Grunt Task Runner

1.Why to use a task runner ?
To create and deploy a front end application,we need to do a lot of tasks such as watching file changes, concatenating or minifying files, linting javascript.Before task runners like Gulp and Grunt existed, these tasks were done using shell or bash scripts on the command line.A task runner automates several repetitive tasks such as minification,compilation,testing and setting up a preview of your AngularJS application.

2.How to achieve minification of Javascript files using Grunt ?

```
module.exports = function (grunt) {
        grunt.initconfig ( {
                meta : {
                        banner : /* This is my minified app, built <% grunt.template.today( ) %> */
                        },
                min : {
                        dist : {
                                src : [ '<banner>' , folderName / *.js ] ,
                                dist : 'fileName.js'
                                },
                        customizedName : {
                                src : [ 'folder1 / file1.js' , 'folder1 / file2.js'],
                                dist : 'fileName.js'
                                }
                        },
                watch : {
                        files : [ 'folderName / *.js ],
                        tasks : [ 'min : dist' ]
                        }
                });
                grunt.registerTask ('default', [ ' min : dist ' ] );
          } ;
```

grunt.initConfig - is used to configure the tasks.Here we are configuring the minification task
dist : is the distribution task.To run this task,use the command min dist
customizedName : is the task name.To run this task,use the command min customizedName
To run all the tasks (here, dist task and customizedName task) , use the command min
meta : is the task,which has a banner task.banner takes the string.To appear this string as the first line in the minified file, add <banner> to src
<% grunt.template.today( ) %> : outputs the current date and time
watch : If we make any changes in a file or add or delete a file then that changes will automatically get reflected in the minified file
grunt.registerTask : Till now, we used the commands like grunt min , grunt watch.But I want to run only grunt command, then we need to configure the commands in the grunt.registerTask

3.How to achieve concatenation of JS files using Grunt ?

```
module.exports = function (grunt) {
        grunt.initConfig ( {
                uglify : {
                        my_target : {
                                files : [ {
```

```
                                src : folderName / *.js ,     // where source JS files reside
                                dest : ‘folerName / ’ ,             // where to generate minified files
                                expand : true ,
                                flat : true,                        // remove unnecessary spaces and new lines
                                ext : ‘.min.js’               // this will be the extension of the minified files
                                } ]
                                }
                         }
                   });
      grunt.loadNpmTasks (‘ grunt-contrib-uglify’) ;    // Load the uglify plugin
      grunt.registerTask (‘default’ , [ ‘uglify’ ] )
      };
```

Automated Javascript Documentation Generators

1.What is an Automated Javascript documentation generator ?
Javascript documentation generator is used for documenting APIs.
API documentation is different from normal documentation because API documentation is generated direct from the source code by reading the comments written in the source code.

2.What are the various Automated Javascript documentation generators availabe ?
The various Automated Javascript documentation generators available are JSDoc , YUIDoc, Doxx, Docco

3.Why do we need an Automated Javascript documentation generator ?
After completing the development of a Javascript project,we need to document it.There might be chances that the code we developed (the code developed for the Javascript project) and the documentation we created may not be in sync and to avoid this problem we make use of Automated Javascript documentation generator.

Code Coverage Tools

1.What is code coverage ?
The idea of code coverage is to record which parts of your code (functions,statements,conditionals) have been executed by your test suite.

2.What are the various code coverage tools available ?
The various code coverage tools available are :
BlanketJS
IstanbulJS

3.What is BlanketJS ?
BlanketJS is a Javascript code coverage tool.

4.What is IstanbulJS ?
IstanbulJS is a Javascript code coverage tool.
IstanbulJS uses Esprima to parse Javascript code and Escodegen to generate the instrumented version of the code.
IstanbulJS is pure Javascript,there is no native code or other kind of native libraries wrapper involved.

5.What is the basic coverage criteria ?
The basic coverage criteria includes :

Function coverage - indicates that each function in the program has been called
Statement coverage - indicates that each statement in the program has been executed
Branch coverage - indicates that each branch of each control structure (such as if and case statements) has been executed
Condition coverage - indicates that each Boolean sub-expression evaluated both to true and false

Javascript Best Practices

1.What are Javascript best practices ?
(i)Always use var to declare variables this prevents polluting the global namespace.Without var the variables will become global variables.
(ii)Declare all the variables at the beginning of the scope even if the variables will not be used yet this prevents lots of error-prone code.
(iii)Use one var declaration for multiple variables.
(iv)Declare each variable on a new line.
(v)use the literal syntax for objects and arrays,that is,use :

```
var obj = { } ;     ( don’t use :   var obj = new object( ) ; )
var arr = [ ]  ;     ( don’t use :   var arr  = new Array( ) ; )
```

(vi)use single quotes ‘ ‘ for strings,that is ,use :

```
var name = ‘petya’ ;   (don’t use  :   var name = “petya” ; )
```

(vii)Never use function declaration in non-function block (non-function blocks such as if,while)
Don’t use this:

```
if ( … ) {
function calc( ) { …}
}
```

use this:

```
if (…) {
var calc = function calc( ) { …}
}
```

(viii)use IIFE to define a scope. Don’t use { } to define a scope
(ix)use = = = and != = .Don’t use = = and !=
(x)use short cuts in conditional statements,that is,use :

```
if ( name) { …}                   (  don’t use  :   if ( name != =  ‘ ‘ ) { …}  )
if ( ! name ) { … }                 (  don’t use  :   if  ( name = = ‘  ‘ ) { … }  )
if ( number.length ) { … }      ( don’t use  :  if ( number.length > 0 )  { …}  )
```

(xi)prevent duplicated identifiers by using function scope or module
(xii)use strict mode
strict mode can be used :
For the whole script : If strict mode is used for the whole scripts,then everything will be in strict mode which is not a good idea,since a third-party script may fail in strict mode.so,it is suggestible to use IIFE and per-function strict mode.
Per-Function

(xiii)start execution of Javascript,when the webpage is ready and to achieve this make use of on-load event.

XSS

1.How does an XSS venerability occur ?
An XSS vulnerability occurs when an input taken from the user is not filtered / santized before it's returned back to the user.

2.What are the different categories of XSS ?
The three different categories of XSS are :
(i)Reflected XSS
(ii)Stored XSS
(iii)DOM Based XSS

3.What is the difference between Reflected XSS, Stored XSS and DOM Based XSS ?
(i)In case of Reflected XSS and Stored XSS , we can see the vulnerability payload in the response page.But in case of DOM based XSS , because the HTML source code and the response of the attack will be exactly the same, we cannot see the vulnerability payload in the response page.But, we can see the vulnerability payload only at runtime or by investigating the DOM of the page.

(ii)One of the biggest differences between Reflected XSS, Stored XSS and DOM Based XSS is that DOM Based XSS cannot be stopped by server-side filters. The reason is quite simple; anything written after the "#" (hash) will never be sent to the server.

(iii)Due to this design anything after the "#" (hash) will never be sent to the server.This means all server-side protection in the code will not work for DOM Based XSS vulnerabilities. As a matter of fact, any other type of web protections such as web application firewalls, or generic framework protections like ASP.NET Request Validation will not protect you against DOM Based XSS attacks.

4.How does DOM Based XSS vulnerability occur ?
A DOM Based XSS vulnerability occurs when a source get's executed as a sink without any sanitization.
A source is commonly referred as any thing that takes input, which apparently in Javascript is "Everything taken from a URL”.
The sources are :
(i)document.URL
(ii)document.documentURI
(iii)location.href
(iv)location.search
(v)location.*
(vi)window.name
(vii)document.referrer
A sink is referred as anything that creates HTML in an insecure way.
The sinks are :
(i)HTML Modification sinks such as “document.write” and “ (element).innerHTML” .
(ii)Execution Related sinks such as “ eval ” , “ setTimeout / setInterval ” , and execScript.

5.How can we detect DOM Based XSS ?
DOM Based XSS can be detected by :
(i)Black Box Fuzzing
(ii)Static Analysis
(iii)Dynamic Analysis.

6.How can we avoid DOM Based XSS ?
We can avoid DOM Based XSS by not using unsafe sinks and by replacing these unsafe sinks with safe methods,such as using innerText in place of innerHTML incase of Javascript. And, using .text( ) in place of .html( ) incase of jQuery.

Static Code Analysis and Dynamic Code Analysis

1.What is Static Code Analysis ?
Static Code Analysis is performed without executing any of the code and is performed to understand the security issues caused within the program code.

2.What is Dynamic Code Analysis ?
Dynamic Code Analysis is performed during execution of the code and is performed to understand the security issues caused by the code's interaction with other system components like SQL databases, application servers or Web services.

3.What are the tools available for Static Code Analysis ?
(i)JSHint
(ii)JSLint
(iii)Esprima

4.What are the tools available for Dynamic Code Analysis ?
(i)BlanketJS
(ii)IstanbulJS

DOM Painting, DOM Reflow

1.What is DOM Painting ?
DOM Painting is a process in which a browser has to convert your DOM and CSS into pixels on the screen, and it does this through a fairly complex process. It starts by reading the markup and from this it creates a DOM tree. It does a similar thing with the CSS and from that it creates the CSSOM. The DOM and CSSOM are then combined and, eventually,a browser creates a structure from which it (browser) starts to paint some pixels.

2.What causes DOM Painting ?
(i)Whenever you scroll up or down in the browser,the browser needs to repaint content before it
(the content) appears onscreen.
(ii)Whenever you hover,click,touch,drag in the browser,the browser needs to repaint content before it
(the content) appears onscreen.

