Ext JS Interview Questions And Answers Guide.

Ext JS Job Interview Preparation Guide.

Question # 1

What do you know about EXTJS ver.2.2?

Answer:-

- It is a framework.
- 100% client side implementation.
- It is used as a standardization of Java side implementation on client side.
- JS DOM and XML DOM are incorporated in EXT JS
- It is a part of Yahoo toolkit.
- It is used by other technologies apart from JAVA.
- ITs a widget which has its own lifetime so we can call but we can't control over the lifetime.
- It's a hierarchical if you hold the root tag you can traverse all the elements within it using getByElementId
- EXT.get('myElementId')
- 2 methods
- class method prefix the calling object with '.'
- Object method prefix the calling object with '#'
- Flyweight Design Pattern is a pattern which traverse the root object and collect all the wastes.
- SPECIFIC FLYWEIGHT when it is used for a single object.
- GLOBAL FLYWEIGHT when it is used for a multiple object.
- It reconstructs old object when it is required.
- 58 widgets are builtin out of which 36 are UI widgets.
- EXT.onReady is an eventhandler which is used to start the hierarchy of DOM structure.
- Node is a super object.

Read More Answers.

Question # 2

What is JSF (Java Server Faces)?

Answer:-

- It is used for server side implementation
- It is used as a standardization of Java side implementation on server side.

Read More Answers.

Question # 3

Explain Ext JS development tools?

Answer:-

You can integrate Ext JS framework development into several popular integrated development environments (IDEs), including Eclipse, Aptana, and Komodo. For information about including Ext JS development support in your IDE of choice.

Read More Answers.

Question # 4

Tell me why did you choose Ext JS?

Answer:-

The overall design of extjs is exemplary.One can learn a lot from it's unified architecture â€" no matter which language one is programming in.
Extjs requires you to start with one of their base classes â€" ensuring a consitent model. Consistency is extremely important for the library to be reusable.
a.Extjs documentation seems to be very comprehensive and well maintained.
b.key aspect of the EXTJS Library is the cross-browser support.
c.Build rich Internet applications with Ext JS
d. Ext JS framework is the multitude of rich UI elements provided. These elements include forms, dialog boxes, tabs, trees, and grids.
e.The Ext JS framework includes support for Ajax implementations.
f.Ext JS integration with other Web server frameworks.
g.Ext JS framework development into several popular integrated development environments (IDEs), including Eclipse, Aptana, and Komodo.
h.Ext JS provides excellent performance.The framework is fully object oriented and extensible. Because it's written in the JavaScript language

Question # 5

Suppose to show our message when any paragraph in our test page is clicked, what is the extjs code on paragraph click?

Answer:-

Ext.onReady(function() {
Ext.select('p').on('click', function() {
alert("You clicked a paragraph");
});
});
or
Ext.onReady(function() {
var paragraphClicked = function() {
alert("You clicked a paragraph");
}
Ext.select('p').on('click', paragraphClicked);
});

Question # 6

Explain the extjs library files to include in JSP page?

Answer:-

ext-base.js
ext-all-debug.js or ext-all.js
ext-all.css
base.css or examples.css

Question # 7

Tell me what is the purpose of each() in store?

Answer:-

Calls the specified function for each of the Records in the cache
each(Function fn, [records Object])

Question # 8

What is vtype?

Answer:-

The validations provided are basic and intended to be easily customizable and extended.
Few vtypes provided by extjs are as below:
emailText : String, The error text to display when the email validation function returns false
alphanumText : String, The error text to display when the alphanumeric validation function returns false
urlText : String, The error text to display when the url validation function returns false

Question # 9

Explain purpose of Load mask?

Answer:-

To apply mask to page level / component level.
restrict user not to access any components in page
var pageProcessBox = new Ext.LoadMask(Ext.getBody(), { msg: 'Loading Employee details.' });
pageProcessBox.show();

Question # 10

Tell me how to commit a record modification?

Answer:-

newRecord.commit();

Question # 11

Explain how to get a value of textfield or combo box?

Answer:-

using getvalue();
var selectedValue = mytextfield.getValue();

Question # 12

Explain how to hide the column in grid panel?

Answer:-

using hidden : true

Question # 13

Tell me what is maximum size of http post request?

Answer:-

Microsoft Internet Explorer has a maximum uniform resource locator (URL) length of 2,083 characters. Internet Explorer also has a maximum path length of 2,048 characters. This limit applies to both POST request and GET request URLs.

