functional to write the tests?

It is advised and recommended to use the functional method while writing tests in Postman. Although there have been no notice of ending the support for JS method. (Refer ***tutorial***)

Write a test code to check whether the response status is 200 or not.

A test code to check whether the response status is 200 or not is as follows

tests["Status Code is 200"] = responseCode.code === 200;

What is the need to Monitor the collections in Postman?

It is very important that your API's responses and performance remain up to the mark throughout the day. Monitors can help you schedule a collection of test runs to monitor the performance and response of your APIs een if you are not available or not handling them. (Refer ***tutorial***)

Can we run monitors in Postman without Signing in?

No, monitors cannot be run without signing in because monitors run your collection even if your system is shut down. So, you need a place to store the collection and let it run automatically. You also need a place to store the reports so that you can look at them when you are free. This all needs to be saved into your postman account and hence you need to sign in. (Refer ***tutorial***)

What is the importance of setNextRequest in Postman?

setNextRequest in Postman is used to define the workflow. setNextRequest is needed to change the order of the requests being executed. (Refer ***tutorial***)

What are the two types of scripts in Postman?

We can write two types of script in Postman

- ***Tests script***
- ***Pre-request script***

(Refer ***tutorial***)

What is Chai Assertion Library?

Chai assertion library is an assertion library which is installed beforehand to use in Postman. This is used to write assertions in Postman which are very beneficial. Chai assertion helps us write

many lines of test code in a few lines which is both understandable and readable. Chai uses BDD approach which means that chai library has codes that are more user friendly.

A simple code written in chai library which tests if number 3 is already in array or not.

```
pm.test("Number included", function(){

pm.expect([1,2,3]).to.include(3);


});
```

(Refer tutorial)

What command line interface is used with Postman normally to serve continuous integration.

Newman is used with Postman normally as a command line interface to serve continuous integration. (Refer ***tutorial***)

Write the command for running a folder in Newman.

In Newman it is not necessary to run the complete collection to check just a bunch of request. This is obviously time consuming and not recommended. We can also run just a folder located inside a collection in the Newman. For running a folder in Newman, the following command is used

newman run <collection_name> –folder <folder name>

What is Jenkins?

Jenkins is used to build and test your project continuously and hence making the work of a developer and a tester easy for the software. Jenkins uses continuous integration and continuous development for the development and deployment of the software.

In what language is Jenkins written?

Jenkins is an open source automation server written in Java. (Refer ***tutorial***)

What do you understand by continuous delivery?

Continuous delivery works as a next step of continuous integration. Continuous delivery is a DevOps software development practice where code changes are automatically built, tested (Unit Tests), and prepared for a release to an environment. (Refer ***tutorial***)

What is the main difference between continuous delivery and continuous deployment?

The main difference between ***continuous delivery*** and ***continuous deployment*** is the presence of a manual approval to update to production. With continuous deployment, production deployment happens automatically without explicit approval. (Refer ***tutorial***)

State any 2 advantages of Jenkins.

- *It has huge plugin support.*

- *It is built in Java and hence, platform independent.*
- *It is an open source tool with great community support*
- *It is easy to install.*
- *It is free of cost.*

(Refer ***tutorial***)

Why is freestyle job the most preferred job for developers in Jenkins?

A freestyle project is a project in which you can run any types of build. This enables the developers to develop huge number of plugins for this type of projects as it gives flexibility. Also, it let us develop any type of build so it becomes a primitive choice.
(Refer ***tutorial***)

Why does plugins not come pre installed in Jenkins?

Plugins are used for extending the features of a software. Since these plugins can be of considerable sizes, it increases the overall size of the software. Moreover, it is quite obvious no one developer will be using all the features in his software. Therefore, plugins do not come pre installed and the developers install themselves whichever plugin they need.

What is the difference in Post-Build and Build with respect to Jenkins?

Build in Jenkins accepts the batch commands that are needed to be executed. For example ***newman run <link>*** is a batch command needed to run a collection through Newman and hence is written in build section.

Post-Build section is used to tell jenkins if there is anything that is needed to be done after the batch commands have been executed. These can include publishing the reports as we publish the report

after the commands have executed and tests are run etc. (Refer ***tutorial***)

What flag is used to remove unicode from the Jenkins console output?

A flag is used in Jenkins to make some changes in the output console. A flag also provide option so that the complete response can be filtered out. For removing the unicode we use ***–disable-unicode*** flag. (Refer ***tutorial***)

Why do we require Reports in Jenkins?

A report in Jenkins is a structured and graphical way where we can see the execution results or test output. Reports also work easier when we have to communicate the results with our team members or with other stakeholders. It is always convenient to see the complete track of your APIs through a report than comparing the console outputs. (Refer ***tutorial***)

What is the command to generate an junit xml report in Jenkins?

To generate a junit xml report we use ***–reporters junit*** command. (Refer ***tutorial***)

What is the difference between –reporters and –reporter flags in Jenkins?

–reporters is used to tell jenkins that we need to publish a report. We then need to specify what type of report we want.

–reporter is used to as an option for additional operations on reports. This is defined by the flags we use after this. For example –reporter-export will export the report to the given location.
(Refer ***tutorial***)

SDET: Rest Assured Interview Questions

- How will you write a simple Rest Assured testcase that validates sample json response?
- API returns a simple integer array [1, 2, 3], how to test such a response using Rest Assured?
- How to get all books with price less than 10 from json response?
- How to treat float and doubles as BigDecimal in json input
- How to pass given parameters and headers to API call
- How to set path params in API calls
- How to test against an expected response code from API call
- How to extract API responses after validation
- How will you extract custom node from Json response using JsonPath?
- How to set cookies in API calls?
- How to set a JSON body for a POST API call?
- How to verify cookie, status, headers & content-type in API response
- How to measure and validate API response time?
- How to test REST API that requires basic authentication
- How to test REST API that requires digest authentication
- How to test REST API that has CSRF enabled
- How to test REST API that requires OAuth1 authentication
- How to test OAuth2 protected resource using rest assured

- Testing file upload in Rest Assured
- How to handle object mapping to JSON - serialization and deserialization
- You have a fleet of APIs that requires n number of custom headers, how can you avoid duplicating the code for passing the same headers again and again?
- How to handle session in test?
- How to relax HTTP validation for invalid certificates? This may be the case for dev environment.
- how to handle parameters that are already URL encoded?

What Is Rest?

Answer: REST is web standards-based architecture and uses HTTP Protocol for data communication. It revolves around resources where every component is a resource and a resource is accessed by a common interface using HTTP standard methods. REST was first introduced by Roy Fielding in 2000.

In REST architecture, a REST Server simply provides access to resources and REST client accesses and presents the resources. Here each resource is identified by URIs/ global IDs. REST uses various representations to represent a resource like text, JSON and XML. Nowadays JSON is the most popular format being used in web services.

Name Some Of The Commonly Used Http Methods Used In Rest Based Architecture?

Answer: Following well-known HTTP methods are commonly used in REST-based architecture:

- GET − Provides read-only access to a resource.
- PUT − Used to create a new resource.
- DELETE − Used to remove a resource.
- POST − Used to update an existing resource or create a new resource.
- OPTIONS − Used to get the supported operations on a resource.

What Are Web Services?

Answer: A web service is a collection of open protocols and standards used for exchanging data between applications or systems. Software applications written in various programming languages and running on various platforms can use web services to exchange data over computer networks like the Internet like inter-process communication on a single computer.

What Are Restful Web Services?

Answer: Web services based on REST Architecture are known as RESTful web services. These web services use HTTP methods to implement the concept of REST architecture. A RESTful web service

usually defines a URI, Uniform Resource Identifier a service, provides resource representation such as JSON and a set of HTTP Methods.

What Is A Resource In Rest?

Answer: REST architecture treats every content as a resource. These resources can be text files, HTML pages, images, videos or dynamic business data. REST Server simply provides access to resources and REST client accesses and modifies the resources. Here each resource is identified by URIs/ global IDs.

How To Represent A Resource In Rest?

Answer: REST uses various representations to represent a resource where text, JSON, XML. XML and JSON are the most popular representations of resources.

What Are The Best Practices To Design A Resource Representation?

Answer: Following are important points to be considered while designing a representation format of a resource in RESTful web services:

- Understandability − Both Server and Client should be able to understand and utilize the representation format of the resource.
- Completeness − Format should be able to represent a resource completely. For example, a resource can contain another resource. The format should be able to represent simple as well as complex structures of resources.
- Linkability − A resource can have a linkage to another resource, a format should be able to handles such situations.

Which Protocol Is Used By Restful Web Services?

Answer: RESTful web services make use of HTTP protocol as a medium of communication between client and server.

What Is Messaging In Restful Web Services?

Answer: A client sends a message in the form of an HTTP Request and the server responds in the form of an HTTP Response. This technique is termed as Messaging. These messages contain message data and metadata i.e. information about the message itself.

What Are The Core Components Of A Http Request?

Answer: An HTTP Request has five major parts :

- Verb − Indicate HTTP methods such as GET, POST, DELETE, PUT, etc.
- URI − Uniform Resource Identifier (URI) to identify the resource on the server.
- HTTP Version − Indicate HTTP version, for example HTTP v1.1 .
- Request Header − Contains metadata for the HTTP Request message as key-value pairs. For example, client (or browser) type, format supported by the client, format of the message body, cache settings, etc.
- Request Body − Message content or Resource representation.

What Are The Core Components Of A Http Response?

Answer: An HTTP Response has four major parts:-

- Status/Response Code − Indicate Server status for the requested resource. For example, 404 means resource not found and 200 means response is ok.
- HTTP Version − Indicate HTTP version, for example HTTP v1.1 .
- Response Header − Contains metadata for the HTTP Response message as key-value pairs. For example, content length, content type, response date, server type, etc.
- Response Body − Response message content or Resource representation.

What Is Addressing In Restful Web Services?

Answer: Addressing refers to locating a resource or multiple resources lying on the server. It is analogous to locate a postal address of a person.

What Is Uri?

Answer: URI stands for Uniform Resource Identifier. Each resource in a REST architecture is identified by its URI.

What Is Purpose Of A Uri In Rest Based Web Services?

Answer: The purpose of a URI is to locate a resource(s) on the server hosting the web service.

What Is Format Of A Uri In Rest Architecture?

Answer: A URI is of the following format:– <protocol>://<service-name>/<ResourceType>/<ResourceID>

What Is The Purpose Of Http Verb In Rest Based Web Services?

Answer: VERB identifies the operation to be performed on the resource.

What Are The Best Practices To Create A Standard Uri For A Web Service?

Answer: Following are important points to be considered while designing a URI −

- Use a Plural Noun − Use a plural noun to define resources. For example, we've used users to identify users as a resource.
- Avoid using spaces − Use underscore(_) or hyphen(-) when using a long resource name, for example, use authorized_users instead of authorized%20users.
- Use lowercase letters − Although URI is case-insensitive, it is good practice to keep url in lower case letters only.
- Maintain Backward Compatibility − As Web Service is a public service, a URI once made public should always be available. In case, URI gets updated, redirect the older URI to new URI using HTTP Status code, 300.

- Use HTTP Verb − Always use HTTP Verb like GET, PUT, and DELETE to do the operations on the resource. It is not good to use operations names in URI.

What Is Statelessness In Restful Web Services?