3.What is DOM Reflow ?
DOM Reflow is a process of laying out the entire page again.
DOM Reflow is one of the main causes of slow DOM scripts,

4.What causes DOM Reflow ?
Sone of the causes of DOM Reflow relevant to writing CSS are :
(i)Resizing the window
(ii)Changing the font
(iii)Adding or removing a stylesheet
(iv)Content changes, such as a user typing text in
(v)an input box
(vi)Activation of CSS pseudo classes such as :hover (in IE the activation of the pseudo class of a sibling)
(vii)Manipulating the class attribute
(viii)A script manipulating the DOM
(ix)Calculating offsetWidth and offsetHeight
(x)Setting a property of the style attribute

5.How to minimize DOM Reflow ?
(i)Change classes on the element you wish to style (as low in the dom tree as possible)
(ii)Avoid setting multiple inline styles
(iii)Apply animations to elements that are position fixed or absolute
(iv)Trade smoothness for speed
(v)Avoid tables for layout
(vi)Avoid JavaScript expressions in the CSS (IE only)

Minimize unnecessary DOM Depths in your page
Avoid using unnecessary complex CSS Selectors ,particularly Descendant selectors

AJAX

1.What is AJAX ?
AJAX: Asynchronous JavaScript and XML ,although X stands for XML, JSON is used more than XML as JSON is lighter and a part of Javascript.

2.Why are the advantages of AJAX ?
AJAX can access the information on the server without having to reload the web page,that is,to retrieve or update one small piece of information only that information needs to be passed to and from the server instead of having to redownload the entire web page.
AJAX allows you to send and receive data asynchronously without reloading the web page,that is,AJAX allows you to send only important information to the server not the entire page,so only valuable data from the client side is routed to the server side,this makes the application interactive and faster.

3.How many ways AJAX can access the server and why AJAX uses asynchronous calls to access the server over synchronous calls ?
AJAX can access the server in two ways: using synchronous calls and asynchronous calls
In case of AJAX using synchronous calls,the script stops and waits for the server to send back a reply before continuing.
In case of AJAX using asynchronous calls,the script allows the page to continue to be processed and will handle the reply if and when it arrives.

4.What are the problems faced if AJAX uses asynchronous calls ?
While using AJAX with asynchronous calls,we have to ensure that each AJAX request uses a separate AJAX object rather than reusing the same AJAX object for all AJAX requests.If we use same AJAX object for all AJAX requests then the response handler will only handle the first response that it receives and will disregard any subsequent responses.

5.What are the disadvantages of AJAX ?
1.Due to security constraints,AJAX can only be used to access information from the host that served the initial page.If you need to display(access) information from another server,it is not possible.
that is,By default browsers block AJAX requests across domains.This means,for example,a game on scirra.com can request other pages on scirra.com, but cannot request pages on Facebook.com.This is an important feature of web browsers.
2.Javascript disabled browsers cannot use the AJAX application,that is,AJAX is dependent on JavaScript.
3.Security is less in AJAX applications because all files are downloaded at client side,that is,view source is allowed and anyone can view the code source written for AJAX.
4.Debugging is difficult in AJAX applications.
5.Search engines cannot index an AJAX application.
6.Dynamically created pages created by AJAX does not support back button,clicking the back button or refresh

button will take the user to an entirely different web page or to the beginning of what your dynamic web page was processing.

6.What is Ajax Caching ?
IE is the only major browser that caches XHR requests.An efficient way to avoid this poor behavior is to set an "HTTP response header of Cache-control” to be “no-cache” for every request.This behavior is the default behavior for modern browsers and helps to provide a better experience for IE users.

```
.config(function($httpProvider) {
$httpProvider.defaults.headers.common [‘Cache-Control’] = ‘no-cache’ ;
} );
```

7.What are the disadvantages of AJAX ?
(i)Ajax is dependent on Javascript
(ii)Ajax is difficult to debug
(iii)Ajax source code is readable
(iv)Attackers can insert script into the system

8.What are the properties of XMLHttpRequest ?
The properties of XMLHttpRequest object are:
(i)onReadyStateChange
(ii)readyState
(iii)responseText
(iv)responseXML

9.What are the methods of XMLHttpRequest ?
(i)open( )
(ii)send( )
(iii)setRequestHeader( )

10.What are the types of post back in Ajax ?
The two types of post bak in Ajax are :
(i)Synchronous postback
(ii)Asynchronous postback

12.What are the different ready states of a request in Ajax ?
The five different ready states of a request in Ajax are:
0 - unopened
1- opened
2 - headers received
3 - loading
4 - done

13.What are the different readystates in XMLHttpRequest ?
0 - Ajax Request isn’t initialized
1 - Ajax Request’s server connection established
2 - Ajax Request received
3 - Ajax Request processing
4 - Request finished and response is ready

0 - The request isn’t initialized

1 - The request has been setup
2 - The request has been sent
3 - The request is in process
4 - The request is complete

14.How is encoding handled in Ajax ?
The encoding can be handled in Ajax in two ways :
encodeActionURL( ) - is used for full page refresh
encodeResourceURL( ) - is used for partial page refresh

15.How to create an Ajax object ?
var myObj = new ajaxObject ('http://www.abc.com');

16.What protocols are used by Ajax ?
The protocols used by Ajax for making a request to the server are:
(i)HTTP GET
(ii)HTTP POST
(iii)XMLHttpRequest

17.What formats are used by Ajax ?
Ajax uses either XML or JSON format

18.How to handle concurrent Ajax requests ?
Javascript closures are used to handle the concurrent Ajax requests.
Javascript functions should be written to handle concurrent requests and call back function is passed as a parameter.those parameters are passed to AjaxInteraction(URL,callback) object.

19.How do you know the Ajax request has completed ?
readyState == 4 - tells the Ajax request has completed

20.How to handle forward and back buttons in Ajax ?
Iframes is used to handle forward and back buttons.Iframes allow an HTML document to be embedded in another HTML document.

21.How to control the duration of an Ajax request ?
AsyncPostBackTimeout property is used to control the duration of an Ajax request.

22.What methods are used for cross domain Ajax calls ?
CORS - Cross Origin Resource Sharing which works with HTTP web browsers.
JSONP-JSON with Padding which works with HTTP GET and on legacy browsers.

23.What are the technologies used by Ajax ?
The technologies used by Ajax are as follows :
(i)Javascript
(ii)XMLHttpRequest
(iii)DOM
(iv)XHTML
(v)CSS

24.What is the difference between Ajax and Javascript ?
(i)Ajax sends a request to the server and doesn't wait for the response.Ajax performs other operations during that time.
(ii)Ajax doesn't require the page to refresh for downloading the whole page.
(iii)Ajax minimizes the overload on the server since the script needs to request only once.

25.How can you test an Ajax code ?
JSUnit is used to test an Ajax code.

26.How can we cancel the XMLHttpRequest in Ajax ?
Abort( ) is used to cancel the XMLHttpRequest in Ajax.

27.What is the name of an object used for Ajax request ?
XMLHttpRequest object is used for Ajax request.

28.What is Script Manager ?
Script Manager manages the client side script of Ajax.Every page that uses Ajax has a Script Manager to enable Ajax libraries.

29.How can you handle exceptions in Ajax ?
ErrorTemplate which is the child tag of Script Manager is used to handle exceptions in Ajax.

30.What are the controls of the Script Management group ?
The controls of Script Management group are :
ScriptManager
ScriptManagerProxy

31.What is the difference between proxied and proxy less calls in Ajax ?
Proxied calls are made using stub objects which can be called form PHP classes on the Javascript in Ajax.
Proxyless calls are made using utility Javascript functions like HTML_AJAX.replace( ) and HTML_AJAX.append( ) in Ajax.

32.How many types of ready states in Ajax ?
The ready states in Ajax are :
(i)Intialization
(ii)Request
(iii)Process
(iv)Ready

33.What is a valid type of Trigger in Ajax ?
SyncPostBackTrigger is a valid Trigger in Ajax

34.What is the invalid control in Ajax ?
Control is the invalid control in Ajax

35.What is the name of the DDL that contains Ajax control tool kit ?
The name of the DDL that contains Ajax control tool kit is “Ajaxcontroltoolkit.dll

36.Which property is used to control the duration of Ajax request ?
AsyncPostBackTimeout property is used to control the duration of Ajax request.

37.Which property is used to check whether Ajax request has been completed ?
readystate is used to check whether Ajax request has been completed.

38.Which manager is used for pre-requisite update panel in Ajax ?
Script Manager is used for pre-requisite update panel in Ajax.

39.What are the common Ajax Frameworks ?
The common Ajax Frameworks are :
(i)Dojo Toolkit
(ii)MooTools
(iii)Prototype
(iv)YUI-Yahoo User interface
(v)Spry
(vi)Google Web Toolkit

40.What is comet in Ajax ?
Comet describes an application where the server keeps pushing data to the client instead of having the browser keep polling the server for the fresh content.
Comet is a web application model in which a long-held HTTP request allows a web server to push data to a browser,without the browser explicitly requesting it.

41.Which Ajax request is used for retrieving data when the data will not change for a given request URL ?
Ajax request should use HTTP GET for retrieving data when the data will not change for a given request URL.

42.Can we send Ajax request to another domain ?
We can’t send an Ajax request to another domain

43.What type of response we can get in Ajax Response ?
(i)Text data
(ii)HTMLl data
(iii)JSON data
(iv)XML data

44.How to handle concurrent Ajax requests ?
Javascript closures is used to handle concurrent Ajax requests.

45.How to set all Ajax calls synchronous ?
$.ajaxSetup ({async : false })

46.How to do synchronous Ajax requests ?
Set "async = false" to do synchronous Ajax requests

API Authorization

1.What is the difference between Authentication and Authorization ?
Authentication = login + password (who you are)
Authorization = permissions (what you are allowed to do)

2.What is an OpenId and OAuth ?
OpenID is about authentication to many sites with one username.
OAuth is about authorization ,that is,site A has permission to call site B's api.

3.What is SSO (Single-SignOn) ?
SSO (Single-SignOn) allows a user to use one set of login credentials (that is,userName and password) to access multiple applications.