Question # 14

Explain why did you choose Ext JS?

Answer:-

Given the wide range of JavaScript libraries available it was important to choose the right one. We needed to choose a library that was consistent in the way that it presents information to the user, but also consistent in the way that you code using the library.
With all Ext components extending the â€˜Observable' class we had the ability to write consistent code in an event-driven manner, much like writing a desktop application, not easily achieved with other libraries. What's more is that we knew it would work cross-browser, again something not easily achieved that saves countless hours on large projects.

Question # 15

Tell me what features could we add to Ext to make building a rich application like PLANet easier in the future?

Answer:-

Once an application gets over a certain size, and customer releases become more frequent, the burden of testing the application starts to take its toll. An Ext supported test suite would save huge amounts of time. The current problem of test suites with Ext is being able to reliably predict the automatic ids that Ext generates for page elements. Test tools are beginning to implement support for CSS selectors to overcome this problem, but few currently exist. Ext could provide better documentation on how to build better test cases with Ext so that more time can be spent on development rather than testing.

Question # 16

Explain EXT-JS Life cycle?

Answer:-

Component Life Cycle
In general, the Component architecture in 2.0 will "just work." It's been designed to handle most of the management of components transparently to the end developer. However, there will come a time when something needs to be customized, or a Component needs to be extended. That's when a thorough understanding of the Component life cycle will become quite helpful. Following are the most important stages in the life cycle of every class based on Component:
1. Initialization:
The config object is applied
Classes that extend Component do not need to (and usually should not) provide a separate constructor. Component's constructor will not only apply any config passed into its subclasses, it also provides all of the following steps.
The base Component events are created
These are events that can be fired by any Component, and they are enable, disable, beforeshow, show, beforehide, hide, beforerender, render, beforedestroy, destroy (see the Component API docs for complete details).
The component is registered in ComponentMgr
As such it will always be available via Ext.getCmp.
The initComponent method is called
This is the most important initialization step for subclasses, as this is a template method intended to be implemented by each subclass to provide any needed constructor logic. The class being created is called first, and each class in the hierarchy back up to Component is expected to call superclass.initComponent. This method makes it easy to implement and, if needed, override the constructor logic at any step in the hierarchy.
Plugins are loaded (if applicable)
If this Component has any plugins specified in the config, they will be initialized at this time.
State is initialized (if applicable)
If the Component is state-aware, its state will be reloaded if available.
The component is rendered (if applicable)
The component is rendered immediately if either renderTo or applyTo is provided in the config, otherwise rendering is deferred until the Component is explicitly displayed in code or is told to render by its container.
2. Rendering:
The beforerender event is fired
This is a cancelable event, giving any handler the ability to prevent the Component from rendering if needed.
The container is set
If no container is specified, the parent node of the Component's underlying DOM element is set as the container.
The onRender method is called
This is the most important rendering step for subclasses, as this is a template method intended to be implemented by each subclass to provide the needed rendering logic. The class being created is called first, and each class in the hierarchy back up to Component is expected to call superclass.onRender. This method makes it easy to implement and, if needed, override the rendering logic at any step in the hierarchy.
The Component is "unhidden"
By default, many components are hidden using special CSS classes like "x-hidden". If the autoShow config value is true, any "hide" classes are removed from the component at this time.
Custom class and/or style applied

All Component subclasses support the special config properties of cls and style which are a custom, user-defined CSS class and rule respectively that will be applied to the DOM element underlying this Component. Specifying the cls value is the preferred method for visually customizing a Component and its constituent parts. Since the class will get applied to the topmost wrapper element of the Component's markup, any sub-elements of the Component can be adjusted using standard CSS inheritance rules.

The render event is fired

This is a notification that the Component has been successfully rendered at this point. You can safely assume that its DOM elements are now available to your code if needed. If you attempt to access a Component prior to rendering, it won't be available and you'll get an error.

The afterRender method is called

This is another template method for subclasses that can be implemented or overridden to provide any special post-rendering logic that may be needed. Each subclass is expected to call superclass.afterRender.

The Component is hidden and/or disabled (if applicable)

The hidden and disabled config values are applied at this point.

Any state-specific events are initialized (if applicable)

State-aware Components can declare special events that are specific to loading and saving state. If supplied, any such events will be added.

3. Destruction :

The beforedestroy event is fired

This is a cancelable event, giving any handler the ability to prevent the Component from being destroyed if needed.