Answer: As per REST architecture, a RESTful web service should not keep a client state on the server. This restriction is called statelessness. It is the responsibility of the client to pass its context to the server and the server can store this context to process the client's further request. For example, a session maintained by the server is identified by a session identifier passed by the client.

What Are The Advantages Of Statelessness In Restful Web Services?

Answer: Following are the benefits of statelessness in RESTful web services:

- Web services can treat each method request independently.
- Web services need not maintain the client's previous interactions. It simplifies application design.
- As HTTP is itself a statelessness protocol, RESTful Web services work seamlessly with HTTP protocol.

What Are The Disadvantages Of Statelessness In Restful Web Services?

Answer: Following is the disadvantage of statelessness in RESTful web services:-

Web services need to get extra information in each request and then interpret to get the client's state in case client interactions are to be taken care of.

What Do You Mean By Idempotent Operation?

Answer: Idempotent operations mean their result will always the same no matter how many times these operations are invoked.

Which Type Of Web Services Methods Are To Be Idempotent?

Answer: PUT and DELETE operations are idempotent.

Which Type Of Web Services Methods Are To Be Read-Only?

Answer: GET operations are read-only and are safe.

What Is The Difference Between Put And Post Operations?

Answer: PUT and POST operation is nearly the same with the difference lying only in the result where PUT operation is idempotent and POST operation can cause a different result.

What Should Be The Purpose Of Options Method Of Restful Web Services?

Answer: It should list down the supported operations in a web service and should be read-only.

What Should Be The Purpose Of Head Method Of Restful Web Services?

Answer: It should return only HTTP Header, no Body and should be read-only.

What Is Caching?

Answer: Caching refers to storing server response in the client itself so that a client needs not to make a server request for the same resource again and again. A server response should have information about how a caching is to be done so that a client caches the response for some time or never caches the server response.

Which Header Of Http Response, Provides The Date And Time Of The Resource When It Was Created?

Answer: Date header provides the date and time of the resource when it was created.

Which Header Of Http Response, Provides The Date And Time Of The Resource When It Was Last Modified?

Answer: Last Modified header provides the date and time of the resource when it was last modified.

Which Header Of Http Response Provides Control Over Caching?

Answer: Cache-Control is the primary header to control caching.

Which Header Of Http Response Sets Expiration Date And Time Of Caching?

Answer: Expires header sets expiration date and time of caching.

Which Directive Of Cache-Control Header Of Http Response Indicates That Resource Is Cacheable By Any Component?

Answer: The public directive indicates that the resource is cacheable by any component.

Which Directive Of Cache-Control Header Of Http Response Indicates That Resource Is Catchable By Only Client And Server, No Intermediary Can Cache The Resource?

Answer: The private directive indicates that resource is cachable by only client and server, no intermediary can cache the resource.

Which Directive Of Cache-Control Header Of Http Response Indicates That Resource Is Not Cacheable?

Answer: no-cache/no-store directive indicates that the resource is not cacheable.

Which Directive Of Cache-Control Header Of Http Response Can Set The Time Limit Of Caching?

Answer: the max-age directive indicates that the caching is valid up to max-age in seconds. After this, the client has to make another request.

Which Directive Of Cache-Control Header Of Http Response Provides Indication To Server To Revalidate Resource If Max-age Has Passed?

Answer: the must-revalidate directive indicates the server to revalidate resources if max-age has passed.

What Are The Best Practices For Caching?

Answer: Always keep static contents like images, CSS, JavaScript cacheable, with an expiration date of 2 to 3 days. Never keep the expiry date too high. Dynamic contents should be cached for a few hours only.

What Are The Best Practices To Be Followed While Designing A Secure Restful Web Service?

Answer: As RESTful web services work with HTTP URLs Paths so it is very important to safeguard a RESTful web service in the same manner as a website is being secured. Following are the best practices to be followed while designing a RESTful web service:

- Validation − Validate all inputs on the server. Protect your server against SQL or NoSQL injection attacks.
- Session-based authentication − Use session-based authentication to authenticate a user whenever a request is made to a Web Service method.
- No sensitive data in URL − Never use username, password or session token in URL, these values should be passed to Web Service via the POST method.
- Restriction on Method execution − Allow restricted use of methods like GET, POST, DELETE. The GET method should not be able to delete data.
- Validate Malformed XML/JSON − Check for well-formed input passed to a web service method.
- Throw generic Error Messages − A web service method should use HTTP error messages like 403 to show access forbidden etc.

What Is The Purpose Of Http Status Code?

Answer: HTTP Status code is standard codes and refers to the predefined status of the task done at the server. For example, HTTP Status 404 states that the requested resource is not present on the server.

What Http Status Code 200 States?

Answer: It means, OK, shows a success.

What Http Status Code 201 States?

Answer: It means, CREATED when a resource is successfully created using POST or PUT request. Return link to a newly created resource using a location header.

What Http Status Code 204 States?

Answer: It means, NO CONTENT, when the response body is empty for example, a DELETE request.

What Http Status Code 304 States?

Answer: It means, NOT MODIFIED, used to reduce network bandwidth usage in case of conditional GET requests. The response body should be empty. Headers should have a date, location, etc.

What Http Status Code 400 States?

Answer: It means, BAD REQUEST, states that invalid input is provided e.g. validation error, missing data.

What Http Status Code 401 States?

Answer: It means, FORBIDDEN, states that user is not having access to the method being used, for example, delete access without admin rights.

What Http Status Code 404 States?

Answer: It means, NOT FOUND, states that the method is not available.

What Http Status Code 409 States?

Answer: It means, CONFLICT, states conflict situation while executing the method, for example, adding a duplicate entry.

What Http Status Code 500 States?

Answer: It means, INTERNAL SERVER ERROR, states that the server has thrown some exception while executing the method.

What Is Jax-rs?

Answer: JAX-RS stands for Java API for RESTful Web Services. JAX-RS is a JAVA based programming language API and specification to provide support for created RESTful Webservices. Its 2.0 version was released in 24 May 2013. JAX-RS makes heavy use of annotations available from Java SE 5 to simplify the development of JAVA based web services creation and deployment. It also provides support for creating clients for RESTful web services.

Explain REST?

Ans. REST stands for Representational State Transfer. REST is an architectural style of developing web services that take advantage of the ubiquity of HTTP protocol and leverages the HTTP method to define actions. It revolves around resource where every component is a resource which can be accessed by a common interface using HTTP standard methods.

In REST architecture, a REST Server provides access to resources and REST client accesses and presents those resources. Here each resource is identified by URIs of global IDs. REST uses different ways to represent a resource like text, JSON, and XML.XML and JSON are the most popular representations of resources these days.

What Is The Most Popular Way To Represent A Resource In REST?

Ans. REST uses different representations to define a resource like text, JSON, and XML.

JSON is the most popular representation of resources.

Explain What Is A “Resource” In REST?

Ans. REST architecture treats every content as a resource. These resources can be either text files, HTML pages, images, videos or dynamic business data.

REST Server provides access to resources and REST client accesses and modifies these resources. Here each resource is identified by URIs/ global IDs.

Which Protocol Is Used By RESTful Web Services?

Ans. RESTful web services make use of HTTP protocol as a medium of communication between client and server.

What Is Messaging In RESTful Web Services?

Ans. RESTful web services make use of HTTP protocol as a medium of communication between client and server. The client sends a message in the form of an HTTP Request.

In response, the server transmits the HTTP Response. This technique is called Messaging. These messages contain message data and metadata i.e. information about the message itself.

State The Core Components Of An HTTP Request?

Ans. Each HTTP request includes five key elements.

- The Verb which indicates HTTP methods such as GET, PUT, POST, DELETE.
- URI stands for Uniform Resource Identifier (URI). It is the identifier for the resource on the server.
- HTTP Version which indicates HTTP version, for example-HTTP v1.1.
- Request Header carries metadata (as key-value pairs) for the HTTP Request message. Metadata could be a client (or browser) type, the format that the client supports, message body format, and cache settings.
- Request Body indicates the message content or resource representation.

What is Rest Assured?

In order to test the REST APIs, I found the REST Assured library so useful. It is developed by JayWay Company and it is a really powerful catalyzer for automated testing of REST-services. REST-assured provides a lot of nice features, such as DSL-like syntax, XPath-Validation, Specification Reuse, easy file uploads and with those features we will handle automated API testing much easier.

How to declare the API details in the Rest Assured Test?

Using Given(), When(), Then()

Name The Most Commonly Used HTTP Methods Supported By REST?

Ans. There are a few HTTP methods in REST which are more popular.

- GET -It requests a resource at the request-URL. It should not contain a request body as it will get discarded. Maybe it can be cached locally or on the server.
- POST – It submits information to the service for processing; it should typically return the modified or new resource.
- PUT – At the request URL it updates the resource.
- DELETE – It removes the resource at the request-URL.
- OPTIONS -It indicates the supported techniques.
- HEAD – It returns meta-information about the request URL.

Is There Any Difference Between PUT And POST Operations? Explain It.

Ans. PUT and POST operation are almost the same. The only difference between the two is in terms of the result generated by them.

PUT operation is idempotent while POST operation can give a different result.

What Is URI? Explain Its Purpose In REST Based Web Services. What Is Its Format?

Ans. URI stands for Uniform Resource Identifier. URI is the identifier for the resource in REST architecture.

The purpose of a URI is to locate a resource(s) on the server hosting the web service. A URI is of the following format-

<protocol>://<service-name>/<ResourceType>/<ResourceID>

How to compare the response values with Rest Assured Assertion?

Example :

given().

parameters(“firstName”, “John”, “lastName”, “Doe”).

when().

post(“/greetXML”).

then().

body(“greeting.firstName”, equalTo(“John”)).

body(“greeting.lastName”, equalTo(“Doe”));

How to Insert cookies in Testing the API using Rest Assured?

given().cookie(“username”, “John”).when().get(“/cookie”).then().body(equalTo(“username”));

How to Insert headers in Testing the API using Rest Assured?

given().header(“MyHeader”, “Something”).

How to Validate Response Headers with Rest Assured?

get(“/x”).then().assertThat().header(“headerName”, “headerValue”).

How to handle Basic Authentication with Rest Assured?

given().auth().preemptive().basic(“username”, “password”).when().get(“/secured/hello”).then().statusCode(200);

What Do You Understand By Payload In RESTFul Web Service?

Ans. The request body of every HTTP message includes request data called as Payload. This part of the message is of interest to the recipient.

We can say that we send the payload in the POST method but not in <GET> and <DELTE> methods.

Question 1. Explain What Is Soap Ui?

Answer :

SOAP UI is a free, open source cross-platform functional testing solution. It enables you to rapidly and easily create and execute automated regression, compliance, functional and load tests.

Question 2. What Are Web Services?

Answer :

Web services are web components that transfer data between client and server. Client sends a web request to the server and the server then responds to client. The response and request are related and different requests evoke the corresponding response.

Web Service is a web component or software program that can be accessed on the Internet. It is mainly used to communicate with the web

based applications through XML messaging concepts. For example, if we want to access a particular location using Google Maps, we can use the corresponding web service URL. For that we have to pass the appropriate inputs.

Question 3. Does Soapui Support Ssl Authentication?

Answer :

Yes.

Question 4. What Is The Primary Challenge When Testing Web Services?

Answer :

Majority of the functional testing is carried out via the GUI; the biggest challenge of webservices is that they do not have a UI.

Question 5. Explain What Is The Role Of Xml, Soap, Wsdl And Uddi In Web Services?