4.How many ways can we implement SSO ?
SSO can be implemented by using the following web security protocols :
(i)SAML (Security Assertion Markup Language)
(ii)OpenID (The latest version of OpenID is OpenIDConnect,which is a combination of both OpenID authentication and OAuth authorization.
(iii)OAuth (Open Authorization)

5.When to use SAML,OpenID and OAuth ?
(i)If the use case is to develop SSO where at least one partner is enterprise then use SAML, otherwise use OpenID.
(ii)If the use case involves mobile devices for API authorization then use OAuth.
(iii)If the use case requires a centralized identity provider then use SAML.
https://spin.atomicobject.com/2016/05/30/openid-oauth-saml/

Page Optimization Techniques

1.How to increase the page performance or How to achieve page optimization or How to load the page faster
(i)Make fewer HTTP request
(ii)Use a CDN
(iii)Add an Expires header
(iv)Grip components
(v)Put CSS at the top
(vi)Move JS to the bottom
(vii)Avoid CSS expressions
(viii)Make JS and CSS external
(ix)Reduce DNS lookups
(x)Minify JS
(xi)Avoid redirects
(xii)Remove duplicate scripts
(xiii)Turn off ETags
(xiv)Make Ajax cachable and small

JSON

1.What is JSON ?
JSON is often used to retrieve data from server with AJAX.

JSON is a non-strict subset of the Javascript,that is JSON allows some unescaped characters in strings that are illegal in Javascript and ECMAScript strings.
JSON is an object useful for exchanging data between client and server

2.What is JSON.parse( ) ?
JSON.parse (text [, reviver])
JSON.parse( ) method is used to convert a JSON string to an object.
text - is a valid JSON string
reviver - is optional.is a function that transforms the results.This function is called for each member of the object.If a member contains nested objects,the nested objects are transformed before the parent object.For each member,the following occurs :
If reviver returns a valid value,the member value is replaced with the transformed value.
If reviver returns the same value it received,the member value is not modified.
If reviver returns null or undefined,the member is deleted.

3.Why do we prefer parse( ) instead eval( ) to parse JSON data ?
When parsing JSON (that is,JSON data) using the eval( ) function,there are some Unicode characters that are valid in JSON strings but invalid in Javascript,so additional escaping may be needed in some cases.
JSON.parse( ) is developed as a safer alternative to Javascript's eval( ).
JSON.parse( ) is specifically intended to process JSON data and not Javascript.
JSON.parse( ) accepts only JSON,preventing potentially malicious code from being inadvertently executed.

4.What are the basic types in JSON ?
JSON basic types are:
(i) Number
(ii) Null
(iii)String
(iv)Boolean
(v) Array
(vi)Object

5.What is JSON.stringify( ) ?
JSON.stringify(value[, replacer[, space]])
JSON.stringify( ) method converts a Javascript value to a JSON string,optionally replacing values if a "replacer function" is specified,or optionally including only the specified properties if a "replacer array" is specified.
value - is a Javascript value,usually an object or array,to be converted
replacer - is optional.is a function or array that transforms the results.
space - is optional.It adds indentation,white space,and line break characters to the return-value JSON text to make it easier to read.

6.What is jQuery.parseJSON (json) ?
jQuery.parseJSON (json) takes a well-formed JSON string and returns the resulting Javascript value.
jQuery.parseJSON (json) throws an error, if empty string,null, or undefined is passed.
Prior to jQuery 1.9, jQuery.parseJSON (json) returns null instead of throwing an error,if empty string, null, or undefined is passed.

7.How can we validate JSON data ?
To validate JSON data,we use JSONLint (that is,if your JSON data isn't properly formatted,it can cause errors, so it is better to validate your JSON data beforehand)

LOAD BALANCING STRATEGIES

1.What are load balancing strategies ?
The two load balancing strategies are :
(i)Round Robin Forwarding : Sends the first event to one indexer and the subsequent event to another indexer (that is,incase of Round Robin consecutive events are alternately forwarded to alternate indexers)
(ii)Auto Load Balancing Forwarding : Intelligently distributes or forwards events across multiple indexers

2.How to detect event boundaries ?
Round Robin cannot deal with uncooked data (that is,it cannot detect the event boundaries).Whereas,Auto Load Balancing deals with uncooked data (that is,it can detect the event boundaries).

3.Can we use Round Robin for light weight forwarders ?
We can't use Round Robin for light weight forwarders because incase of light weight forwarders the event boundaries are not discovered

4.Is Round Robin a blocking ?
Yes,Round Robin is a blocking because Round Robin keeps attempting to send the events to the indexer even when the indexer goes down and doesn't automatically switch the events to the other indexer.Whereas,Auto Load Balancing is a non-blocking.

5.which load balancer did you use ?
we used Barracuda load balancer for scalability and high availability of server network.

Maven

1.What is Maven ?
Maven is used for building and managing a Java Project,that is,Maven is a Java Application Management tool.

2.What tool was used before Maven ?
Ant Build Automation Tool was used before Maven.The problem with Ant is,we need to manually write the code or script to do the activities like :
(i)Managing the dependencies of a project
(ii)Building the WAR File
(iii)Deploying the WAR File to Tomcat Server

3.What is POM (Project Object Model) ?
POM (Project Object Model) is an XML file that contains information about project and configuration details used by Maven to build a Java project.

4.Explain pom.xml file ?
pom.xml file contains :
(i)Name - name of the project (Name is a combination of <groupId> and <artifactId>)
(ii)Version - version number of the project
(iii)Packaging - can be a JAR,WAR, or EAR
(iv)Dependencies - can be Hibernate,JUnit, or Spring
(v)Plugins - can be Maven-Tomcat

5.How to exclude a dependency using pom.xml ?
In pom.xml ,place the dependency name you want to exclude in the <exclusions> tag.

6.How do you specify the range of version of a dependency using pom.xml ?
(i)<version> [,4.1) </version> - installs any version less than 4.1

(ii)<version> [,4.1] </version> - installs any version less than or equal to 4.1

(iii)<version> [4.1,] </version> - installs any version greater than 4.1

(iv)<version> [4.1,4.4] </version> - installs any version between 4.1 and 4.4

7.What is <scope> in pom.xml ?
<scope> tag defines where the library should be used in our project.
For an instance:
<scope>test</scope> - tells that the library can be used only for the unit testing not for the source code.

8.What does <source>and <target> tags define in pom.xml ?
<source> and <target> tags define which version of Java to be used.
For an instance:
<source>1.8</source>
<target>1.8</target>
defines 1.8 version of Java to be used.

9.What is the folder structure suggested by Maven and how do you compile the Java source files and Java test files ?
step1 :
The default folder structure that Maven makes you to follow is :
(i)src / main / java
(ii)src / test / java
(iii)pom.xml

step2 :
(i)mvn compile
creates a "target" folder ("target" folder contains "classes folder" and "maven-status folder".The "classes folder" contains the "source class file")
Note : mvn compile - is used to compile the Java source files

(ii)mvn test-compile
creates a "test-classes" folder inside "target" folder ("test-classes" folder contains the "test class file")
Note : mvn test-compile - is used to compile the Java test files

(iii)mvn test
compiles the Java source files,compiles the Java test files, and runs the JUnit test cases.

10.How can you do dependency management at a single place(that is,in a Parent POM) in a multi module project ?
In a multi module project,if different modules are using the same dependencies,then you can manage the dependencies at a single place,that is ,in a parent POM.
For an instance,if different modules are using the spring dependency and you want to upgrade the version of spring,then you should be able to do that with one change.

11.Which command is used to deploy your project on Tomcat server ?
mvn tomcat7 :run - is used to deploy the WAR,JAR or EAR on Tomcat server.
mvn tomcat7 :run - this command downloads the Tomcat server, deploys the application in Tomcat and runs the

application.

mvn clean - is used to delete the “target” folder.

mvn install - creates the JAR File

Maven Archetypes is used to generate web applications.

WEB ACCESSIBILITY

1.What is Web Accessibility ?
Web Accessibility means people with disabilities can perceive, understand, navigate and interact with the web.

2.How can you measure Web Accessibility ?
Web Accessibility can be measured by WCAG (Web Content Accessibility Guidelines)

3.What tool can be used for testing the issues while implementing the web accessibility ?
AMP (Accessibility Management Platform) can be used for testing the issues while implementing the web accessibility

4.How can you make your website accessible with the keyboard ?
Writing tabindex=0 on an element (when you write tabindex=0,then the focus will go to that particular element) and also you need to make use of native HTML controls like < a href > , <button> , <input> instead of custom controls like <span> or <div> with custom event handlers as native controls are generally accessible already

5.What is voice over ?
voice over will read the content you write in alt attribute for an <img> tag,similarly voice over will read the content you write in the aria-label attribute for an <a> tag.
aria -haspopup = true , says there is a popup
aria -haspopup = false , says there is no popup
aria - expanded = true , says expanded if the drop down is expanded , says collapsed if the drop down is not expanded

6.What challenges did you face with web accessibility ?
(i)Incase of a modal pop-up,the focus is always inside the modal pop-up,but according to WCAG the focus should go to the browser control and come back to modal pop-up.To achieve this, we have to hack the Bootstrap’s default behavior
(ii)It is very difficult for us to hack the default behavior of the third party plugins because we should never edit the third party plugins.
(iii)con sequencing problems,that means if the website is accessible with the keyboard then it may not be accessible with the mouse sometimes,this is called as con sequencing problem
(iv)when we fix something in chrome it may not work in safari exactly
(v)Regression issues,that is , when we fix something in the HTML it might break other part of the HTML like child or parent then we have to work again fixing that part

7.What screen readers have you used ?
NVDA (Non Visual Desktop Access) is an open source screen reader for the windows operating system
NVDA :
H - To read Heading

K - To read link
G - To read image
L - To read list
I - To read an item in a list
T - To read a table
Ins - To stop the screen reader
Ins + Down arrow - To continue reading

8.What is a role attribute ?
role attribute conveys the role of an element
For an instance,we can mention the role of an <a> tag as “link”

9.What aria (Accessible Rich Internet Applications) attributes have you used ?
aria - autocomplete - Indicates whether user input completion suggestions are provided and used for the text boxes and combo boxes
aria - checked - Indicates the current checked state of check boxes and radio buttons
aria - disabled - used to disable an element
aria-expanded - Indicate whether an element is currently expanded or collapsed
aria - haspopup - Indicate whether the element has a popup
aria - hidden - used to hide an element
aria - level - Indicate the hierarchical level of an element within a structure
aria - pressed - Indicate the current pressed state of a button
aria- label - defines a string that labels the current element
aria- selected - Indicates the current selected state of various widgets
aria- setsize - defines the number of items in list

TRANSPILING

1.What is Transpiling ?
Transpiling is simply the process of taking one language and converting it to another language. If you've used Sass to streamline your CSS workflow, you have transpiled Sass code in the CSS.