The beforeDestroy method is called

This is another template method that can be implemented or overridden to provide any special pre-destruction logic that may be needed. Each subclass is expected to call superclass.beforeDestroy.

Element and its listeners are removed

If the Component has been rendered, its underlying Element's event listeners are removed and the Element itself is then removed from the DOM.

The onDestroy method is called

This is another template method that can be implemented or overridden to provide any special post-destruction logic that may be needed. Each subclass is expected to call superclass.onDestroy. Note that the Container class (and any Container subclasses) provides a default implementation of onDestroy that automatically loops through its items collection and calls destroy on each child Component recursively.

Component is unregistered from ComponentMgr

It will no longer be available via Ext.getCmp.

The destroy event is fired

This is simply a notification that the Component has been successfully destroyed at this point and is no longer available in the DOM.

Event listeners on the Component are removed

The Component itself can have event listeners separately from its underlying Element. If any exist, they are removed.

EXTJS:
Elements
Event Handling
Widget Examples
Application Layout/scope
Ext.extend
Element:
var div = Ext.get('myDiv');
div.setWidth(100);

Question # 17

Tell me what is method to Update the message box body text for MessageBox?

Answer:-

updateText([String text]) : Ext.MessageBox

Question # 18

Tell me how to apply css on select of combo box?

Answer:-

using config option as
emptyClass : 'emptycss', where emptycss is a css classname

Question # 19

Explain how to register callbacks to the load and exception events of the JsonStore?

Answer:-

```
var grid = new Ext.grid.GridPanel({
store: new Ext.data.JsonStore({
[...]
listeners: {
load: this.onLoadSuccess.crateDelegate(this),
exception: this.onLoadException.createDelegate(this)
}
}),
onLoadSuccess: function () {
// success
},
onLoadException: function () {
// error
},
[...]
}
```

Question # 20

Explain extjs decode()?

Answer:-

var json = Ext.decode(response.responseText);
Ext.Msg.alert('Error', json.error);

Question # 21

What is Ext JS?

Answer:-

Ext JS is a pure JavaScript application framework for building interactive web applications using techniques such as Ajax, DHTML and DOM scripting.

Question # 22

Explain Namespace importance?

Answer:-

An example which utilizes 3 distinctly different sets of scripts from different organizations. We use Ext JS for enhancements, Google Analytics for tracking site usage and the native vBulletin scripts. You can see how all of this code from different sources has been included in the same page.
Namespacing is important for developers in order to organize their code and ensure that their code is not overwritten when loaded in the JavaScript interpreter. If another developer defines a variable with the same name your existing definition will be overwritten.
Because JavaScript is a functionally scoped language creating a function and/or variable which is not wrapped in another function will result in that variable being created in the global scope (window). To combat this, developers place their classes in Objects.
As the client-side JavaScript included in web applications gets larger and more advanced, organization of 3rd party code and your own code becomes increasingly important. Using namespaces will ensure your JavaScript code is safe from other code overwriting it in the global namespace.
Example grid pre-configured class,apply config, Register Grid , used as xtype
http://examples.extjs.eu/ (grid in border layout)

Question # 23

Explain requirements for a simple EXT-JS?

Answer:-

1.Call Ext-Base.js in a html file
2.This file should exist under webcontent/adapter/ext - Library is added.
3.Default stylesheet ext-all.css is avb under resources/css
4.Have your own js file link it under HTML file if required.
5.Helping js can also be called inside your HTML file.
6.Have layer to print your object. The layer is called Layer Dialog Area.
7.Use onReady() inside which object can be created and declared
8.Call the object with the EXT.get()
9.To open a window EXT.window() to set the properties and use show() to show the result in the monitor.

Question # 24

Explain Ext JS UI elements?

Answer:-

The heart of the Ext JS framework is the multitude of rich UI elements provided. These elements include forms, dialog boxes, tabs, trees, and grids.

Question # 25

Tell me what is the purpose of Element Object in Extjs?

Answer:-

->Element wraps most of the DOM methods and properties that you'll need, providing a convenient, unified, cross-browser DOM interface (and you can still get direct access to the underlying DOM node when you need it via Element.dom)
->The Element.get() method provides internal caching, so multiple calls to retrieve the same object are incredibly fast
->The most common actions performed on DOM nodes are built into direct, cross-browser Element methods (add/remove CSS classes, add/remove event handlers, positioning, sizing, animation, drag/drop, etc.)