Answer :

Web services are a medium through which web based application can be integrated or communicated over an internet protocol backbone. Web based application can be integrated using XML, SOAP, UDDI and WSDL. SOAP is used for transmitting the data; WSDL is used for describing or relating the services, UDDI is used for listing what services are accessible, XML is used to tag the data.

Question 6. What Is Groovy Script And Where Can It Be Used?

Answer :

Groovy is a scripting language which internally includes all the java libraries – it helps us to customize and add custom validations to SoapUI tests

Question 7. Inside The Soapui Where Groovy Script Can Be Used?

Answer :

Inside the SOAPUI groovy script can be used in two places

- Groovy script test step
- Script Assertion- within test step

Question 8. What Are The Communication Channels Available For A Web Service?

Answer :

In general, web service is combined with the following protocols:

- HTTP / POST
- HTTP / GET
- SOAP

While exposing the web services, these channels will be used for communication with the clients. Here HTTP / POST protocol transfers the information between the clients with secure mode. HTTP / GET protocol allows the clients to view transferred data partially at the browser's address bar. SOAP is used for transferring the confidential data safely.

Question 9. Mention What Soapui Can Do?

Answer :

SoapUI can:

- It can do automated testing like load tests, scenario-based tests and data driven tests
- It has got built-in reporting abilities

- It has got unique ability to impersonate web services and run functional and load test against them even before they are executed

Question 10. Explain About Xpath Assertion In Soapui?

Answer :

In SoapUI, XPath assertion is used for asserting the web service response value by specifying the absolute path. If the absolute path is matched with the response value, then the test case or test suite will be considered as PASS otherwise it will be notified as FAILED. We can see the results of assertion at bottom of the screen where the Assertion tab will have resultant information.

Question 11. Mention What Is The General Format For Reading The Custom Or Default Property Value?

Answer :

The general format for reading the custom or default property value is ${#levelname#key}

Question 12. What Are The Different Components Can Be Used In The Web Services?

Answer :

There are five components are used in web services.

They are:

- WSDL – Web Service Description Language
- SOAP – Simple Access Object Protocol
- UDDI – Universal Description, Discovery and Integration
- RDF – Resource Description Framework
- XML – eXtensible Markup Language

Question 13. Mention How Soapui Is Used To Structure Functional Tests?

Answer :

SOAP UI uses three levels to structure functional tests:

- **TestSuite:** It is a pool of test cases used for combining functional tests into logical units
- **TestCase:** It is a group of test steps that are bring together to test some specific aspect of your services. You can augment any number of testcases to test suites
- **TestSteps:** They are “building blocks” of functional tests in SOAPUI. They together makes TestCase and determines the flow of execution of the service to be tested.

Question 14. What Are Soapui Assertions?

Answer :

Assertions compare the parts/all of the response message to the expected outcome.

Question 15. Explain How You Can Parametrize The Endpoints In Soap Ui?

Answer :

To parameterize the endpoints in SOAP UI is most important as it is the first step in automation testing. It is more time consuming for changing endpoints manually. In order to do that

- Define a project property holding the endpoint
- Change the endpoint to use this property via property expansion
- Ensure that your requests are availing the configured endpoint
- When a request is run, the property will inevitably be with its current value. To use a different value just configure the endpoint in the UI, you can use the P option from the command line

Question 16. What Are The Tools Used For Testing Web Services?

Answer :

To perform functional testing for web services, we can use the following tools.

- SoapUI
- RESTClient – This is a Firefox plug-in
- JMeter – Specially made it for performance testing tool and also we can do functional testing the web services.

Question 17. Explain What Is The Properties In Soapui?

Answer :

Properties in SOAPUI is used to retrieve and store data. The data is stored as key, value or format.

Question 18. What Hierarchy Soapui Follows To Build A Proper Testing Project?

Answer :

In a SoapUI project, the following order should be maintained:

- TestSuite – This is combination of functional tests and logical blocks
- Testcase – Its a group that contains several test steps for the specific aspects of the service.
- Teststep – it contains the set of functional tests

Question 19. Explain Where You Use Properties In Soapui?

Answer :

We can read the property values into test step endpoint, username, header values, password, domain, POST, PUT, GET and DELETE method

properties.

Question 20. What Is The Role Of Wsdl Document In Web Service Testing?

Answer :

Validating webservices in only possible with WSDL document because to configure web services in SoapUI, WSDL document is mandatory. If the WSDL document is not valid, SoapUI will throw an exception immediately.

Question 21. What Is Meant By Protocols And What Are The Major Types Are Used In Web Services?

Answer :

A protocol is a set of standard rules that helps to communicate the hardware devices through the software applications. There are different types of protocols used in the Internet and Intranet applications. They are

- TCP which stands for Transmission Control Protocol. It has the rules to exchange the messages between two different Internet applications.
- Internet Protocol uses the rules for sending and receiving the information between two different Internet addresses.
- Similarly, HTTP, FTP and DHCP protocols are used the set of rules to transfer the data other than Internet applications.

Question 22. What Are The Advantages Of Soap?

Answer :

Since its XML based, it is platform and programming language independent. RPC (Remote procedure calls) are sometimes blocked by firewalls and proxy servers- Soap overcomes that.

Question 23. What Is Soap?

Answer :

Simple Object access protocol that uses XML to interact with web applications. It uses XML based content to communicate between two client machines across any network

Question 24. What Is Data-driven Testing? How Can You Do This Using Soap Ui?

Answer :

Data-driven testing is when you store test data (input, expected output, etc) in some external storage (database, spreadsheet, xml-files, etc) and then use that data iteratively in your tests when running them. For example to test your phone-lookup service, you might have a list of names and expected phone-numbers in a database which you would use to "drive" your test, checking that each name gets the right phone-number back. It's really quite simple.

Question 25. How Do I Save The Changes I Make To My Projects, Requests, Test Cases, Etc?

Answer :

Soap UI automatically saves everything on exit. If you want to save your projects without exiting (for example if you want to commit your project file to CVS) use the "Save All" option in the main File menu

Question 26. How To Access The Project Name From A Groovy Script Test Step?

Answer :

testRunner.testCase.testSuite.project.name (Almost all items have a name property)

Question 27. How To Create A Mock Service?

Answer :

- Right-click on one of the SOAP interfaces and selects Generate Mock Service.
- In the dialog Generate Mock Service you can specify the local port/path for the service you're creating but for the moment just click OK.
- Enter the name of your Mock Service in the Name dialog and click OK.
- after creating the Mock Service, you should get a Mock Service with one operation and one request.

Question 28. Name A Few Http Status Codes/messages?

Answer :

- 200 OK This response code indicates that the request was successful.
- 201 Created This indicates the request was successful and a resource was created. It is used to confirm success of a PUT or POST request.
- 400 Bad Requests The request was malformed. This happens especially with POST and PUT requests, when the data does not pass validation, or is in the wrong format.
- 404 Not Found This response indicates that the required resource could not be found. This is generally returned to all requests which point to a URL with no corresponding resource.
- 401 Unauthorized. This error indicates that you need to perform authentication before accessing the resource.
- 405 Method Not Allowed The HTTP method used is not supported for this resource.
- 409 Conflicts This indicates a conflict. For instance, you are using a PUT request to create the same resource twice.
- 500 Internal Server Errors When all else fails; generally, a 500 response is used when processing fails due to unanticipated circumstances on the server side, which causes the server to error out.

Question 29. What Languages Are Supported In Soapui?

Answer :

Groovy and JavaScript

Question 30. What Would Be The Message Format Of Soap Protocol?

Answer :

Generally, all the SOAP based web services are written by using XML language which uses standard message format that is accepted across the universe. In this format, it is easy to read, identify the errors, avoids interoperability problems etc.

Here's the sample SOAP message format.

POST /InStock HTTP/1.1

Host: localhost

Content-Type: application/soap+xml; charset=utf-8

Content-Length: 399

SOAPAction: "http://www.wisdomjobs.com/soap-envelope"

Question 31. What Does A Soap Document Contain?

Answer :

Envelope element is the top most tag which identifies the XML document as a SOAP message. Followed by Envelope element, you see the header element that has header information. The Body element specifies the call and response information. Finally, you have a Fault element which contains errors and status information.

Question 32. How To Start And Stop Mock Services?

Answer :

This can come in handy both from the Project onLoad script (if you want to start your MockServices when the project is opened) or from a TestCase/TestSuite setup script (if you need the MockService running for your functional tests). Here from a TestCase setup script:

- def runner = testCase.testSuite.project.mockServices["My MockService"].start()
- context.mockRunner = runner

The returned runner object is required to stop the MockService... so save it to the context and use it in the tearDown script:

context.mockRunner.stop()

Have a look at the WsdlMockService and the WsdlMockRunner classes for more methods and properties that might come in handy.

Question 33. How To Add An Assertion To The Test?

Answer :

- Open the request editor.
- In the request editor, click the Add an Assertion to Test Request button.
- Select Response SLA from the drop down in the Select Assertion dialog box.
- In the Configure Response SLA Assertion dialog box, write 500 and click OK. This will validate that the response of the SLA is fewer than 500.
- Now that you've added the assertion, you are going to run the request to validate the response. If all assertions are successful, the SOAP icon should turn green in three places .
- You can also validate the response by adding an X-Path Match assertion.
- In Soap UI Pro you can open the Select X-Path dialog, which lets you select nodes to assert using point-and-click. As a result Soap UI creates the matching X-Path expression that refers to the selected node.
- Soap UI automatically populates the expected result to match what's in the response. But of course, you can change this to what's relevant for you.

Question 34. What Exactly Wsdl Document Contains?

Answer :

- It is a document written in XML and used to describe web services.
- It specifies the location of the service and the operations (or methods) the service exposes.

Question 35. Explain Soapui And Soapui Pro?

Answer :

SoapUI is a webservices testing tool and SoapUI Pro is its commercial version. SoapUI can help create functional, security and load testing test suites. SoapUI Pro does all that with advanced drag and drop, data driven testing, advanced reporting and coverage analysis.

Question 36. What Are The Different Types Of Assertions Used In Soapui?

Answer :

The following are the different types of assertions:

- Contains & Not Contains
- XPath match
- XQuery match
- Schema compliance
- Soap Faults
- Response SLA
- WS security Status
- Script Assertion
- WS- Addressing Request or Response Assertion

Question 37. What Is Data Driven Testing?

Answer :

Data driven testing means to store our test data which includes input and expected output in an external data source called Excel / Database / XML file. Later, we need to iterate the data source using respective component. In SoapUI, Datasource and Datasource Loop test steps are used for performing data driven testing.

Question 38. What Are The Major Types Of Assertions Available In Soapui?

Answer :

Assertions are the one of the major feature in SoapUI. It offers the following types of assertions:

- Simple contains
- Schema compliance
- Simple not contains
- Soap Faults
- Response SLA
- XPath Match
- XQuery Match
- WS security status
- Script Assertion
- WS- Addressing Request or Response Assertion
- Additionally Equals assertion is introduced in SoapUI NG Pro version.

Question 39. What Is The Basic Method To Automate Web Services In Soapui?

Answer :

- Create a project and add the WSDL file
- Add test suites, Test cases and Test cases- in that order
- Include custom programming/validation using by adding Groovy steps
- Call external data sources if using
- Add assertions if necessary
- Then RUN.

Question 1. What Is Jmeter?