2.What is the difference between compiling and transpiling ?
Transpiling is a special kind of compiling.
Compiling is a process of taking one language and converting it to another language.
Transpiling is a process of taking one language and converting it to another language that has a similar level of abstraction.

3.What is Babel ?
Babel is a Javascript transpiler.
Babel takes the ES6 code and JSX and compiles it into ES5 code.ES5 code is understandable by all the browsers.

4.What are the tools used for transpiling ES6 to ES5 ?
The two main tools used for transpiling ES6 to ES5 are Babel and Traceur.

5.What is the difference between Babel and Traceur ?
Babel has an online REPL.
Tracer is not compliant with ReactJS
Babel doesn't need any runtime extra script to run. Everything is done server-side. You just have to execute a compilation task once, and then deploy the compiled sources. At the opposite, Traceur needs to embed such a script,

bringing an extra overhead

6.What is a web pack ?
web pack is a node module that compiles NodeJs code into browser executable Javascript.
web pack is a module bundler.
With web pack,you can easily split your application into multiple files.For an instance,your codebase can be split into multiple chunks and those chunks can be loaded on demand reducing the initial loading time of your application.

7.what is a webpack-dev-server ?
webpack-dev-server allows a browser to reload whenever you change your code.

8.what do the *-loader modules do ?
*-loader modules tell web pack how to compile A into B so the browser can read it.For an instance,if we use LESS then web pack will translate that into CSS.

MODULE PATTERNS

1.Why do we use Module patterns ?
Module patterns have come into the picture to overcome (the fear of global state)the fear of using the global variables.

2.What is Bundling ?
Bundling is the process of concatenating multiple JS files into one JS file

3.Why do we use Bundlers ?
With Bundlers,we can achieve dynamic loading during development and bundled loading during production.
Bundlers use topological sorting algorithm internally,this algorithm resolves the order in which things need to execute in order for their dependencies to be resolved before they execute.This states that

4..What are the different bundlers available ?
Browserify, web pack,Esperanto and Rollup

5.What are different module systems you used ?
(i)Globals modules or Global module pattern
(ii)AMD (Asynchronous Module Definition) modules or AMD module pattern or AMD module system
(iii)CommonJS modules or CommonJS module pattern or CommonJS module system
(iii)ES6 modules or ES6 module pattern or ES6 module system

6.How did you achieve bundling when you used Globals modules or Globals module pattern ?
using cat command (cat command available in shell scripting)

7.How did you achieve bundling when you used AMD modules or AMD module pattern ?
we can use any one of the bundlers :
Browserify, web pack,Esperanto or Rollup

8.How did you achieve bundling when you used CommonJS modules or CommonJS module pattern ?
we can use any one of the bundlers :
Browserify, web pack,Esperanto or Rollup

9.How did you achieve bundling when you used ES6 modules or ES6 module pattern ?
we can use any one of the bundlers :
Browserify, web pack,Esperanto or Rollup

10.What are the different JS module loaders you used ?
RequireJS - RequireJS is used to load AMD modules
BrowserifyJS - BrowserifyJS is used to load commonJS modules
SystemJS - SystemJS is used to load AMD modules,CommonJS modules and ES6 modules.

11.What is systemJS ?
systemJS is used for module loading.systemJS can load all types of modules, that is,AMD modules,CommonJS modules and ES6 modules.When systemJS loads the ES6 modules,systemJS compiles ES6 modules into ES5 in the browser

12.Explain commonJS module pattern ?
CommonJS module pattern consists of a require function to load the modules and an export function to define the modules.
CommonJS module pattern is used on server side.
CommonJS module pattern is designed for synchronous loading of modules.
CommonJS module pattern supports circular dependency (that is,cyclic dependency)

13.What is the disadvantage of CommonJS module pattern ?
The main disadvantage of CommonJS module pattern is that the modules are loaded synchronously

14.Explain AMD module pattern ?
To create a module in AMD module pattern,we call a define method with three parameters :
(i)identifier of the module
(ii)dependencies of the module
(iii)factory function which returns the module
To utilize a module in AMD module pattern,we call a require method that takes the dependencies of the script and the script itself.
AMD module pattern is used on client-side
AMD module pattern is designed for asynchronous loading

15.Explain ES6 module pattern ?
ES6 module pattern is designed to support both synchronous and asynchronous loading.
ES6 module pattern supports circular dependency (that is,cyclic dependency).

ASYNC

async.parallel
I need to run multiple tasks that doesn’t depend on each other and when they all finish do something else

async.series
I need to run multiple tasks that depends on each other and when they all finish do something else

async.forEach
I need to iterate over a collection, perform an asynchronous task for each item, and when they're all done do something else

async.forEachLimit
I need to iterate over a collection, perform an asynchronous task for each item, but only let x tasks run at the same time, and when they're all done do something else

async.forEachSeries
I need to iterate over a collection, perform an asynchronous task for one item at a time, and when they're all done do something else

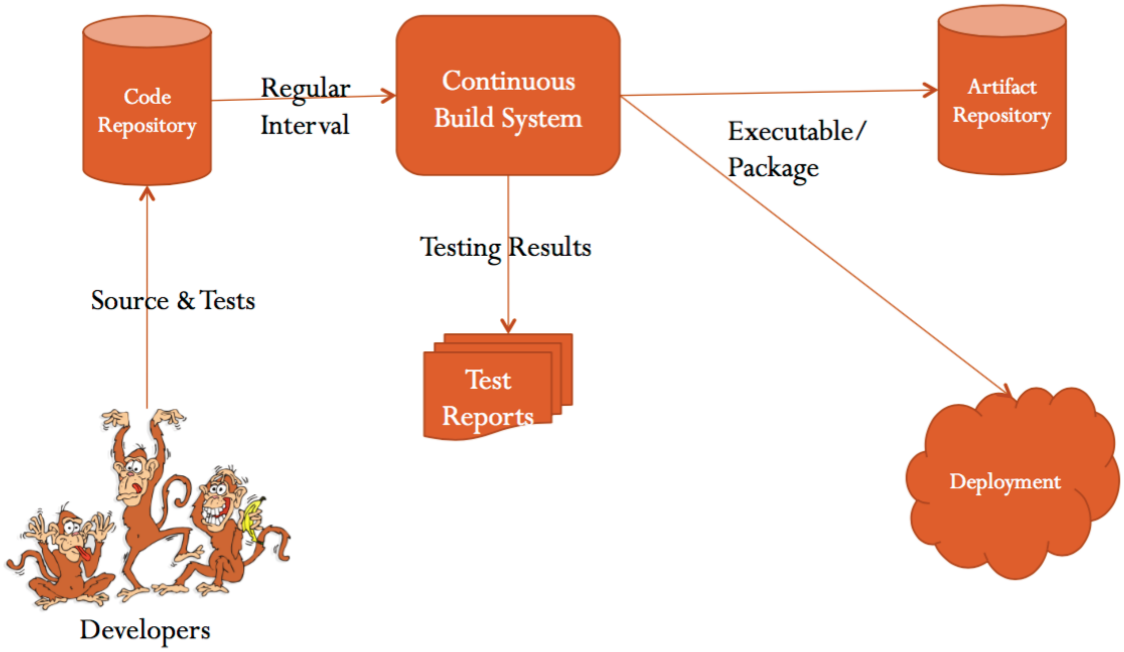
async.queue
I need to perform an arbitrary set of asynchronous tasks

use combination of all three : async.parallel, async.series, async.forEach
I need to perform some parallel tasks, some serial tasks and iterate over a collection performing an asynchronous task for each item

Continous Integration

1.What is Continuous Integration ?
Continuous Integration is a software development practice where members of a team integrate their work frequently, usually each person integrates at least daily - leading to multiple integrations per day.Each integration is verified by an automated build (including test) to detect integration error as quickly as possible.
In other words,at every commit the system is :
(i)integrated - that is,all changes up until that point are combined into the project.
(ii)built - that is,the code is compiled into an executable or package.
(iii)tested - that is,automated test suites are run.
(iv)archived - that is,versioned and stored so it can be distributed as is, if desired.
(v)deployed - that is,loaded onto a system where the developers can interact with it.