Question # 26

Tell me how to handle event for a extjs component?

Answer:-

a. using listeners config object.
For ex for grid events : listeners: {rowclick: gridRowClickHandler,rowdblclick: gridRowDoubleClickHandler}
b. using addListener(String eventName, Function handler, [Object scope], [Object options]) : void

Appends an event handler to this component
c. using on(String eventName, Function handler, [Object scope], [Object options]) : void
Appends an event handler to this element (shorthand for addListener)
For ex: store.on("datachanged", function(store){ ..... });

Read More Answers.

Question # 27

Explain how to handle exception while loading datastore?

Answer:-

using loadexception event.
syntax: store.loadexception() : Fires if an exception occurs in the Proxy during loading.
use beforeload : (Store this, Object options) : Fires before a request is made for a new data object. If the beforeload handler returns false the load action will be canceled.
syntax:
store.on('loadexception', function(event, options, response, error) {
alert("Handling the error");
event.stopEvent();
});

Read More Answers.

Question # 28

Explain how to handle updates for store changes?

Answer:-

use store.commitChanges()

Read More Answers.

Question # 29

Explain how to get modified records using store object?

Answer:-

store.getModifiedRecords() : Gets all records modified since the last commit.

Read More Answers.

Question # 30

Explain what is the purpose of load() in store?

Answer:-

store.load() : returns boolean
Loads the Record cache from the configured Proxy using the configured Reader.
For remote data sources, loading is asynchronous, and this call will return before the new data has been loaded.
store.load({callback: fnCheckData, scope: this});

Read More Answers.

Question # 31

Explain how to stop editing a record?

Answer:-

newRecord.endEdit();

Read More Answers.

Question # 32

Tell me how to start editing a record?

Answer:-

newRecord.beginEdit();

Read More Answers.

Question # 33

Explain Extjs Vs jQuery?

Answer:-

ExtJs and JQuery are kind of apples and oranges. You can compare Ext Core to JQuery, and ExtJs to JQuery UI.
Ext JS is a full-fledged widget library while jQuery (not jQuery UI) and Mootools are JavaScript frameworks that help with DOM manipulation etc.
Whilst jQuery and Mootools help with the general workings of a site.
jQuery UI is a much less rich set of components.
Ext JS seems to be focussed on tables and storing data, plus manipulating it.

Read More Answers.

Question # 34

Explain EXTJS components?

Answer:-

Form panel
Border layout
Form panel components:
combo box,textfield,html editor,numberfield
combo box:
select,change events
mode,lazyinit
grid panel:
colmodel,json reader,json store,simple store methods
cellclick,rowclick,click
grid selection models
tree panel:
node,event handlers

Read More Answers.

Question # 35

Explain Integration of Ext JS?

Answer:-

You can use Ext JS with other common Web development server-side frameworks, including PHP, the Javaâ„¢ language, MicrosoftÂ® .NET, Ruby on Rails, and ColdFusion.

Read More Answers.

Question # 36

Tell me why we need JavaScript Library?

Answer:-

Javascript is an awesome language. It's super flexible.Browsers are the modern UI paradigm.
The javascript Libraries now must provide a rich set of UI Widgets.
javascript libraries:
* JQuery
* Qooxdoo
* Dojo
* Prototype.js
* mootools
* extjs

Read More Answers.

Question # 37

Explain integration of Web development server-side frameworks with Ext JS?

Answer:-

You can use Ext JS with other common Web development server-side frameworks, including PHP, the Javaâ„¢ language, MicrosoftÂ® .NET, Ruby on Rails, and ColdFusion.

Read More Answers.

Question # 38

Explain how to find no of records in a store?

Answer:-

using store.getCount() : Gets the number of cached records.
store.getTotalCount() : Gets the total number of records in the dataset as returned by the server.

Read More Answers.

Question # 39

Explain purpose of renderer in grid panel?

Answer:-

using config option,
renderer: fnCellColor where fnCellColor is method to apply color to a cell.

Read More Answers.

Question # 40

Tell me what is use of combo select event function?

Answer:-

To get the selected value from a combo.using getvalue();
var selectedComboValue = mycombo1.getValue();

Read More Answers.

Question # 41

Explain what are components required for grid panel?

Answer:-

store, columnmodel, id, width,height
46. how to disable menu option for header in columnModel?
using menuDisabled: true

Read More Answers.