Answer :

JMeter is one of the Java tools which is used to perform load testing client/server applications. Apache JMeter is open source software, a 100% pure Java desktop application designed to load test functional behavior and measure performance of the application. It was originally

designed for testing Web Applications but has since expanded to other test functions.

Question 2. Explain How Jmeter Works?

Answer :

JMeter acts like a group of users sending requests to a target server. It collects response from target server and other statistics which show the performance of the application or server via graphs or tables.

Question 3. Explain Where You Can Use Functions And Variables?

Answer :

Variables and functions can be written into any field of any test component.

Question 4. Mention What Are Regular Expressions In Jmeter?

Answer :

Based on the pattern, regular expression are used to search and manipulate text. JMeter is useful in interpreting forms of regular expression or patterns being used throughout a JMeter test plan.

Question 5. Explain What Is Samplers And Thread Groups?

Answer :

Thread group: For any test plan, JMeter is the beginning part of thread group elements. It is an important element of JMeter, where you can set number of users and time to load all the users given in the thread group

Samplers: Sampler generates one or more sample results; these sample results have many attributes like elapsed time, data size, etc. Samplers

allow JMeter to send specific types of requests to the server, through samplers, thread group decides which type of request it need to make. Some of the useful samplers are HTTP request, FTP request, JDBC request and so on.

Question 6. Whether The Test Plans Built Using Jmeter Are Os Dependent?

Answer :

Usually, test plan are saved in their XML format, so there is nothing to do with any particular O.S. It can be run on any OS where JMeter can run.

Question 7. Mention What Are The Types Of Processor In Jmeter?

Answer :

The types of processor in JMeter are:

- Pre-processor
- Post processor

Question 8. Explain What Are Pre-processor Elements? List Some Of The Pre-processor Elements?

Answer :

A pre-processor is something that will happen before sampler executes. To configure the sample request prior to its execution or to update variables that are not extracted from response text pre-processor elements are used.

Some of the pre-processor elements are:

- HTTP URL re-writing modifier
- HTTP user parameter modifier
- HTML link parser

- BeanShell PreProcessor

Question 9. Mention The Execution Order Of Test Elements?

Answer :

The test plans elements execution order is

- Configuration elements
- Pre-processors
- Timers
- Samplers
- Post-processors
- Assertions
- Listeners

Question 10. What Does “contain” And “matches” Indicates In The Regular Expression?

Answer :

In the regular expression, contains indicates that the regular expression matched at least some part of the target. While matches means the regular expression matched the whole target. So, ‘alphabet’ is “matched” by ‘al.*t.’

Question 11. Explain What Is Configuration Elements?

Answer :

A configuration element works parallel with a Sampler. To set up defaults and variables for later use by samplers configuration elements can be used. At the start of the scope, these elements are processed before any samplers in the same scope.

Question 12. Explain What Is A Timer In Jmeter And What Are The Types Of It?

Answer :

A JMeter thread by default will send requests continuously without any pause. To get a pause between the request, Timers are used. Some of the Timers used are Constant Timer, Gaussian Random Timer, Synchronizing Timer, Uniform Random Timer and so on.

Question 13. Explain What Is Test Fragment?

Answer :

Test fragment is also a type of element like Thread Group element. The only difference is test fragment is not implemented unless it is referenced by either a Module controller or an Include controller.

Question 14. Explain What Is Assertion In Jmeter? What Are The Types Of Assertion?

Answer :

Assertion helps to verify that your server under test returns the expected results

Some commonly used Assertion in JMeter are

- Response Assertion
- Duration Assertion
- Size Assertion
- XML Assertion
- HTML Assertion

Question 15. Explain How You Can Reduce The Resource Requirement In Jmeter?

Answer :

To reduce the resource requirements in JMeter

- Use non-GUI mode: jmeter –n –t test.jmx –l test.jtl
- During the load, test doesn't use "view results tree" or "view results in table" listeners, use them only during scripting phase
- Don't use functional mode
- Instead of using lots of similar samplers, use the same sampler in loop and use variable to vary the sample

Question 16. Explain How You Can Perform Spike Testing In Jmeter?

Answer :

By synchronizing, timer JMeter spike testing can be achieved. Synchronizing timer, blocks thread until a specific amount of threads has been blocked and then release them all together thus creating large instantaneous load.

Question 17. Explain How You Can Capture The Script Of The Authentication Window In Jmeter?

Answer :

Normally, you can capture script by recording.

- First you have to Threadgroup in Testplan and then make HTTPProxyServer in Workbench
- After that, set port number in Global Setting box (e.g., 8911) and modify your connection setting in IE as local host in address 8911 as in port Then you can start http proxy server in JMeter and run your application for login.

Question 18. List Out Few Jmeter Listeners?

Answer :

Some of the JMeter Listeners are:

- Spline Visualizer
- Aggregate Report
- View Result Tree

- View Result in Table
- Monitor Results
- Distribution Graph
- BeanShell Listener
- Summary Report and so on

Question 19. What Is Distributed Load Testing? How It Can Be Achieved?

Answer :

Distributed load testing is the process through which numerous systems can be used for simulating load of a large number of users. By using the master-slave configuration, JMeter can do distribute load testing.

Question 20. In Jmeter Is It Necessary To Call Embedded Resources Explicitly?

Answer :

You can eliminate all embedded resources from being explicitly called. Requests have a checkbox at the bottom that says “retrieve embedded resources.” It would grab all CSS, JPG, etc. It is a brilliant way to find resources and broken link in a web App.

Question 21. Explain What Is The Role Of Timer In Jmeter?

Answer :

With the help of timer, JMeter can delay the time between each request, which a thread makes. It can solve the overload problem of the server.

Question 22. Explain What Is Post-processor?

Answer :

To perform any action after making a request, Post- processor is used. For example, if JMeter sends an http request to the web server, and if you want JMeter to stop sending the request if the web server shows an error, then you will use post-processor to perform this action.

Question 23. What Are The Benefits That Jmeter Offers For Performance Testing?

Answer :

JMeter offers benefits on performance testing like

- It can be used to test performance for both, static resources as well as dynamic resources
- It can handle a maximum number of concurrent users then your website can handle
- It provides the graphical analyses of performance reports

Question 24. What Are The Protocols Supported By Jmeter?

Answer :

The protocols supported by JMeter are:

- Web: HTTP, HTTPS sites 'web 1.0' web 2.0 (ajax, flex and flex-ws-amf)
- Web Services: SOAP / XML-RPC
- Database via JDBC drivers
- Directory: LDAP
- Messaging Oriented service via JMS
- Service: POP3, IMAP, SMTP
- FTP Service

Question 25. List Some Of The Features Of Jmeter.?

Answer :

Following are some of the features of JMeter:

- Its free. Its an open source software.
- It has simple and intuitive GUI.

- JMeter can load and performance test many different server types: Web - HTTP, HTTPS, SOAP, Database via JDBC, LDAP, JMS, Mail - POP3
- It is platform-independent tool. On Linux/Unix, JMeter can be invoked by clicking on JMeter shell script. On Windows it can be invoked by starting the jmeter.bat file.
- It has full Swing and lightweight component support (precompiled JAR uses packages javax.swing.*).
- JMeter store its test plans in XML format. This means you can generate a test plan using a text editor.
- It's full multi-threading framework allows concurrent sampling by many threads and simultaneous sampling of different functions by separate thread groups.
- It is highly Extensible.
- Can also be used to perform automated and functional testing of your application.

Question 26. What Is A Test Plan In Jmeter?

Answer :

A Test Plan defines and provides a layout of how and what to test. For example the web application as well as the client server application. It can be viewed as a container for running tests. A complete test plan will consist of one or more elements such as thread groups, logic controllers, sample-generating controllers, listeners, timers, assertions, and configuration elements. A test plan must have at least one thread group.

Question 27. List Some Of The Test Plan Elements In Jmeter.?

Answer :

Following is a list of some of the test plan elements:

- ThreadGroup
- Controllers
- Listeners
- Timers
- Assertions
- Configuration Elements
- Pre-Processor Elements
- Post-Processor Elements

Question 28. What Is Thread Group?

Answer :

Thread Group elements are the beginning points of your test plan. As the name suggests, the thread group elements control the number of threads JMeter will use during the test.

Question 29. What Are Controllers And Its Types?

Answer :

JMeter has two types of Controllers:

Samplers Controllers : Samplers allow JMeter to send specific types of requests to a server. They simulate a user's request for a page from the target server. For example, you can add a HTTP Request sampler if you need to perform a POST, GET, DELETE on a HTTP service

Logical Controllers : Logic Controllers let you control order of processing of Samplers in a Thread. Logic Controllers can change the order of request coming from any of their child elements. Some examples are: ForEach Controller, While Controller, Loop Controller, IF Controller, Run Time Controller, Interleave Controller, Throughput Controller, Run Once Controller.

Question 30. What Is Configuration Element?

Answer :

Configuration Elements allow you to create defaults and variables to be used by Samplers. They are used to add or modify requests made by Samplers.

They are executed at the start of the scope of which they are part, before any Samplers that are located in the same scope. Therefore, a Configuration Element is accessed only from inside the branch where it is placed.

Question 31. What Are Listeners?

Answer :

Listeners let you view the results of Samplers in the form of tables, graphs, trees or simple text in some log files. They provide visual access to the data gathered by JMeter about the test cases as a Sampler component of JMeter is executed.

Listeners can be added anywhere in the test, including directly under the test plan. They will collect data only from elements at or below their level.

Question 32. What Are Pre-processor And Post-processor Elements?

Answer :

A Pre-Procesor is something that will happen before a sampler executes. They are often used to modify the settings of a Sample Request just before it runs, or to update variables that are not extracted from response text.

A Post Processor executes after a sampler finishes its execution. This element is most often used to process the response data, for example, to retrieve particular value for later use.

Question 33. What Is The Execution Order Of Test Elements?

Answer :

Following is the execution order of the test plan elements:

- Configuration elements
- Pre-Processors
- Timers
- Sampler
- Post-Processors (unless SampleResult is null)
- Assertions (unless SampleResult is null)
- Listeners (unless SampleResult is null)

Question 34. How Do You Ensure Re-usability In Your Jmeter Scripts?

Answer :

- Using config elements like "CSV Data Set Config", "User Defined Variables", etc for greater data reuse.
- Modularizing shared tasks and invoking them via a "Module Controller".
- Writing your own BeanShell functions, and reusing them.

Question 35. Are The Test Plans Built Using Jmeter Os Dependant?

Answer :

Test plans are usually saved in thr XML format, hence they have nothing to do with any particular OS. You can run those test plans on any OS where JMeter can run.

Question 36. What Are The Monitor Tests?

Answer :

Uses of monitor tests are:

- Monitors are useful for a stress testing and system management.
- Used with stress testing, the monitor provides additional information about server performance.
- Monitors makes it easier to see the relationship between server performance and response time on the client side.
- As a system administration tool, the monitor provides an easy way to monitor multiple servers from one console.

Question 37. What Are Jmeter Functions?

Answer :

JMeter functions are special values that can populate fields of any Sampler or other element in a test tree.

A function call looks like this:

${__functionName(var1,var2,var3)}

Question 38. Where Can Functions And Variables Be Used?

Answer :

Functions and variables can be written into any field of any test component.

Question 39. What Are Regular Expressions In Jmeter?

Answer :

Regular expressions are used to search and manipulate text, based on patterns. JMeter interprets forms of regular expressions or patterns being used throughout a JMeter test plan, by including the pattern matching software Apache Jakarta ORO.