CI - Workflow



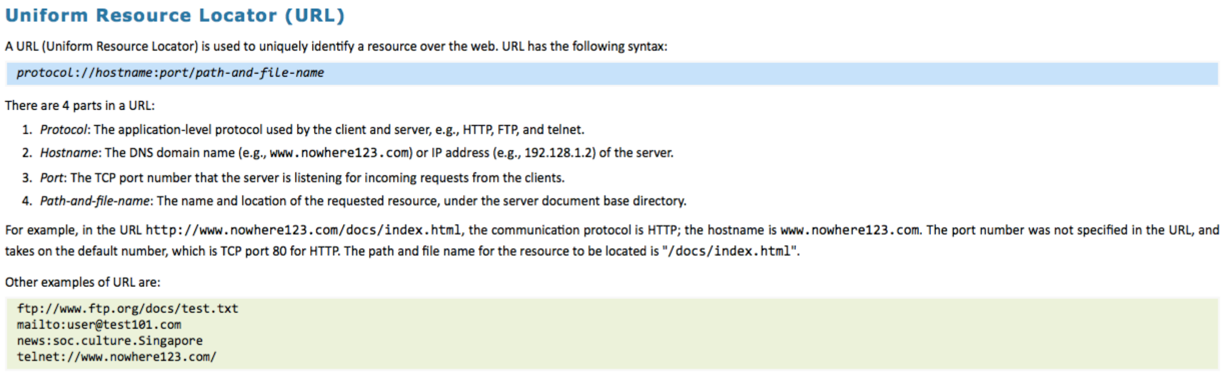
2.What are CI tools ?
The CI tools are :
(i)Version Control Systems like GIT,SVN and Mercurial
(ii)Continuous Build Systems like Jenkins,Bamboo,Cruise Control
(iii)Test Frameworks like Unit,Cucumber,CppUnit
(iv)Artifact Repositories like Nexus, Artifactory, Archiva

3.What can Jenkins do ?
(i)Integrate with many different Version Control Systems
(ii)Generate test reports
(iii)Notify stakeholders of build status
(iv)Push to various artifact repositories
(v)Deploy directly to production or test environments

4.What are the Build Automation Tools available ?
(i)Makefiles
(ii)Shell Scripts
(iii)Ant
(iv)Maven
(v)Gradle

URL

Uniform Resource Locator (URL)



Uniform Resource Locator (URL)

A URL (Uniform Resource Locator) is used to uniquely identify a resource over the web. URL has the following syntax:

```
protocol://hostname:port/path-and-file-name
```

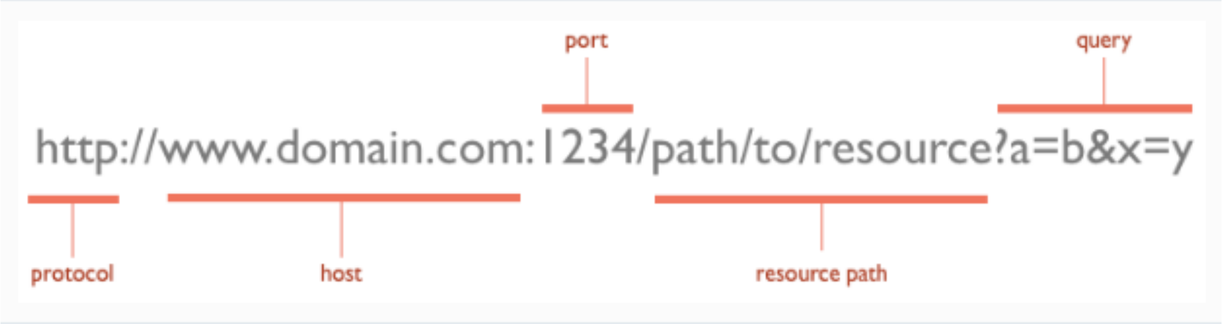
There are 4 parts in a URL:

1. *Protocol*: The application-level protocol used by the client and server, e.g., HTTP, FTP, and telnet.
2. *Hostname*: The DNS domain name (e.g., `www.nowhere123.com`) or IP address (e.g., 192.128.1.2) of the server.
3. *Port*: The TCP port number that the server is listening for incoming requests from the clients.
4. *Path-and-file-name*: The name and location of the requested resource, under the server document base directory.

For example, in the URL `http://www.nowhere123.com/docs/index.html`, the communication protocol is HTTP; the hostname is `www.nowhere123.com`. The port number was not specified in the URL, and takes on the default number, which is TCP port 80 for HTTP. The path and file name for the resource to be located is "`/docs/index.html`".

Other examples of URL are:

```
ftp://www.ftp.org/docs/test.txt
mailto:user@test101.com
news:soc.culture.Singapore
telnet://www.nowhere123.com/
```



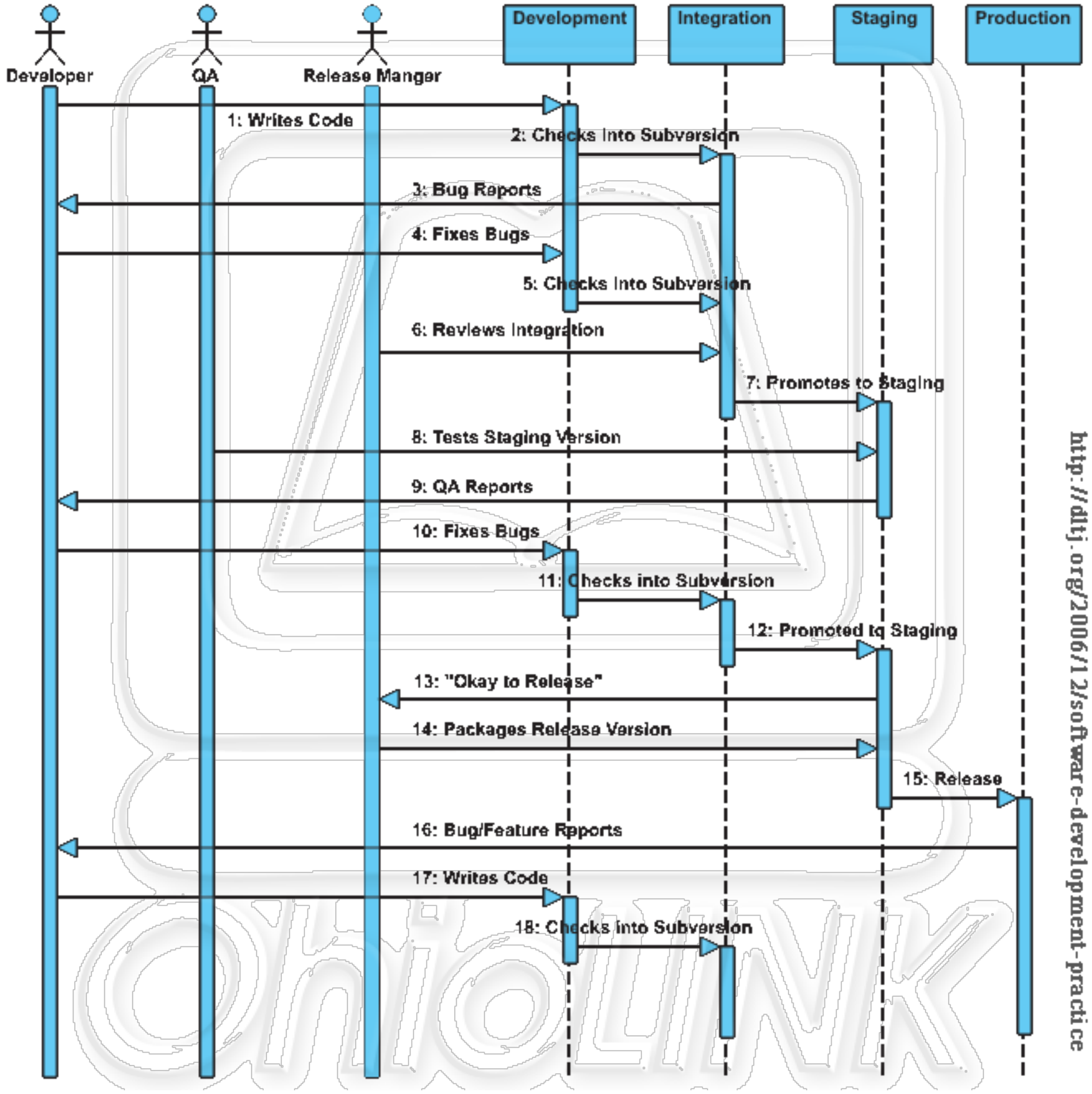
Deployment Process

1.Explain Deployment process ?

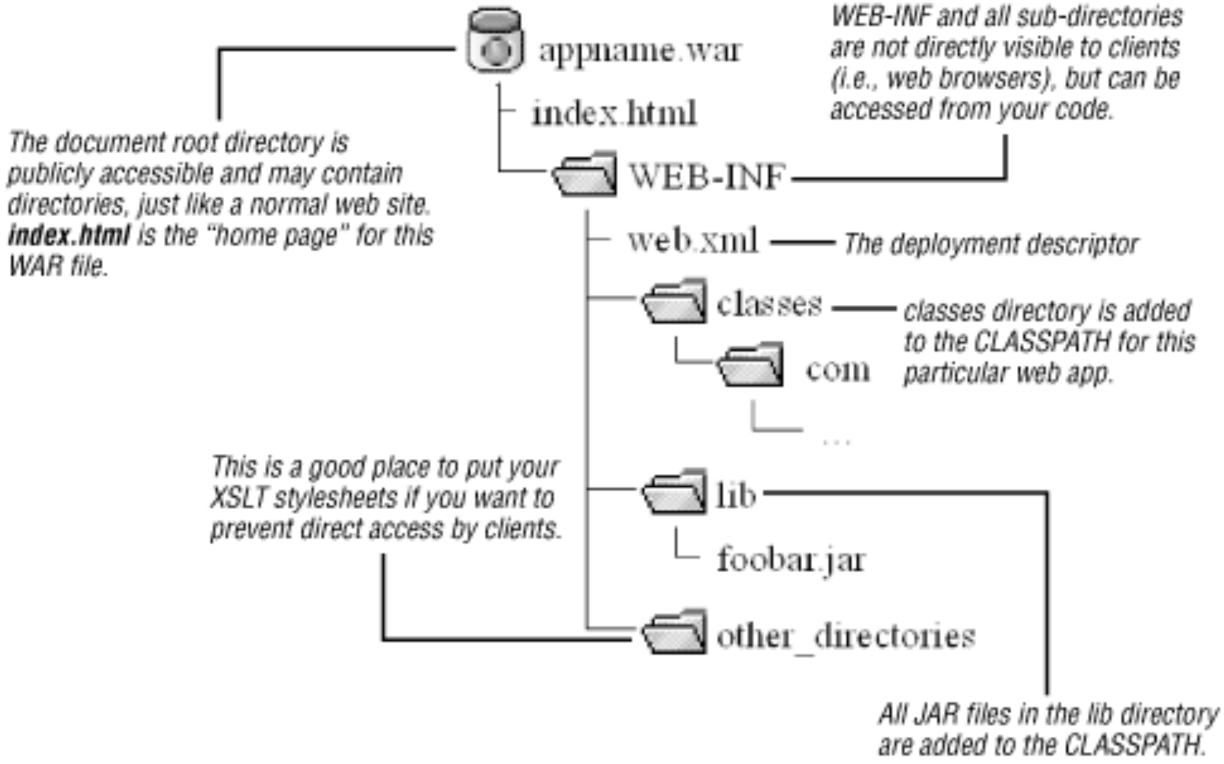
I as a developer develop the code in Development environment and check the code into GIT (source code repository, that is into Integration environment)

Other developers report bugs and I make changes to my code (basically,fixing the bugs reported by other developers) and check the code into GIT (source code repository, that is into Integration environment).