Question # 42

Please tell us do you have any advice for developers using Ext for the first time?

Answer:-

Ext can be used by Web Application developers who are familiar with HTML but may have little or no experience with JavaScript application development. If you are starting to build a new web application, or you are revamping an existing application, then take your time to understand the basics of the library including:

Read More Answers.

Question # 43

Ext-JS Interview Questions:

Answer:-

Understanding the major UI components (grid, form, tree etc)
How to integrate your existing data with Ext?
How to use Ext Layouts?
The Ext Component Model?

Read More Answers.

Question # 44

Explain Ext JS integration with other Web server frameworks?

Answer:-

You can use Ext JS with other common Web development server-side frameworks, including PHP, the Java language, Microsoft .NET, Ruby on Rails, and ColdFusion. For integration specifics for each of these frameworks.

Read More Answers.

Question # 45

Tell me What are major Web browsers supported by Ext JS framework?

Answer:-

* Windows® Internet Explorer® version 6 and later.
* Mozilla Firefox version 1.5 and later (PC and Macintosh).
* Apple Safari version 2 and later.
* Opera version 9 and later (PC and Mac).

Read More Answers.

Question # 46

Explain where Extjs extended from?

Answer:-

Ext JS as a project to extend the functionality that the YUI Library.A key aspect of the YUI Library is the cross-browser support.The Extjs framework is fully object oriented and extensible. Because it's written in the JavaScript language.

Read More Answers.

Question # 47

Tell me do you have any advice for developers using Ext for the first time?

Answer:-

Ext can be used by Web Application developers who are familiar with HTML but may have little or no experience with JavaScript application development. If you are starting to build a new web application, or you are revamping an existing application, then take your time to understand the basics of the library including.

Read More Answers.

Question # 48

Explain how to access Dom element using EXTJS?

Answer:-

The Element API is fundamental to the entire Ext library.
Using traditional Javascript, selecting a DOM node by ID is done like this:
var myDiv = document.getElementById('myDiv');
Using Extjs:
Ext.onReady(function() {
var myDiv = Ext.get('myDiv');
});

Read More Answers.

Question # 49

Please write syntax for MessageBox show() method?

Answer:-

Ext.MessageBox.show({
title: 'Paragraph Clicked',
msg: 'User clicked on Paragraph',
width:400,
buttons: Ext.MessageBox.OK,
animEl: paragraph
});

Read More Answers.

Question # 50

Tell me what is parent class for all stores in extjs? how many stores exists?

Answer:-

Ext.data.Store is parent class for all stores.
A Store object uses its configured implementation of DataProxy to access a data object unless you call loadData directly and pass in your data.
subclasses for Store:
GroupingStore, JsonStore, SimpleStore

Read More Answers.

Question # 51

Please explain how to get record using index?

Answer:-

store.getAt(Number index) : Get the Record at the specified index.

Read More Answers.

Question # 52

Explain what is purpose of loadData() in store?

Answer:-

store.loadData(Object data, [Boolean append]) : void
Loads data from a passed data block and fires the load event.
loadData(storeData,false); False to replace the existing records cache.
loadData(storeData,true) : True to append the new Records rather than replace the existing cache.

Read More Answers.

Question # 53

What is xtype?

Answer:-

The xtype will be looked up at render time up to determine what type of child Component like TextField, NumberField etc to create. i,e
xtype = Class
----
button = Ext.Button
textfield = Ext.form.TextField
radio - Ext.form.Radio
grid = Ext.grid.GridPanel
combo = Ext.form.Combobox
toolbar = Ext.Toolbar

Read More Answers.

Question # 54

Do you know how to get selection model used in a grid panel?

Answer:-

using grid.getSelectionModel(); method

Read More Answers.

Question # 55

Explain Ext JS and Ajax?

Answer:-

The Ext JS framework includes support for Ajax implementations. Typically, a common feature of Ajax applications is for an application to asynchronously respond to user input by updating the UI without redisplaying the entire Web page. A typical Ext JS Ajax implementation: an HTML text field and button element that posts data in the text field to a Web server when the button is clicked.

Read More Answers.

Question # 56

Explain Extjs Ajax implementation?

Answer:-

A typical Ext JS Ajax implementation: an HTML text field and button element that posts data in the text field to a Web server when the button is clicked.

Question # 57

Explain what is syntax for Extjs Button click event?