Question 40. How Can You Reduce Resource Requirements In Jmeter?

Answer :

Below are some suggestion to reduce resource requirements:

- Use non-GUI mode: jmeter -n -t test.jmx -l test.jtl.
- Use as few Listeners as possible; if using the -l flag as above they can all be deleted or disabled.
- Disable the “View Result Tree” listener as it consumes a lot of memory and can result in the console freezing or JMeter running out of memory. It is, however, safe to use the “View Result Tree” listener with only “Errors” checked.
- Rather than using lots of similar samplers, use the same sampler in a loop, and use variables (CSV Data Set) to vary the sample. Or perhaps use the Access Log Sampler.
- Don't use functional mode.
- Use CSV output rather than XML.
- Only save the data that you need.
- Use as few Assertions as possible.

- Disable all JMeter graphs as they consume a lot of memory. You can view all of the real time graphs using the JTLs tab in your web interface.
- Do not forget to erase the local path from CSV Data Set Config if used.
- Clean the Files tab prior to every test run.

1. Why do we use Jenkins?

Answer: Jenkins is a continuous integration software tool which is open -sourced and written in the Java programming language to test and report on isolated changes in a larger code base in real time. The **Jenkins software can enable developers for finding and solving defects in a code base rapidly and for automate testing of builds.**

2. What is Maven and what is Jenkins?

Answer: Maven is a build tool which is a successor of ant. It will help in build and version control. **Jenkins is continuous integration system which** can be used to automate the deployment process.

3. What is continuous integration in Jenkins?

Answer: In Continuous integration process all development work is integrated at earliest. The resulting artifacts will be automatically created and tested. This process will allow to identify errors at earliest. **Jenkins is a popular open source tool** which will perform continuous integration and build automation.

4. Why do we use Jenkins with selenium?

Answer: Running Selenium tests in Jenkins will allow to run tests every time software changes and deploy the software to a new environment when the tests will pass. Jenkins will schedule tests to run at specific time.

5. What are CI Tools?

Answer: The list of the top eight **Continuous Integration tools:**

1. Jenkins

2. TeamCity

3. Travis CI

4. Go CD

5. Bamboo

6. GitLab CI

7. CircleCI

8. Codeship

6. What is a CI CD pipeline?

Answer: A **continuous integration** and deployment pipeline **(CD/CI)** is an important aspect of a software project. It will save a ton of manual, error-prone deployment work. It will result in higher quality software for continuous integration, **automated tests**, and code metrics.

7. What is build pipeline in Jenkins?

Answer: In **Jenkins**, Job chaining is the process of automatically starting other job(s) after the execution of a job. This approach will build **multi-step build pipelines** or trigger the rebuild of a project, if one of its dependencies will be updated.

8. What is a Jenkins pipeline?

Answer: The **Jenkins Pipeline plugin** is based on a Domain Specific Language (DSL) in Groovy, the Pipeline plugin will make pipelines scriptable. This is an incredibly powerful way for developing complex, multi-step **DevOps pipelines**.

9. What is a DSL Jenkins?

Answer: The Jenkins "Job DSL / Plugin" has two parts: The Domain Specific Language (DSL) itself that will allow users for describing jobs using a Groovy-based language, and a Jenkins plugin that will manage the scripts and the update of the Jenkins jobs which will be created and will be maintained as a result.

10. What is continuous integration and deployment?

Answer: Continuous Integration (CI) is a development practice which will require developers to integrate code into a shared repository several times a day. Each check-in will be verified by an automated build and allows teams to detect problems earliest.

11. What is the tool used for provisioning and configuration?

Answer: Ansible is an agent-less configuration management and orchestration tool. In Ansible, the configuration modules will be called as Playbooks. Like other tools, Ansible will be used to cloud provision.

12. What is the difference between Maven, Ant and Jenkins?

Answer: Maven and ANT both is build tool. Maven will provide dependency management, standard project layout and project management. Jenkins is a continuous integration tool that is much more than build tool.

It will consist of a pom.xml file which is specified in Jenkins to run the code. Whereas, Jenkins can be used as a continuous integration tool and automate the deployment process. The reports of the builds will be used to set a mark for continuous delivery as well.

Following are the differences between Maven, Ant, and Jenkins:

Maven	Ant	Jenkins
It is a Build	Java	Continuous

Automation Tool.	Library/Command Line Tool.	Integration Tool.
Defines how the software is built and describes the software dependencies.	Drives build process.	Automates the software development process with Continuous Integration and facilitates Continuous Deliver.
Supports projects written in C#, Ruby.	Supports projects written in C and C++.	Supports version control tools like Git, AccuRev.
Executes Unit Tests will be the part of the normal build cycle.	Supports single file execution introduced with Java II.	Can execute Apache Ant and Apache Maven.

13. Which SCM tools Jenkins supports?

Answer: Jenkins can support version control tools that include AccuRev, CVS, Subversion, Git, Mercurial, Perforce, ClearCase and RTC, and will execute Apache Ant, Apache Maven and sbt based projects and an arbitrary shell scripts and Windows batch commands.

14. How schedule a build in Jenkins?

Answer: In Jenkins, we will define various build triggers under the job configuration.

1. Find the 'Build Triggers' section,

2. check the ' Build Periodically' checkbox.

With the periodically build we will schedule the build definition by the date or day of the week and the time for executing the build.

15. Why do we use Pipelines in Jenkins?

Answer: Pipeline will add a powerful set of automation tools onto Jenkins, which supports use cases. Use cases will span from simple continuous integration to comprehensive continuous delivery pipelines. By modeling a series of related tasks, users will take advantage of the many features of Pipeline:

Code: Pipelines will be implemented in code. It will typically check into source control, providing teams the ability to edit, review, and iterate upon their delivery pipeline.

Durable: Pipelines will survive both planned and unplanned restarts of the Jenkins master.

Pausable: Pipelines will optionally stop and wait for human input or approval before continuing the Pipeline run.

Versatile: Pipelines will support complex real-world continuous delivery requirements, which will include the ability for fork/join, loop, and perform work in parallel.

Extensible: The Pipeline plugin will support custom extensions to its DSL and multiple options to integrate with other plugins.

16. What is a Jenkinsfile?

Answer: Jenkinsfile is a text file which has the definition of a Jenkins Pipeline and will be checked into source control.

Creating a Jenkinsfile, which will be checked into source control, will provide a number of immediate benefits:

1. Code review/iteration on the Pipeline

2. Audit trail for the Pipeline

Single source of truth for the Pipeline, that will be viewed and will be edited by multiple members of the project.

17 What is blue ocean in Jenkins?

Answer: Blue Ocean is a project which will model and present the process of software delivery by surfacing information which is important for development teams with as few clicks as possible, while still will be staying true to the extensibility which is core to Jenkins.

18. What are the important plugins in Jenkins?

Answer: List of some important **Plugins in Jenkins:**

1. Maven 2 project

2. Git

3. Amazon EC2

4. HTML publisher

5. Copy artifact

6. Join

7. Green Balls

19. What are Jobs in Jenkins?

Answer:

will be used for performing the typical build server work, like doing continuous/official/nightly builds, run tests, or perform some repetitive batch tasks. These jobs are called **"free-style software project"** in Jenkins.

20. How do you create a Job in Jenkins?

Answer: Steps to create a Job in Jenkins:

1. Go to **Jenkins** top page, select “New Job”,

2. choose “Build a free-style software project”.

This job type consists of the following elements:

Optional **SCM,** like **CVS** or **Subversion** where source code will be resided.

Optional triggers for controlling when Jenkins will perform builds.

some sort of build script which will perform the build **(ant, maven, shell script, batch file, etc.)** where the real work will happen optional steps for collecting information out of the build, like archiving the artifacts and/or recording javadoc and test results.

Optional steps to notify other people/systems with the build result, like sending e-mails, IMs, updating issue tracker, etc.

21. How do you configuring automatic builds in Jenkins?

Answer: Builds in Jenkins will be triggered periodically on a schedule. It is specified in configuration, or when source will change in the project have been detected, or they will be automatically triggered by requesting the URL:
http://YOURHOST/jenkins/job/PROJECTNAME/build

22. How to create a backup and copy files in Jenkins?

Answer: To create a backup, all we required to do is to periodically back up **JENKINS_HOME** directory. This will contain all of build jobs configurations, we will slave node configurations, and build history. For creating a back-up of Jenkins setup, just copy this directory.

23. What are the various ways in which build can be scheduled in Jenkins?

Answer: Builds will be triggered by source code management commits, will be triggered after completion of other builds and it can be scheduled to run at specified time (crons) Manual Build Requests

24. What is the relation between hudson and Jenkins?

Answer: Hudson was the earlier version of current Jenkins and the project name was changed from Hudson to Jenkins.

25. What you do to make sure that your project build doesn’t break in Jenkins?

Answer: To make sure that project build doesn’t break in Jenkins, perform successful clean install on my local machine with all unit tests. Check all code changes and Synchronize it with repository for making sure that all required config and POM changes and any difference will be checked into the repository.

26. What you do when you see a broken build for your project in Jenkins?

Answer: Open the console output for the build and will try to see if any file changes were missed when we see a broken build build for project in Jenkins.

If not able to find the issue in the above way, it should be cleaned and update local workspace to replicate the problem on local and try to solve it.

27. Explain what is continuous integration?

Answer: When multiple developers will work on different segments of same web application, it is required to perform integration test by integrating all modules in software development. An automated process for each piece of code is performed daily therefore that all code will be tested to perform this task.

28. What is the requirement for using Jenkins?

Answer: Following are the requirement for using Jenkins:

1. A source code repository that is accessible, for instance, a Git repository

2. A working build script, example for a Maven script, checked into the repository.

29. What are the advantages of Jenkins?

Answer: Following are some advantages of using Jenkins or by any matter any integration tools:

1. it saves developer time: Most of the integration task is being handled by Jenkins via automatic integration; the developer time will be focused on development activities mostly.

2. Improved software quality: Since the software is can be tested immediately after any code check-in, it will keep the quality check frequently, thus improves overall software quality.

3. Faster Delivery: Jenkins automatically perform continuous integration, which will lead to very early detection of bugs / defects and hence it will lead to faster delivery of software.

4. Decreased development time: Since most of the integration work will be automated by Jenkins, it will lead to the faster development of application.

5. Easily portable: Since Jenkins will be developed using Java, it will easily portable to other platforms.

6. Early tracking of defects: Jenkins will help tacking of defects at very early stage in development environment only rather than production environment.

Email Notification to developers: Jenkins is easily integrated with LDAP server, so developer can be notified about build success / failure via mail.

30. What is the difference between Jenkins and Bamboo??

Answer:

Parameters	Jenkins	Bamboo

Open Source	It is open-source	It is not open source
Pricing Logic	completely free	charges for the number of build agents required
Operating System	Windows, Ubuntu, Red Hat, Mac OS	Windows, Linux, Solaris
Browsers	Chrome, Firefox, Internet Explorer	Firefox, Chrome, Safari, Edge
Plugin Support	supports a lot of plugins	It does not support many plugins as Jenkins
Support	It has a lot of support from communities	It has less support

31. Define the process of Jenkins.

Answer: 1. A developer will commit the code to the source code repository. The Jenkins server will check the repository at regular intervals for changes.

2. The Jenkins server will detect the changes which have occurred in the source code repository. Jenkins can pull those changes and will start preparing a new build.