If all the developers are happy with the behavior of the Integration environment,then the Release Manager creates a copy or “Tag” of the code in the GIT and updates the Staging environment to this Tag.At this point,QA’s start their review and reports bugs to me and I fix the bugs and check the code into GIT (source code repository, that is into Integration environment).If QA’s are happy with the behavior of the Staging environment,then the Release Manager creates a copy or “Tag” of the code in the GIT and updates the Staging environment to this Tag,and the Release Manager packages up the release version from GIT and deploys it on the Production servers.As time goes on,bug reports and feature requests are made and I develop the code in Development environment and check the code into GIT (source code repository, that is into Integration environment) and the same above process continues.



2.What is the standard structure of a WAR File ?


appname.war
index.html
WEB-INF
web.xml
classes
com
...
lib
foobar.jar
other_directories
WEB-INF and all sub-directories are not directly visible to clients (i.e., web browsers), but can be accessed from your code.
The document root directory is publicly accessible and may contain directories, just like a normal web site. index.html is the "home page" for this WAR file.
The deployment descriptor
classes directory is added to the CLASSPATH for this particular web app.
This is a good place to put your XSLT stylesheets if you want to prevent direct access by clients.
All JAR files in the lib directory are added to the CLASSPATH.

index.html is the home page for your web application
web.xml (that is,Web Deployment Descriptor File)contains information about the deployment.
classes filter can contain Domain classes,Business classes, DAO classes
lib folder can contain Hibernate JARs , Web Services related JARs
META-INF folder is under appname.war
META-INF folder has METAFILE.MF (METAFILE.MF is used to mention the order of the classes to be executed)

3.How to deploy WAR file on Apache Tomcat web server ?
(i)Edit tomcat-users.xml file , to configure the userName , Password and Roles
(ii)Start the Apache Tomcat web server
(iii)Go to localhost:8080
(iv)Click Manager App, which pops up a window where you can enter the userName and Password (that is, the userName and Password that you configured in tomcat-users.xml file) and click Login
(v)Go to "Deploy" section,where you can see "WAR file to Deploy" section and click "ChooseFile" and then click "Deploy".

4.What is the difference between JAR , WAR and EAR files ?
JAR (Java Archive) File - A JAR File contains a group of .class Files
WAR (Web Archive) File - A WAR File contains HTML, CSS and Javascript files.A WAR File represents a Web Application.
EAR (Enterprise Archive) File - An EAR File contains HTML,CSS and Javascript Files along with EJB's and JMS. An EAR File represents an Enterprise Application.

5.How to create an Executable JAR File ?
To create an Executable JAR File, follow the two steps :
jar -cvfm <DesiredJARFileName> <ManifestFileName.MF> <ClassName1.class> <ClassName2.class> _ _ _ <ClassNameN.class>

dir   <JARFileName>
To run the Executable JAR File :
Type the command : java -jar <JARFileName>   or double click the JAR File

HTTP

1.What is the evolution of HTTP ?
1999 -HTTP/1.1
2000 -REST
2005 -AJAX
2006 -SSE (Server-Sent Events)
2008 -Web socket
2009 -SPDY
2015 -HTTP/2

2.What is Polling ?
In this case the Browser opens a connection before sending a request to the server and this connection is closed when the server sends a response to the Browser,that is,the Browser has to open the connection every time it sends a request to the server.
In this case the Browser sends the requests to the server at regular intervals and server immediately sends responses to the Browser even when no notifications is received by the server.This leads to unnecessary opening and closing of connections.

3.What is long-polling ?
In this case also the Browser opens a connection before sending a request to the server and this connection is closed when the server sends a response to the Browser,that is,the Browser has to open the connection every time it sends a request to the server.
In this case the Browser sends a request to the server and the server keeps the request open for a set period of time.If any notification is received by the server within that set period of time,then server sends a response to the Browser.If no notification is received by the server within the set period of time,then server sends a response to the Browser to terminate the open request.

4.What is streaming ?
Browser sends a request to the server and the server keeps the request open forever.
If any notification is received by the server then only the server sends the response to the Browser.

5.What is HTTP Framing ?
HTTP Framing means HTTP contains three sections they are :Header,Payload and Footer

6.What is Web socket Framing?
Web socket Framing means Web socket contains only one section that is payload and this is the reason the Web socket has lean Framing.

7.What is a Web socket used for ?
A Web socket is used for facilitating live content and the creation of real-time games.

8.What are the different events dispatched by a Web socket ?
The different events dispatched by a Web socket are :
(i)on open

(ii)on message
(iii)on close
(iv)on error

9.What is Half duplex communication ?
one / single directional communication is known as Half duplex communication.
HTTP has Half duplex communication.

10.What is Full duplex communication ?
Two / Bi directional communication is known as Full duplex communication.
Web socket has Full duplex communication.

11.What are the different states of a Web socket readystate ?
The different states of a Web socket readystate are :
(i)0 - connecting
(ii)1 - open
(iii)2 - closing
(iv)3 - closed

12.What are the steps included in establishing a Full-Duplex communication ?
The five steps included in establishing a Full-Duplex communication are:
(i)client sends request for protocol upgrade to server
(ii)server sends response and confirms protocol upgrade
(iii)websocket.readyState == 1
(iv)websocket message from client to server always MASKED
(v)echo message from server

HTTP/2

1.What is a Protocol ?
Protocols usually consist of three main parts: Header, Payload and Footer. The Header placed before the Payload contains information such as source and destination addresses as well as other details (such as size and type) regarding the Payload. Payload is the actual information transmitted using the protocol. The Footer follows the Payload and works as a control field to route client-server requests to the intended recipients along with the Header to ensure the Payload data is transmitted free of errors.
The system is similar to the post mail service. The letter (Payload) is inserted into an envelope (Header) with destination address written on it and sealed with glue and postage stamp (Footer) before it is dispatched.

2.What are the features of HTTP/2 ?
The features of HTTP/2 are:
(i)Multiplexing
In HTTP/1.1, the connections are persistent which allows multiple requests to be sent or pipelined over the same TCP connection.
Even though multiple requests are sent or pipelined over the same TCP connection,the responses arrive synchronously in the same order as they are requested.This means an expensive resources (For an instance,loading a large image file) will block a lightweight response if requested in the wrong order.This phenomenon is known as head-of-line (HOL) blocking.
HTTP/2 allows multiplexing that solves HOL issue by allowing responses to be arrived out of order,thus eliminating the need to open multiple connections.

(ii)Compression and Metadata to reduce Header Overhead
In HTTP/1.1,each and every request and response has a header typically between 200 bytes to 2kb in size.The first issue is that the headers are not permitted to be compressed.The second issue is that the headers contain a lot of redundant information that is exchanged several hundred times as browsers make as many requests to load a webpage.Static headers like Accept* and User-Agent only need to be exchanged once.
HTTP/2 fixes these two issues by compressing and eliminating unnecessary headers.

(iii)Server Push
In HTTP/1.1,web browser requests a webpage,waits for it (webpage) to be downloaded,parses it(webpage) to find all linked assets such as CSS and Javascript and then makes separate requests to download these assets.
HTTP/2 Server Push allows the server to send all these files to the web browser proactively with the first request knowing that the web browser will need these files to display the webpage.I think there is a way to tell the server to stop being so proactive if the web browser already has these files in its cache.

(iv)Binary Encoding
In HTTP/1.1,the protocols are text based and human readable.
In HTTP/2,the protocols are binary based and not human readable,this means HTTP/2 will be more efficient to parse and compact on the wire.

HTTP

1.Is HTTP a stateless protocol ?
HTTP is stateless protocol,which means there is no memory of the past.Every transaction is performed as if it were being done for the very first time.
Statefull protocol means there is memory of the past and previous transactions are remembered and may affect the current transaction.
HTTP uses client-server model.An HTTP client opens a connection and sends a request message to an HTTP server,the server then returns a response message,usually containing the resource that was requested.After delivering the response,the server closes the connection making HTTP a stateless protocol,that is not maintaing any connection information between transactions.

2.How can you overcome the problem of stateless ?
The solution to overcome the problem of stateless is to use sessions,which means including a session identifier in each response or request.This can be done either by creating a session cookie or by including the session identifier in all the links.

3.What are HTTP request methods ?
HTTP request methods are:
(i)GET - retrieves data from a given server using a given URI -GET request has length restriction -GET requests can be cached,and bookmarked-Data is visible to everyone in URL
(ii)POST - submits data to be processed to the given server - POST request no length restriction-POST request cannot be cached,and bookmarked- Data is not displayed in the URL
(iii)PUT - replaces all current representations of the target resource with the uploaded content
(iv)DELETE - removes all current representations of the target resource given by a URI
(v)HEAD - same as GET,but transfers the status line and header section only
(vi)TRACE - performs a message loop-back test along the path to the target resource
(vii)OPTIONS - describes the communication options for the target resource

4.What are HTTP Status Codes ?