Answer:-

Ext.onReady(function() {
Ext.get('myButton').on('click', function(){
alert("You clicked the button");
});
});

Question # 58

Explain what is use of Ext.onReady() function?

Answer:-

Ext.onReady is probably the first method that you'll use on every page. This method is automatically called once the DOM is fully loaded, guaranteeing that any page elements that you may want to reference will be available when the script runs
syntax:
Ext.onReady(function() {
alert(â€œCongratulations! You have Ext configured correctly!â€•);
});

Question # 59

Explain the css file required to apply Extjs Theme property?

Answer:-

xtheme-gray.css
ext-all.css

Question # 60

Do you know what is purpose of MessageBox?

Answer:-

MessageBox is asynchronous.
MessageBox call, which demonstrates the readable message to user.
MessageBox used for multiple purpose like
Ext.Msg.alert()
Ext.Msg.prompt()
Ext.Msg.show({});
Ext.Msg.wait();

Question # 61

What is a widget?

Answer:-

A widget is a tiny piece or component of functionality.

Question # 62

Tell me how to get record using id?

Answer:-

store.getById(String id) : Get the Record with the specified id.

Question # 63

How to how many types of layout managers exist in extjs?what are they?

Answer:-

Layouts fall under this package Ext.layout.*
Types of layouts:
Absolute Layout:
This is a simple layout style that allows you to position items within a container using CSS-style absolute positioning via XY coordinates.
Accordion Layout:
Displays one panel at a time in a stacked layout. No special config properties are required other than the layout.
All panels added to the container will be converted to accordion panels.
AnchorLayout:

This type of layout is most commonly seen within FormPanels (or any container with a FormLayout) where fields are sized relative to the container without hard-coding their dimensions.

BorderLayout:

Border layouts can be nested with just about any level of complexity that you might need.

Every border layout must at least have a center region. All other regions are optional.

CardLayout (TabPanel):

The TabPanel component is an excellent example of a sophisticated card layout. Each tab is just a panel managed by the card layout such that only one is visible at a time

CardLayout (Wizard):

You can use a CardLayout to create your own custom wizard-style screen.

FitLayout:

A very simple layout that simply fills the container with a single panel.

FormLayout:

FormLayout has specific logic to deal with form fields, labels, etc.FormLayout in a standard panel,

ColumnLayout:

This is a useful layout style when you need multiple columns that can have varying content height.Any fixed-width column widths are calculated first, then any percentage-width columns specified using the columnWidth config

TableLayout:

Outputs a standard HTML table as the layout container.you want to allow the contents to flow naturally based on standard browser table layout rules.

Read More Answers.

Question # 64

Explain how we can apply pagination in grid panel?

Answer:-

using Ext.PagingToolbar plugin, we can implement pagination to a grid panel
syntax:
new Ext.PagingToolbar({
pageSize: 25,
store: store,
displayInfo: true,
displayMsg: 'Displaying topics {0} - {1} of {2}',
emptyMsg: "No topics to display",
})
// trigger the data store load
store.load({params:{start:0, limit:25}});

Read More Answers.

Question # 65

Tell me how to get record object from store?

Answer:-

var record = grid.getStore().getAt(rowIndex);

Read More Answers.

Client Side Scripting Most Popular Interview Topics.

About Global Guideline.

Global Guideline is a platform to develop your own skills with thousands of job interview questions and web tutorials for fresher's and experienced candidates. These interview questions and web tutorials will help you strengthen your technical skills, prepare for the interviews and quickly revise the concepts. Global Guideline invite you to unlock your potentials with thousands of **Interview Questions with Answers** and much more. Learn the most common technologies at Global Guideline. We will help you to explore the resources of the World Wide Web and develop your own skills from the basics to the advanced. Here you will learn anything quite easily and you will really enjoy while learning. Global Guideline will help you to become a professional and Expert, well prepared for the future.

* This PDF was generated from https://GlobalGuideline.com at **November 29th, 2023**

* If any answer or question is incorrect or inappropriate or you have correct answer or you found any problem in this document then don't hesitate feel free and e-mail us we will fix it.

You can follow us on FaceBook for latest Jobs, Updates and other interviews material.
 www.facebook.com/InterviewQuestionsAnswers

Follow us on Twitter for latest Jobs and interview preparation guides
https://twitter.com/InterviewGuide

Best Of Luck.

Global Guideline Team
https://GlobalGuideline.com
Info@globalguideline.com