3. If the build will fail, then the concerned team can be notified.

4. If the build will be successful, then Jenkins can deploy the build in the test server.

5. After testing, Jenkins will generate feedback and then it will notify the developers about the build and test results.

6. It can continue to check the source code repository for changes made in the source code and the whole process will keep on repeating.

32. Mention what are the commands you can use to start Jenkins manually?

Answer: To start Jenkins manually, we will use either of the following

1. (Jenkins_url)/restart: Forces a restart without waiting for builds for completeion

2. (Jenkin_url)/safeRestart: Allows all running builds for completion.

33. List useful plugins in Jenkin?

Answer: Some of the important plugins in Jenkin will include

1. Maven 2 project

2. Amazon EC2

3. HTML publisher

4. Copy artifact

5. Join

6. Green Balls

34. Explain how you can deploy a custom build of a core plugin?

Answer: To deploy a custom field of a core plugin,We have to do following things:

1. Stop Jenkins

2. Copy the custom HPI to $Jenkins_Home/plugins

3. Delete the previously expanded plugin directory

4. Make an empty file called .hpi.pinned

5. Start Jenkins

35. Explain how you can clone a Git repository via Jenkins?

Answer: To clone a Git repository via Jenkins, we have to enter the e-mail and user name for Jenkins system. To do that we have to switch into job directory and execute the "git config" command.

36. Explain how you can set up Jenkins job?

Answer: To create a project this is handled via jobs in Jenkins.

1. Select new item from the menu.

2. Once this done enter a name for the job and select free-style job.

3. Click OK to create new job in Jenkins.

4. The next page enables to configure job.

37. Mention what are the two components Jenkins is mainly integrated with?

Answer: Jenkin is integrated with following components

1. Version Control system like GIT, SVN

2. Build tools like Apache Maven

38.Mention what are the commands you can use to start Jenkins manually?

Answer: Following are the commands we can use to start Jenkins manually:

 # service jenkins start

 # service jenkins stop

 # service jenkins restart

(or)

 # systemctl start jenkins

 # systemctl stop jenkins

 # systemctl restart jenkins

39. How do you restart jenkins without disturbing the running jobs?

Answer: Using safeRestart. In jenkins URL add '/safeRestart' for safely restart the jenkins.

'safeRestart' waits to finish all the running jobs and then it will restarts jenkins.

'/restart' will terminate all the running jobs and it will restart jenkins.

40. What are the steps in while creating Jenkins job?

Answer: Following are the steps in while creating Jenkins Job:

General, Source code Management, Build trigger, Build, Post-Build.

41. How can you decide the no. executors?

Answer: No. Executors is number of parallel jobs, based on requirement If we required, jenkins should run at max 5 parallel jobs we will provided the number at No. executors.

- Manage Jenkins → Configure system → Number of parallel jobs: 5

42. Where did your plugins come from Jenkins?

Answer: Plugins can be downloaded from plugins.jenkins.io site

43. How do you install Jenkins?

Answer: To install Jenkins, we just required to follow these following steps:

1. Install Java Version 8 – Jenkins is a Java based application.

2. Install Apache Tomcat Version 9 – Tomcat is essential to deploy Jenkins war file.

3. Download Jenkins war File – This war is must to install Jenkins.

4. Deploy Jenkins war File – You deploy Jenkins war file using Tomcat to run Jenkins.

5. Install Suggested Plugins – Install a list of plugins suggested by Jenkins.

Once the installation is complete, we will be able to see the Jenkins dashboard.

44. What is Maven? List the benefit of integrating Maven with Jenkins.

Answer: Maven is a build management tool which will use a simple pom.xml for configuring all the dependencies required to build, test and run the code. Maven will manage the full lifecycle of a test project. Once integrated with Jenkins, the maven Webdriver can build the project and will execute all tests efficiently.

45. What you mean by auto layout in Objective C?

Answer: Auto Layout is a new way for defining dynamic GUIs. Before we had autoresizing masks, which can describe how a subview can resize or move when superview is resized. With Auto Layout we will do the same and complicated GUIs also quite easily.

46. Can we access jenkins in command mode?

Answer: We have 'Jenkins CLI' to access jenkins in command mode under 'Manage Jenkins'.

47. How do you create Multibranch Pipeline in Jenkins?

Answer: The Multibranch Pipeline project type will enable us to implement different Jenkinsfiles for different branches of the same project. In a Multibranch Pipeline project, Jenkins will automatically discover, manage and execute Pipelines for branches which will contain a Jenkinsfile in source control.

48. How many environments are there in your current project?

Answer: Dev, Test, Pre-production and Production environment are there in current project

49. Is it possible to exchange jobs between two different jenkins?

Answer: Yes, it is possible by using 'Job Import' plugin. We will exchange the job between two jenkins. We required providing the source instance (Jenkins) URL. We will import a particular job and all the jobs and also views.

50. Explain role-based strategy plugin?

Answer: We will create 'Global roles', 'Project roles' and 'Slave roles' and assigning roles to users by using role-based strategy,

1. **Global roles:** admin, Job creator, anonymous, etc. will allow to set Overall, slave, job, Rn, View and SCM permissions on a global basis.

2. **Project roles:** Allow to set only Job and Run permissions on a project basis.

3. **Slave roles:** Allow to set node-related permissions.

51. Suppose we have two projects in my jenkins both are of different Java versions. How can we build the projects when both projects are of different versions and only one version of java is configured in Global tool configuration?

Answer: We can configure multiple java versions in Global Tool configuration to support different versions of projects.

52. How your jenkins come to know which version to choose when two versions of Java are configured?

Answer: When we will start build jenkins it is asked to choose on which version of Java the build/job has to run.

53. What are Declarative Pipelines in Jenkins?

Answer: Declarative Pipelines are the newest additions to Jenkins which will simplify the groovy syntax of Jenkins pipelines (top-level pipeline) with some exceptions, like:

No semicolon are used as a statement separator. The top-level pipeline can be enclosed within block viz;

The common syntax is:

```
pipeline {

/* Declarative Pipeline */

}
```

Blocks must contain Sections, Directives, steps or assignments.

```
pipeline {

agent any

stages {

stage('Build') {

steps {

// Statements…

}

}

stage ('Test') {

steps {

// Statements…

}

}

}

}
```

The above code has following major elements

1. Pipeline: The block of script contents.

2. Agent: It will define where the pipeline will start running from.

3. Stage: The pipelines will contain several steps enclosed in the block called Stage.

54. How do you change the Jenkins Home Directory?

Answer: First copy entire Jenkins home directory to which directory we required to make a new home directory, next Update that directory in /etc/sysconfig/jenkins file.

55. Which plugin is required to deploy a .war file into application server?

Answer: 'Deploy to Container Plugin' is required to deploy a.war file into application server.

56. How can you send the BIOS a query message directly from the command line?

Answer: In pipeline we can break our jobs out into different stages and we have whatever stage we required to represent the process we use to deploy software and of course, if anything will be wrong, we can see which stage had the problem.

- The important difference between any job and a Pipeline Job is that the Pipeline Scripted job will run on the Jenkins master, in the Freestyle job and everything will be executed in the agent, but for the Scripted Pipeline Job, the pipeline code will be translated in the master to atomic commands which are sent to the agents.

57. What is upstream and downstream projects in jenkins?

Answer: Nothing but dependency of jobs.

- Upstream: if we will set job1 is upstream for job2. Then job1 will be built to build job2.
- Downstream: if we will set job2 is downstream in job1. Then if we can build job1 then automatically job2 can also be build.

> In 'Build triggers' section we will set the upstream projects.

> In 'Post-build' section we will set downstream projects.

58. How to re-execute a parameterized build without entering the parameter value when the job fails?

Answer: 'Rebuild plug-in' will allow the user to rebuild a parametrized build without entering the parameters again.

59. How can you pass parameters from one job to another job?

Answer: When we will create new/existing job go to –> General tab –

–> enable 'This project will be parameterized' –> select 'Run parameter' from drop down –> in project tab select the project name (from which project it will take the parameters).

60. How master system will communicate to slave system?

Answer: using 'slave.jar' file in slave system master system will communicate to slave system. This file must be present in slave to communicate with.

61. How do you make a job to run only on slave node?

Answer: Go to job configuration tab –> under 'General' section enable 'Restrict where this project will be run' then we will provide the lable name of the node.

62. How do you make all the jobs to be run only on slave node?

Answer: Go to –>'Manage jenkins' –> 'configure system' –> at 'Labels' mention the node system label.

63.What is Quite period in Jenkins?

Answer: Time gap between the builds. In seconds.

64. What is SCM Checkout retry count?

Answer: It can check the code in SCM tool twice in between the builds, if the value is 2.

65.How do you delete old builds?

Answer: Delete old builds in following way

go to –> General tab –> enable 'Discard old Builds'

checkbox. Here we will set 'Days to keep builds' and 'Max # of builds to keep'

66. Which plugins you are using?

Answer: Role-based strategy – create global roles, project roles and slave roles.

Backup –

Thin Backup –

Green Ball – Green ball plugin will indicate when job is successfully completed it shows green color instead of blue.

Build Pipeline –

Delivery Pipeline –

Next Build Number –

Rebuild –

SonarQube –

Parameter trigger –

Disk Usage –

67. Plugins website down what will you do and how can you download plugins?

Answer: Usually companies maintain repository for plugins download. If plugins is not maintained in the repository by the companies then we download from any other website and upload in to company's repository and add the plugin.

68. What is difference between Build periodically and Poll SCM?

Answer: Build periodically – It can trigger builds as per the schedule, even if we haven't changed anything.

Answer: Poll SCM – It can check for changes before triggering any build, if there will be any changes to the previous version than only build can be triggered. Otherwise it will not the build.

69. Explain how do create a backup and copy files in Jenkins?

Answer: Jenkins will save all the settings, build artefacts and logs in its home directory so when we will required to create backup of jenkins setup, copy the jenkins home directory and rename the directory.Install the Backup plugin by plugin backup the jenkins.

If we will automate the backup using the thin_backup plugin. Using this plugin, we will schedule the backup.

70. How will you secure Jenkins?

Answer: Following way Jenkins can be secured:

1. Ensure global security is on.

2. Ensure that matrix based is enabled to fine tune access.

3. Automate the process of setting rights/privileges in Jenkins with custom version-controlled script.

4. Limit physical access to Jenkins data/folders.

5. Periodically run security audits on same.

71.Explain how you can deploy a custom build of a core plugin?

Answer: Following are the steps to deploy a custom build of a core plugin:

1. Stop Jenkins.

2. Copy the custom HPI to $Jenkins_Home/plugins.

3. Delete the previously expanded plugin directory.

4. Make an empty file called .hpi.pinned.

5. Start Jenkins.

72. How will you give the customized/custom build numbers in jenkins?

Answer: By using Next_Build_Number plugin, we will change the build number of a jenkins.

- We will only change the build numbers for failed builds, not for success builds.

73. Why we use Nodes in jenkins?

Answer: 1. Single jenkins server will not handle the whole workload of large and heavier build projects.

2. Sometimes we may required several different environments to test the builds. This will not be done by a single server.