100 - continue - A part of the request has been received without any problems,and the rest of the request should now be sent.
101 - Switching protocols
200 -OK - the request was fulfilled.
201-Created -
202 -Accepted
203 - Partial Information
204 - No response - Server has received the request but there is no information to send back,and the client should stay in the same document view.
400- The request had bad syntax.
401 -Unauthorized
402 -Payment required
403 -Forbidden -The request is for something forbidden,authorization will not help.
404 - The server hasn’t found anything matching the URI given
500- Internal server error
501 -Server doesn’t support the facility required.
502 - Service temporarily overloaded
503 -Gateway timeout - 503 is equivalent to 500

RequireJS

1.What is the purpose of using RequireJS on server side ?
RequireJS when used on server side, we can compress Javascript and CSS files before sending them to the user,that is using NodeJS.

2.How to include RequireJS in index.html ?

```
<script    data-main=“config”   src=“require-2.1.22.js”></script>
<script>
require ( [‘config’], function( ) {
          // all of your code here.
}
</script>
```

For an instance :

```
<script    data-main=“config”   src=“require-2.1.22.js”></script>
<script>alert( );</script>
```

The above code allows the loading of “config” file only after clicking the “OK” button of “alert”popup.

```
<script    data-main=“config”   src=“require-2.1.22.js”></script>
<script>
require ( [‘config’], function( ) {
          alert( );
          } );
</script>
```

The above code allows the loading of “config” file before the “alert”popup is loaded.

How to configure the config.JS file ?

```
requires.config ( {
  baseUrl : ‘js’ ,                              // js is a folder which contains all the libraries installed
  paths: {
          angular: ‘angular.min.js’ ,    // this file is contained in js folder (the .js extension is optional)
          jQuery: ‘jQuery.min.js’                  // this file is contained in js folder (the .js extension is optional)
          customFile : ‘customFolder / customFile.js    //customFile.js is contained in customFolder
          }
} );
```

Note: angular,jQuery,customFile which are on the left side are called as Module IDs.
Note: If baseUrl: ‘js’ is not mentioned then the above code will look like this :

What happens if baseUrl isn’t used while configuring the config.JS file ?

```
requires.config ( {
          paths: {
                    angular: ‘js / angular.min.js’ ,         // this file is contained in js folder
                    jQuery: ‘js / jQuery.min.js’             // this file is contained in js folder
              customFile : ‘customFolder / customFile.js    //customFile.js is contained in customFolder

          }
} );
```

Here,we are mentioning ‘js /’ each time so to avoid this,we make use of baseUrl: ‘js’

Is it mandatory to mention two resources for each library while configuring the config.JS file ?
It is suggested to mention two resources for each library,that is,CDN and min.JS
The above code look like this :

```
requires.config ( {
  baseUrl : ‘js’ ,                              // js is a folder which contains all the libraries installed
  paths: {
          angular: [
          ‘https://ajax.googleapis.com/ajax/libs/angularjs/1.5.8/angular.min.js',     /* this is called as external URL
resource */

     ‘angular.min.js’ ,     /*This resource is available in your local machine*/    /* this is called as fallback path,which
will be loaded if the external URL resource is failed to load*/

        ],
          jQuery: ‘jQuery.min.js’
          customFile : ‘customFolder / customFile.js    //customFile.js is contained in customFolder


                    }
} );
```

How to declare define( ) ?

```
define ( [‘jQuery’, ‘angular’, ‘ return statement of customFile’], function($,x,y ) {

                              });
```

What are shims ?

```
require.config({
    paths: {
        'jquery': '../lib/jquery-2.1.1',
        'bootstrap': '../relib/bootstrap'
    }
});
```

The problem with the above code is that the above code doesn’t work always because RequireJS will request the jQuery and Bootstrap modules in parallel.In this case,if jQuery module is loaded first then everything is fine,but if

Bootstrap module is loaded first then it will throw an exception as Bootstrap is dependent on jQuery and which(jQuery) is not available.
To fix this problem,we make use of shims as follows :

```
require.config({
   paths: {
      'jquery': '../lib/jquery-2.1.1',
      'bootstrap': '../lib/bootstrap
   },

   shim: {
      'bootstrap': {
         deps: ['jquery']
      }
   }
});
```

In this case,if jQuery module is loaded first then everything is fine,and if Bootstrap is trying to load first then its (Bootstrap’s) dependency, that is, jQuery will be loaded first.

What is a module ?
A module is a self contained and reusable piece of code.

what is an r.JS ?
(i)create a “config-build.js” file and configure it as follows :

```
({
   name: 'config',              /* config is the normal version*/
   out: 'config-build.js’      /*config-build.js is the minified version*/
})
```

(ii)Run the command : node r.js -o   config-build.js
(iii)<script   data-main=“config-build.js”   src=“require-2.1.22.js”></script>

Note:Earlier,we mentioned data-main=“config”,which is a normal version, but
data-main=“config-build.js” is a minified version.

The problem with the above code is that ,we cannot “r.js” command on external URLs like jQuery and Bootstrap,so the problem can be resolved as follows :

```
({
   name: 'config',              /* config is the normal version*/
   out: 'config-build.js’      /*config-build.js is the minified version*/
   paths: {
                  ‘jquery’ : ‘empty: ’
                  ‘bootstrap: ‘empty: ’
                  }
})
```

What are source maps ?
source maps are used to translate a minified file to a non-minified file.

What is the difference between RequireJS module system and Angular module system ?
AngularJS module system deals with configuration of the injector which (configuration of the injector) defines how things are built at runtime and the building of the things is something that happen throughout the lifetime of the app.
RequireJS module system deals with how the script tags are loaded into the browser and this is something that happens only at the beginning and only once.

How to specify the order for loading the scripts ?

RequireJS is an implementation of AMD specification.

AMD should be named as Asynchronous Module Dependencies but it is named as Asynchronous Module definition

Lodash, Underscore

1.What is the difference between Lodash and Underscore ?
Lodash is a superset of Underscore,and is faster than Underscore.

2.Why should we use Lodash ?
Lodash makes JavaScript easier by taking the hassle out of working with arrays,numbers, objects, strings.
Lodash's modular methods are great for:
(i)Iterating arrays, objects, & strings
(ii)Manipulating & testing values
(iii)Creating composite functions

HAS.js

1.What is HAS.js ?
HAS.js is similar to Modernizer,but Modernizer tests only for HTML5 / CSS3 features,whereas HAS.js tests for :
(i)HTML5 features.

(ii)CSS features like rgba,border radius.
(iii)ES5 features like Object.freeze, Array.map
(iv)Script loading features like defer and async.
(v)Native JSON support,Audio support ,and Video support
(vi)XHR support

Azure Cloud Service

1.What is a cloud service ?
A cloud service is a container for one or more virtual machines you create.You can create a cloud service for a single virtual machine or you can load balance multiple virtual machines by placing them in the same cloud service.
Virtual machines are of two types Web Role Virtual Machine and Worker Role Virtual Machine.
Web Role runs windows server with IIS(Internet Information Service).
Worker Role runs windows server without IIS.

2.How to deploy your package to a cloud service ?
Deploying your package to a cloud service means uploading the two files :
(i)Package file which is .cspkg file
(ii)Cloud service configuration file which is .cscfg file

3.What does .cspkg file consists of ?
.cspkg file consists of application code and resources.
.cspkg file also consists of .csdef file (cloud service definition file).
.csdef consists of setting used by Azure to create a cloud service.
.cspkg file is created by the developers.
.cspkg file is in zip format

4.What does .cscfg file consists of ?
.cscfg file consists of configuration settings for the cloud service and individual roles,including the number of instances.

5.Where does the .cspkg file and .cscfg file reside ?
bin / Release / app.publish /   - has the .cspkg file and .cscfg file

6.What are the methods of deploying a package to a cloud service ?
The methods of deploying to a cloud service are :
(i)Through visual studio publish wizard
(ii)CSPack - command line tool
(iii)Management Portal
(iv)PowerShell

7.What are the different slots of deployment ?
When you are deploying your package to the cloud service,you have an option to deploy your package to either production slot or staging slot
The two slots of deployment are :
(i)staging slot : The package is first deployed to staging slot which allows the testers and pilot users to test the web application in order to check everything is working fine.
(ii)production slot: The package is deployed to production slot after ensuring that the web application is working

fine in the staging slot.

Purify-css

1.What is purify-css ?
purify-css is used to remove unused CSS.
purify-css reduces the page loading time by removing the unused CSS.
purify-css includes the following steps to reduce the page loading time :
(i)concat
(ii)purify
(iii)minify

Charting libraries

1.What are the different graphics or charting libraries available for Javascript and which one is preferred ?
Chart.JS
D3.JS (Data Driven Documents)
n3-charts - is built on top of D3.JS and AngularJS
HighchartsJS

Client-side database

1.What are the different methods to store data on the client side ?
(i)cookies
(ii)local storage
(iii)web sql
(iv)IndexedDB
local storage, session storage , Web Sequel, IndexDB are about storing user data.

2.Why do we want to store data on the client side ?
The main reason is practicality. JavaScript code running on the browser does not necessarily need to send all information to the server. There are several use cases:
You want to increase performance. You can cache data client-side so it can be retrieved without additional server requests.
You have a significant quantity of client-side-only data, e.g. HTML strings or widget configuration settings.
You want you make your application work off-line.

3.What is the difference between local storage and IndexDB ?
local storage is synchronous whereas IndexDB is asynchronous
local storage doesn’t offer any transactions support whereas IndexDB offers transaction support
local storage is simply a durable HashMap

4.What is IndexedDB ?
IndexedDB is a new HTML5 concept and is used to store data inside user’s browser.

IndexedDB is more powerful than local storage and is useful for applicants that requires to store large amount of data and these applications can run more efficiently and load faster.

Selenium WebDriver

1.What are the different types of locators available in selenium web driver ?
The eight different types of locators available in Selenium web driver are :
(i)By.id( )
(ii)By.name( )
(iii)By.tagName( )
(iv)By.className( )
(v)By.linkText( )
(vi)By.partialLinkText( )
(vii)By.xpath
(viii)By.cssSelector( )

2.What is XPath ?
XPath is used to identify the dynamic web elements in a web page.

3.What is the difference between Assert and Verify ?
Assert is used to verify the result.If the test case fails then it will stop the execution of the test case there itself and move the control to the other test case.
Verify is used to verify the result.If the test case fails then it will not stop the execution of that test case.

4.How to upload a file ?
sendKeys( ) method is used to upload a file.
driver.findElement (By.xpath (“input field”)).sendKeys(“path of the file which you want to upload”);

5.How to simulate browser back and forward ?
driver.navigate( ).back( );
driver.navigate( ).forward( );

6.How to get the current page URL ?
driver.getCurrentURL( ) method is used to get the current page URL.

7.What is the difference between ‘ / ’ and ‘ // ’ ?
‘ / ’ is used to find the immediate child.
‘ // ’ is used to search in the entire structure.

8.What is the difference between findElement and findElements ?
findElement and findElements are abstract methods of WebDriver interface and are used to find the web element in a web page.

9.What are the different types of exceptions you got while working with WebDriver ?
(i)ElementNotVisibleException
(ii)ElementNotSelectableException
(iii)NoAlertPresentException
(iv)NoSuchAttributeException
(v)NoSuchWindowException
(vi)TimeoutException
(vii)WebDriverException

10.How to clear the contents of a text box in selenium ?
clear( ) method is used to clear the contents of a text box.

11.How to invoke an application in WebDriver ?
driver.get(“url”) or driver.navigate( ).to(“url”);

12.Which package needs to be imported in order to work with WebDriver ?
org.openqa.selenium

13.How to check if an element is visible or not ?
isDisplayed( ) method is used to check if a element is visible or not.

14.How to check if a button is enabled or not ?
isEnabled( ) method is used to check if a button is enabled or not.

15.How to check if a checkbox or radio button is selected ?
isSelected( ) method is used to check if a checkbox or radio button is selected.

16.How to get the title of a web page ?
getTitle( ) method is used to get the title of a web page.

17.How to get the width and height of the text box ?
driver.findElement (By.xpath (“xpath of textbox”)).getSize( ).getWidth( );
driver.findElement (By.xpath (“xpath of textbox”)).getSize( ).getHeight( );

18.What is getOptions( ) ?
getOptions( ) is used to get the selected option from the dropdown list.

19.What is deSelectAll( ) ?
deSelectAll( ) is used to deselect all the options which have been selected from the dropdown list.

20.Is WebElement an interface or class ?
WebElement is an interface

21.Is FireFoxDriver an interface or class ?
FireFoxDriver is a class and it implements all the methods of a WebDriver interface.

22.Which is the super interface of WebDriver ?
SearchContext is the super interface of WebDriver.

23.What is the difference between close( ) and quit( ) ?
close( ) is used to close the browser where the control is.
quit( ) is used to close all the browsers opened by WebDriver.

24.How to hover the mouse of an element ?

```
Actions act = new Actions(driver);
act.moveToElement(webelement);    //webelement on which you want to move cursor
```

25.What is the command line we have to write inside a .bat file to execute a selenium project when we are using TestNg ?

```
java -cp bin;jars/* org.testng.TestNG testng.xml
```

26.What are the different ways to refresh a page ?
The three different ways to refresh a page are :

```
driver.get (driver.getCurrentUrl ( ) );
driver.navigate( ).to (driver.getCurrentUrl ( ) );
driver.findElement (By.xpath(“ ”)).sendKeys(Keys.F5 );
```

27.How to set the size of a window ?

```
driver.manage( ).window( ).setSize( new dimension (300,500) );
```

28.How to get the size of a window ?

```
driver.manage( ).window( ).getSize( ).getHeight( );
```

29.How to set the position of a window ?

```
driver.manage( ).window( ).setPosition( new Point (50, 200) );
```

30.How to get the position of a window ?
To get the position of a window from left side :

```
driver.manage( ).window( ).getPosition( ).getX( );
```

To get the position of a window from top side of a screen :

```
driver.manage( ).window( ).getPosition( ).getY( );
```

31.How many types of waits available in Selenium web driver ?
(i)Implicit waits
(ii)Explicit waits

Implicit wait :

```
WebDriver driver = new FirefoxDriver();
driver.manage().timeouts().implicitlyWait(10, TimeUnit.SECONDS);
driver.get("http://url_that_delays_loading");
WebElement myDynamicElement = driver.findElement(By.id("myDynamicElement"));
```

Explicit wait :

```
WebDriverWait wait = new WebDriverWait(driver, 10);
WebElement element = wait.until(ExpectedConditions.elementToBeClickable(By.id("someid")));
```

32.How to type the value in prompt dialog’s box input field using selenium web driver ?

```
driver.switchTo( ).alert( ).sendKeys(“John”);
```

33.What are the disadvantages of xpath locator as compared to cssSelector locator ?
(i)xpath locator is slower than cssSelector locator
(ii)xpath locator which works in one browser may not work in other browser.For an instance,Firefox browser reads the upper-cased Tag name and Attribute name but IE browser cannot read the upper-cased Tag name and Attribute name.IE browser reads only lower-cased Tag name and Attribute name.

34.Can we perform drag and drop operation in Selenium web driver and how to drag X element and drop it on Y element ?

```
new Actions(driver).dragAndDrop(X, Y).build().perform();
```

35.How to read Javascript alert message string,clicking on OK button and clicking on Cancel button ?
(i)To read alert message string :

```
String alrtmsg = driver.switchTo().alert().getText();
```

(ii)To click on OK button of alert :

```
driver.switchTo().alert().accept();
```

(iii)To click on Cancel button alert :

```
driver.switchTo().alert().dismiss();
```

36.What cannot we automate using selenium web driver ?
(i)Bitmap comparison
(ii)Automating Captcha
(iii)Reading bar code

37.How to automate HTML5 video ?

```
document.getElementById(“VideoId”).play( ); - To play the video
document.getElementById(“VideoId”).pause( ); - To pause the video
document.getElementById(“VideoId”).paused; - To check if video is paused
document.getElementById(“VideoId”).volume=0.5; - To increase volume
document.getElementById(“VideoId”).load( );  -  To reload video
document.getElementById(“VideoId”).muted;  - To check if video is muted
document.getElementById(“VideoId”).currentTime=0; - To replay the video from starting
```

Application cache is about storing web artifacts,which is what we use to store HTML,CSS,Javascript and Images

With Application Cache,our website still functions even when we are disconnected from the internet.

Application cache is a Map.

In case of Safari on desktop,the Application Cache size is unlimited.
In case of Safari on mobile, the Application Cache size is 10mb.

In case of Chrome on desktop,the Application Cache is 5mb.
In case of Chrome on Android mobile,the Application Cache is unlimited.

To use the files from application cache, we can use the application Cache API.
The "window.applicationCache" object provides events and properties that can be used to deal with data retrieval.
Current status of the application cache can be checked using window.applicationCache.status, which returns a numeric value mapped to the following states:
0 – uncached
If the page is not linked to the application cache; also, the very first time the application cache is being downloaded
1 – idle
When the browser has the latest version of the AppCache, and there are no updated versions to download, then the status is set to "Idle"
2 – checking
The duration of when the page is checking for an updated manifest file, then the status is set to "Checking"
3 – downloading
The duration of when the page is actually downloading a new cache (if an updated manifest file has been detected); the status is set to "Downloading"
4 – update ready
Once the browser finishes downloading that new cache and is ready to be used (but is not being used yet); during this time, the status is set as "Update ready"
5 – obsolete
In case the manifest file cannot be found, then the status is set to obsolete and the application cache gets deleted.

Events
Certain events also get fired, depending on what is going on with the AppCache at the moment.
Checking
This gets fired when browser is attempting to download the manifest for the first time, or is checking if there is an updated version of the manifest file.
Noupdate
If there is no updated version of the manifest file on the server, then "noupdate" is fired.
Downloading
If the browser is downloading the cache for the first time, or if it is downloading an updated cache, then this is fired.
Progress
This is fired for each and every file which is downloaded as part of the AppCache.
Cached
This is fired when all the resources have finished downloading, and application is cached.
Updateready
Once resources have finished re-downloading for an updated cached file, then updateready is called. Once this happens, then we can use swapCache() to make the browser to use this newly updated cache.
Obsolete
This is fired if the manifest file cannot be found (404 error or 410 error).

Error

This can be fired for a number of reasons.

If the manifest file cannot be found, then the application cache download process has to be aborted, in which case this event can be fired.

It can also be fired in case the manifest file is present, but any of the files mentioned in the manifest file cannot be downloaded properly.

It can even be fired in case the manifest file changes while the update is being run (in which case the browser will wait a while before trying again), or in any other case where there is a fatal error.

window.applicationCache.update(): This will trigger the application cache download process, which is nearly the same as reloading the page. It simply checks if the manifest has changed, and if so downloads a fresh version of all the content in the cache

window.applicationCache.swapCache(): This function tells the browser to start using the new cache data if it is available

Add the manifest file to the local server

Once the .manifest file is created we need to add that file to the root Web directory of our application (in case of Apache) which is done as follows:

```
AddType text/cache-manifest .manifest
```

The above code adds a directive in the “.htaccess” file.

This directive makes sure that every manifest file is served as “text/cache-manifest”. If the file is not served as required, the whole manifest will have no effect and the page will not be available offline.

Linking manifest file to html page

The manifest file once created can be added to the html pages with the help of “manifest” attribute of the <html> element.

```
<html manifest="offline.manifest">
```

Once we add the manifest file in our html page we can run it to check the functionality.

At this point of time we need to remember the fact that we need to allow the browser to enable the usage of AppCache, if asked.