74. How do you send the notifications whether build fails or success?

Answer: Method to send the notifications whether build fails or success

1. Configure Email Notification in Configure system under Mange Jenkins.

2. Email notifications will be configured in job configuration under post-build section.

75. How do you manage Users? Explain method to create particular project-based authentication in jenkins.

Answer: MWe will manage users in two ways –

1. Role-based strategy

2. Project based' authentication

create particular project-based authentication in jenkins in following ways

1. We will manage user's permissions. Such as view, create, modify, delete a job and restricting user to access 'Manage Jenkins' and etc.

2. We will Create global roles, such as admin, job creator, anonymous, etc. suc Overall, Slave, Job, Run, View and SCM permissions on a global basis)

3. Create project roles and assign the roles to users.

4. It will restrict user to access particular project/job.

76. How to migrate a jenkins job to new jenkins server?

Answer: To migrate a Jenkins job to new Jenkins server use following process:

1. Copy a job from one jenkins server to another jenkins server corresponding job directory.

2. we should 'reload config' in 'Manage Jenkins'

77. How you can move or copy Jenkins from one server to another?

Answer: Following way we can move or copy Jenkins from one server to another:

1. By copying the related job directory slide a job from one installation of Jenkins to another.

2. By a different name Renaming an existing job by renaming a directory, make a copy of an already existing job by making clone of a job directory.

78. How you can deploy a custom build of a core plugin?

Answer: To deploy a custom field of a core plugin, We do following things:

1. Stop Jenkins

2. Copy the custom HPI to $Jenkins_Home/plugins

3. Delete the previously expanded plugin directory

4. Make an empty file called .hpi.pinned

5. Start Jenkins

79. How you can setup Jenkins jobs?

Answer: Jenkins jobs can be setup by following ways:

1. Select new item from the menu.

2. After that enter a name for the job and select free-style job.

3. Then click OK to create new job in Jenkins.

4. The next page enables to configure job.

80. How can you clone a Git repository via Jenkins?

Answer: If we want to clone a Git repository via Jenkins, we have to enter the e-mail and user name for Jenkins system. Switch into job directory and execute the "git config" command for that.

81. Name two ways a Jenkins node agent can be configured to communicate back with the Jenkins master.

Answer: The tool will provide two mechanisms for starting a Jenkins node agent:

1. Launching a Jenkins node agent from a browser window.

2. Launching a Jenkins node agent from the command line.

When a Jenkins node agent will be launched from a browser, a JNLP file can be downloaded. When it will run, the JNLP file will launche a new process on the client machine which runs Jenkins jobs.

To launch from the command line, the agent.jar file is needed on the client. This executable JAR file will run from the command line, along with a reference to the slave agent's JNLP file which is hosted on the server. Like the JNLP file is downloaded through a web browser, running this command will launche a process on the client which can communicate with the Jenkins master and will run Jenkins build jobs when it will have idle clock cycles.

82. How will you secure Jenkins?

Answer: In following way we can secure Jenkins:

1. Ensure global security is on.

2. Ensure Jenkins is integrated with my company's user directory with appropriate plugin.

3. Ensure matrix/Project matrix is enabled to fine tune access.

4. Automating the process to set rights/privileges in Jenkins with custom version controlled script.

5. Limit physical access to Jenkins data/folders.

6. Periodically run security audits on same.

83. Minimum JRE required for a run of Jenkin 2.1?

Answer: Minimum JRE required for a run of Jenkin 2.1 is JRE8 or Else JDK8

84. Name a Jenkins environment variable you have used in a shell script or batch file.

Answer: There are many environment variables which are available by default in any Jenkins build job.

Following are commonly used ones include:

1. $JOB_NAME
2. $NODE_NAME

3. $WORKSPACE
4. $BUILD_URL
5. $JOB_URL

Note that, as new Jenkins plug-ins will be configured, more environment variables will be available. For example, if the Jenkins Git plug-in will be configured, new Jenkins Git environment variables, such as $GIT_COMMIT and $GIT_URL, become available to be used in scripts.

85. Describe the standard process to configure and use third-party tools within Jenkins?

Answer: The process to use a third-party tool, like Artifactory, Node, SonarQube or Git typically will follow following steps.

1. The third-party software will be installed.

2. A Jenkins plug-in which will support the third-party tool must be installed through the Jenkins admin console.

3. The third-party tool will be configured in the Tools tab of the Manage Jenkins section of the admin console.

4. Finally, the plug-in will be used from within a Jenkins build job. The plug-in can then facilitate communication between the Jenkins build job and the third-party tool.

Every third-party tool is not configured in exactly the same way. For example, Jenkins will be configured to install Maven itself, rather than requiring a pre-existing installation.

Third-party tools, such as Checkstyle or JaCoCo, will be downloaded at build time by Maven. Therefore these above steps are not always adhered to strictly, but these are the typical steps required to install and configure a third-party Jenkins tool.

86. Name three security mechanisms Jenkins uses to authenticate users.

Answer: Jenkins will authenticate users in one of following ways:

1. Jenkins will use an internal database to store user data and credentials. (default method)

2. Jenkins will be configured for authenticating against a Lightweight Directory Access Protocol server.

3. Jenkins will be configured for employing the authentication mechanism used by the application server upon which it will be deployed.

87. How to make sure that your project builds doesn't break in Jenkins?

Answer: We should follow Following steps to make sure that project builds does not break in Jenkins:

1. Perform successful clean install on local machine with all unit tests.

2. Check all code changes.

2. 3. Synchronize with repository to make sure that all required config and POM changes and any difference is checked into the repository.

88. Name three steps or stages a typical Jenkins pipeline might include.

Answer: A full-blown Jenkins pipeline can build a project from source code, put it through a variety of unit, integration, performance and user acceptance tests. Finally, if every test will succeed, deploy a packaged application to an application server, Nexus repository or Docker container.

Therefore, three fundamental stages would be:

1. Build

2. Test

3. Deploy

The best method to implement a Jenkins pipeline as code is to employ many modular steps.

89. How can you temporarily turn off Jenkins security if the administrative users have locked themselves out of the admin console?

Answer: The JENKINS_HOME folder will contain a file named config.xml. When security is enabled, this file will contain an XML element named useSecurity which will be set to true. By changing this setting to false, security can be disabled the next time Jenkins will restarted.

false

Disabling security will always be both a last resort and a temporary measure. Once any authentication problem are resolved, be sure to re-enable Jenkins security and reboot the CI server.

90. Polling a Git repository for new commits is considered a Jenkins anti-pattern. What is a sound alternative to SVN polling?

Answer: Polling a source code management tool such as Git or Subversion to check if a new commit will be issued is a waste of clock cycles and must be avoided.

A good approach is by reversing this process and has the source code tool trigger a Jenkins build when new commit will happen. With GitHub or GitLab, it will be relatively easy to configure a post-commit hook which will run every time a commit is successful. When it is provided with the URL of the Jenkins build, the post-commit hook will easily trigger a Jenkins build, eliminating the required to have Jenkins constantly poll the source code repository.

91. What are Jenkins Installation Requirements?

Answer: Following are Jenkins Installation requirements:

1. Jenkins will require Java7 or above and Servlet 3.1 to function.

2. Java8 can be recommended.

3. Jenkins will require a fair amount of memory to operate well.

Smaller installations will start around 256MB 1GB.

92. Explain how you can deploy a custom build of a core plugin?

Answer: Following are the steps to deploy a custom build of a core plugin:

1. Stop Jenkins.

2. Copy the custom HPI to **$Jenkins_Home/plugins**

3. Delete the previously expanded plugin directory.

4. Make an empty file called **.hpi.pinned.**

5. Start Jenkins.

93. Explain the terms Agent, post-section, Jenkins file?

Answer: 1. Agent– It is directive to inform Jenkins for executing the pipeline in particular manner and order.

2. Post-section– If it is required to add some notification and for performing other tasks at the end of a pipeline, post-section can definitely run at the end of every pipeline's execution.

3. Jenkins file – It will be a text file having the information about Jenkins pipeline and will be checked into source control.

94. What is the programming language used to build Jenkins?

Answer: Java is to build Jenkins which is an open source automation server.

95. What is a Continuous delivery pipeline?

Answer: A continuous delivery pipeline which is an automated expression of process to get software from version control right through to users and customers.

96. Can you write a simple Jenkins Pipeline Code for Java?

Answer: Following is the simple Jenkins Pipeline Code for Java:

Jenkinsfile (Declarative Pipeline)

pipeline {

agent { docker 'maven:3.3.3' }

stages {

stage('build') {

```
steps {

sh 'mvn –version'

}

}

}

}
```

97. What is Declarative Pipeline in Jenkins?

Answer: Declarative Pipeline will be a relatively recent addition to Jenkins Pipeline [1] which will present a more simplified and opinionated syntax on top of the Pipeline sub-systems.

All valid Declarative Pipelines should be enclosed within a pipeline block, for example:

```
pipeline {

/* insert Declarative Pipeline here */

}
```

98.What is the agent directive in Jenkins?

Answer: The agent directive inform Jenkins where and how to execute the Pipeline, or subset thereof.

Underneath the hood, there are following things agent causes to happen:

All the steps which contained within the block are queued for execution by Jenkins. As soon as an executor will be available, the steps can begin to execute.

A workspace will be allocated which contains files checked out from source control and any additional working files for the Pipeline.

99. What are Parameters in Jenkins?

Answer: To support the wide variety of use-cases Pipeline, authors might have the agent section which will support a few different types of parameters. These parameters will be applied at the top-level of the pipeline block, or within each stage directive.

100. What is post?

Answer: The post section defines one or more additional steps which are run upon the completion of a Pipeline's or stage's run which depends on the location of the post section within the Pipeline.

Post will support one of the following post-condition blocks: always, changed, failure, success, unstable, and aborted.

These condition blocks will allow the execution of steps within the post section depending on the completion status of the Pipeline or stage.

101. What are stages?

Answer: Stage – Containing a sequence of one or more stage directives, the stages section is where the bulk of the "work" will be described by a Pipeline can be located. At a minimum it will be recommended which stages will contain at least one stage directive for each discrete part of the continuous delivery process, like Build, Test, and Deploy.

102. What is environment directive?

Answer: The environment directive will specify a sequence of key-value pairs which can be defined as environment variables for the all steps, or stage-specific steps, which depends on where the environment directive will be located within the Pipeline.

103. What are triggers?

Answer: The triggers directive will define the automated ways in which the Pipeline can be re-triggered. For Pipelines which will be integrated with a source like GitHub or BitBucket, triggers will not be necessary as webhooks-based integration can likely already be present. cron, pollSCM and upstream are currently available triggers.

104. What is input directive?

Answer: The input directive on a stage will allow to prompt for input, using the input step. The stage can pause after any options which has been applied, and before entering the stage`s agent or evaluating its when condition. If the input will be approved, the stage can then continue. Any parameters will be provided as part of the input submission can be available in the environment for the rest of the stage.

105. What is Parallel in Jenkins?

Answer: Stages in Declarative Pipeline will declare a number of nested stages within them, which will be executed in parallel. A stage should have only one of either steps or parallel. The nested stages cannot have further parallel stages themselves, otherwise behave the same as any other stage. Any stage which contains parallel cannot have agent or tools, since those are not relevant without steps

106. What is Scripted Pipeline in Jenkins?

Answer: Scripted Pipeline, such as Declarative Pipeline, is built on top of the underlying Pipeline sub-system. Unlike Declarative, Scripted Pipeline will be effectively a general purpose DSL [2] built with Groovy. Most functionality is provided by the Groovy language which is made available to users of Scripted Pipeline, that means it will be a very expressive and flexible tool with which one will author continuous delivery pipelines.

107. What is Flow Control in Jenkins?

Answer: Scripted Pipeline will serially executed from the top of a Jenkinsfile downwards, such as most traditional scripts in Groovy or other languages.

108. What are the software prerequisites that must be met before Jenkins is installed?

Answer: Jenkins with TeamCity:

Comparison	Jenkins	TeamCity
Open Source	Yes	No
Default security	No	Yes
Individual validation	No	Yes
Popularity	Widely used	Not so widely used

The software prerequisites for installing Jenkins are that first we required to install Java Development Kit. We can also required to install the Jakarta Enterprise Edition. Jenkins come along with an embedded Jetty Runtime that can be used if WebSphere or Tomcat is not available.

109. How to turn off Jenkins Security if the administrative users have locked out of the admin console?

Answer: There is a folder which will contain a file named config.xml. We required to change the settings to false for the security should be disabled when Jenkins will be started the next time.

110. Explain the Jenkins tool.

Answer: Jenkins will be thought of as an open-source automation tool which will be used for continuous integration. With Jenkins, we can be able to continuously test software projects therefore that developers can be able to integrate the changes to the project. We will also integrate this automation tool with a large number of testing and deployment technologies.

111. What is the difference between Continuous Integration, Continuous Delivery, and Continuous Deployment?**Answer: Continuous Integration:**

It will Involve keeping the latest copy of the source code at a commonly shared hub where all the developers will check to fetch out the latest change in order to avoid conflict.

Continuous Delivery:

Manual Deployment to Production. It will not involve every change to be deployed.

Continuous Deployment:

Automated Deployment to Production. It will involve every change to be deployed automatically.

112. What are Scripted Pipelines in Jenkins?

Answer: Scripted Pipeline will follow Groovy Syntax as following:

```
Node {

}
```

In the above syntax, the node is a part of the Jenkins distributed mode architecture, where there are two types of node Master that will handle all the tasks in the development environment and the Agent is being used for handling multiple tasks individually.

Question 1. What Is Git?

Answer :

GIT is a distributed version control system and source code management (SCM) system with an emphasis to handle small and large projects with speed and efficiency.

Question 2. What Is A Repository In Git?

Answer :

A repository contains a directory named .git, where git keeps all of its metadata for the repository. The content of the .git directory are private to git.

Question 3. What Is The Command You Can Use To Write A Commit Message?

Answer :

The command that is used to write a commit message is “git commit –a”. The –a on the command line instructs git to commit the new content of all tracked files that have been modified. You can use “git add<file>” before git commit –a if new files need to be committed for the first time.

Question 4. What Is The Difference Between Git And Svn?

Answer :

The difference between GIT and SVN is:

a) Git is less preferred for handling extremely large files or frequently changing binary files while SVN can handle multiple projects stored in the same repository.

b) GIT does not support ‘commits’ across multiple branches or tags. Subversion allows the creation of folders at any location in the repository layout.

c) Gits are unchangeable, while Subversion allows committers to treat a tag as a branch and to create multiple revisions under a tag root.

Question 5. What Are The Advantages Of Using Git?

Answer :

a) Data redundancy and replication

b) High availability

c) Only one.git directory per repository

d) Superior disk utilization and network performance

e) Collaboration friendly

f) Any sort of projects can use GIT

Question 6. What Language Is Used In Git?

Answer :

GIT is fast, and ‘C’ language makes this possible by reducing the overhead of runtimes associated with higher languages.

Question 7. What Is The Function Of ‘git Push’ In Git?

Answer :

‘GIT PUSH’ updates remote refs along with associated objects.

Question 8. Why Git Better Than Subversion?

Answer :

GIT is an open source version control system; it will allow you to run ‘versions’ of a project, which show the changes that were made to the code overtime also it allows you keep the backtrack if necessary and undo those changes. Multiple developers can checkout, and upload changes and each change can then be attributed to a specific developer.

Question 9. What Is “staging Area” Or “index” In Git?

Answer :

Before completing the commits, it can be formatted and reviewed in an intermediate area known as ‘Staging Area’ or ‘Index’.s

Question 10. What Is Git Stash?

Answer :

GIT stash takes the current state of the working directory and index and puts in on the stack for later and gives you back a clean working directory. So in case if you are in the middle of something and need to jump over to the other job, and at the same time you don’t want to lose your current edits then you can use GIT stash.

Question 11. What Is Git Stash Drop?

Answer :

When you are done with the stashed item or want to remove it from the list, run the git ‘stash drop’ command. It will remove the last added stash item by default, and it can also remove a specific item if you include as an argument.

Question 12. How Will You Know In Git If A Branch Has Been Already Merged Into Master?

Answer :

- Git branch merged lists the branches that have been merged into the current branch
- Git branch no merged lists the branches that have not been merged

Question 13. What Is The Function Of Git Clone?

Answer :

The git clone command creates a copy of an existing Git repository. To get the copy of a central repository, ‘cloning’ is the most common way used by programmers.

Question 14. What Is The Function Of ‘git Config’?

Answer :

The ‘git config’ command is a convenient way to set configuration options for your Git installation. Behaviour of a repository, user info, preferences etc. can be defined through this command.

Question 15. What Does Commit Object Contain?

Answer :

a) A set of files, representing the state of a project at a given point of time

b) Reference to parent commit objects

c) An SHAI name, a 40 character string that uniquely identifies the commit object.

Question 16. How Can You Create A Repository In Git?

Answer :

In Git, to create a repository, create a directory for the project if it does not exist, and then run command “git init”. By running this command .git directory will be created in the project directory, the directory does not need to be empty.

Question 17. What Is ‘head’ In Git And How Many Heads Can Be Created In A Repository?

Answer :

A ‘head’ is simply a reference to a commit object. In every repository, there is a default head referred as “Master”. A repository can contain any number of heads.

Question 18. What Is The Purpose Of Branching In Git?

Answer :

The purpose of branching in GIT is that you can create your own branch and jump between those branches. It will allow you to go to your previous work keeping your recent work intact.

Question 19. What Is The Common Branching Pattern In Git?

Answer :

The common way of creating branch in GIT is to maintain one as “Main“ branch and create another branch to implement new features. This pattern is particularly useful when there are multiple developers working on a single project.

Question 20. How Can You Bring A New Feature In The Main Branch?

Answer :

To bring a new feature in the main branch, you can use a command “git merge” or “git pull command”.

Question 21. What Is A ‘conflict’ In Git?

Answer :

A ‘conflict’ arises when the commit that has to be merged has some change in one place, and the current commit also has a change at the same place. Git will not be able to predict which change should take precedence.

Question 22. How Can Conflict In Git Resolved?

Answer :

To resolve the conflict in git, edit the files to fix the conflicting changes and then add the resolved files by running “git add” after that to commit the repaired merge, run “git commit”. Git remembers that you are in the middle of a merger, so it sets the parents of the commit correctly.

Question 23. To Delete A Branch What Is The Command That Is Used?

Answer :

Once your development branch is merged into the main branch, you don't need development branch. To delete a branch use, the command "git branch –d [head]".

Question 24. What Is Another Option For Merging In Git?

Answer :

"Rebasing" is an alternative to merging in git.

Question 25. What Is The Syntax For "rebasing" In Git?

Answer :

The syntax used for rebase is "git rebase [new-commit] "

Question 26. What Is The Difference Between 'git Remote' And 'git Clone'?

Answer :

'git remote add' just creates an entry in your git config that specifies a name for a particular URL. While, 'git clone' creates a new git repository by copying and existing one located at the URI.

Question 27. What Is Git Version Control?

Answer :

With the help of GIT version control, you can track the history of a collection of files and includes the functionality to revert the collection of files to another version. Each version captures a snapshot of the file system at a certain point of time. A collection of files and their complete history are stored in a repository.

Question 28. Mention Some Of The Best Graphical Git Client For Linux?

Answer :

Some of the best GIT client for LINUX is:

a) Git Cola

b) Git-g

c) Smart git

d) Giggle

e) Git GUI

f) qGit

Question 29. What Is Subgit? Why To Use Subgit?

Answer :

'Subgit' is a tool for a smooth, stress-free SVN to Git migration. Subgit is a solution for a company -wide migration from SVN to Git that is:

a) It is much better than git-svn

b) No requirement to change the infrastructure that is already placed

c) Allows to use all git and all sub-version features

d) Provides genuine stress –free migration experience.

Question 30. What Is The Function Of ‘git Diff ’ In Git?

Answer :

‘git diff ’ shows the changes between commits, commit and working tree etc.

Question 31. What Is ‘git Status’ Is Used For?

Answer :

As ‘Git Status’ shows you the difference between the working directory and the index, it is helpful in understanding a git more comprehensively.

Question 32. What Is The Difference Between The ‘git Diff ’and ‘git Status’?

Answer :

‘git diff’ is similar to ‘git status’, but it shows the differences between various commits and also between the working directory and index.

Question 33. What Is The Function Of ‘git Checkout’ In Git?

Answer :

A ‘git checkout’ command is used to update directories or specific files in your working tree with those from another branch without merging it in the whole branch.

Question 34. What Is The Function Of ‘git Rm’?

Answer :

To remove the file from the staging area and also off your disk ‘git rm’ is used.

Question 35. What Is The Function Of ‘git Stash Apply’?

Answer :

When you want to continue working where you have left your work, ‘git stash apply’ command is used to bring back the saved changes onto the working directory.

Question 36. What Is The Use Of ‘git Log’?

Answer :

To find specific commits in your project history- by author, date, content or history ‘git log’ is used.

Question 37. What Is ‘git Add’ Is Used For?

Answer :

‘git add’ adds file changes in your existing directory to your index.

Question 38. What Is The Function Of ‘git Reset’?

Answer :

The function of ‘Git Reset’ is to reset your index as well as the working directory to the state of your last commit.

Question 39. What Is Git Is-tree?

Answer :

‘git Is-tree’ represents a tree object including the mode and the name of each item and the SHA-1 value of the blob or the tree.

Question 40. How Git Instaweb Is Used?

Answer :

‘Git Instaweb’ automatically directs a web browser and runs webserver with an interface into your local repository.

Question 41. What Does ‘hooks’ Consist Of In Git?

Answer :

This directory consists of Shell scripts which are activated after running the corresponding Git commands. For example, git will try to execute the post-commit script after you run a commit.

Question 42. Explain What Is Commit Message?

Answer :

Commit message is a feature of git which appears when you commit a change. Git provides you a text editor where you can enter the modifications made in commits.

Question 43. How Can You Fix A Broken Commit?

Answer :

To fix any broken commit, you will use the command “git commit—amend”. By running this command, you can fix the broken commit message in the editor.

Question 44. Why Is It Advisable To Create An Additional Commit Rather Than Amending An Existing Commit?

Answer :

There are couple of reason

a) The amend operation will destroy the state that was previously saved in a commit. If it's just the commit message being changed then that's not an issue. But if the contents are being amended then chances of eliminating something important remains more.

b) Abusing "git commit- amend" can cause a small commit to grow and acquire unrelated changes.

Question 45. What Is 'bare Repository' In Git?

Answer :

To co-ordinate with the distributed development and developers team, especially when you are working on a project from multiple computers 'Bare Repository' is used. A bare repository comprises of a version history of your code.

Question 46. Name A Few Git Repository Hosting Services:

Answer :

- Pikacode
- Visual Studio Online
- GitHub
- GitEnterprise
- SourceForge.net