[EXTJS4 Interview Questions & Answers](http://extjsexamples.blogspot.com/2013/05/extjs4-interview-questions-answers.html)

**1. Requirements for a simple EXTJS ?**

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.

**2.**       **What modern browsers does EXTJS work on?**

There are 5, name them:  
EXTJS4 support cross platform browser compatibility.

IE 6+

FF 3+ (PC, Mac)

Safari 4+

Chrome 10+

Opera 11+ (PC, Mac)

**3.**       **What are the prerequisites for learning ExtJS?**

-First and most interest of self learning & work on open source libraries.

- Good knowledge on HTML , CSS, Javascript

-Drive through EXTJS API docs & practice samples

**4.**       **What is new in ExtJs 4 compared with its older versions?**

-          Extremely customizable Charts, free graphs and visualization using HTML5, Charting engine

-          Powerful  Themes introduced using SASS, makes extremely easy to customize your application theme.

-          ARIA,RTL (right to left language support) such as Arabic etc.  Fast interaction with mouse & keyboard (section 508A); EXTJS gives world class accessibility support.

-          End of Form Layout. With EXTJS4, forms  can use any layout, making it easy  to achieve any look & feel. (Forms not tied to Form layout)

-          Upgraded Components: component wise fixing bugs, improved UI, given new level of visual  polish.

-          Upgrade in GRID extension components - Row Editor, Tree Grid are rewritten and are in build into the framework. Locking, editing and infinite scrolling that can scroll forever.

-          Improved Data package – This is strongest part of EXTJS, package uses stores,readers,proxies and rest to seamlessly load data from any source. Using this data package build modern, scalable enterprise grade web apps using MVC. New Proxies like WebStorage,WebSQL, Session storage proxies used HTML5 feature to save data in client side, and achieved by single line of configuration.

-          Double the documentation including a new API browser.

-          Application Architecture – It enables us to create some incredible tools to help automate much of the design and maintenance into creating applications. Model –View-Controller is introduced in Architecture, defined with common file structure, best practices, 4000+ unit testcases for EXTJS components.

-          Faster,Easier,more stable, learn easy using 200+ examples with EXTJS4s

**5.**       **Explain a little about Ext.Element class?**

- Element wraps most of the DOM methods and properties that you'll need, providing a convenient, unified, cross-browser DOM interface.

Ext.dom.Element - Encapsulates a DOM element, adding simple DOM manipulation facilities, normalizing for browser differences.

// by id

var el = Ext.get("my-div");

// by DOM element reference

var el = Ext.get(myDivElement);

**6.**       **What do you understand by 'xtype' in ExtJS ?**

      The xtype will be looked up at render time up to determine what type of child Component like TextField, NumberField etc to create in a container. Some xtypes as below,

xtype     =   Class

button = Ext.Button

textfield = Ext.form.TextField

radio - Ext.form.Radio

grid   =  Ext.grid.GridPanel

combo =  Ext.form.Combobox

toolbar = Ext.Toolbar

eg:

var text1 = Ext.widget({

     xtype: 'textfield',

     fieldLabel: 'Foo'

 });

**7.**       **Explain the difference between 'ext.js' and 'ext-all.js' file?**

ext-all.js: This file contains the entire Ext JS framework (used for Development & testing)

ext.js: This file contains the minimum Ext JS code (Ext JS base library)- used in Production

**8.**       **What is the use of bootstrap.js file?**

If developer is not sure which file to include between ext-all.js or ext-all-debug.js then you can make right use of bootstrap.js file.

The only thing that this file does is import ext-all.js or ext-all-debug.js, depending on the environment we are using.

**9.**       **What is Statics in ExtJS 4?**

Any class can define static methods, which means we do not need to instantiate the class to call the method;

eg: ClassName.methodName().

To declare a static method or property, simply define it as statics in its class property.

List of static methods for this class. For example:

Ext.define('Computer', {

     statics: {

         factory: function(brand) {

             // 'this' in static methods refer to the class itself

             return new this(brand);

         }

     },

     constructor: function() { ... }

});

**10.**   **What do you mean by Sandbox mode in ExtJS 4?**

-To run multiple versions of EXTJS side by side. Leverage your investments and migrate at your own place.

- To use Ext JS 4 in the sandbox mode, you need to import the ext-all sandbox files and the ext-sandbox.css file as well.

**11.**   **How does Ext JS is useful in detecting the DOM is ready?**

Ext JS achieves cross browser compatibility by detecting which browser the code is executing on and manages the detection of the DOM ready state.

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(“Congratulations! You have Ext configured correctly!”);

});

**12.**   **Explain about utility class 'Ext.util.TaskRunner'?**

Provides the ability to execute one or more arbitrary tasks in a asynchronous manner. Generally, you can use the singleton Ext.TaskManager instead, but if needed, you can create separate instances of TaskRunner. Any number of separate tasks can be started at any time and will run independently of each other.

Example usage:

 // Start a simple clock task that updates a div once per second

 var updateClock = function () {

     Ext.fly('clock').update(new Date().format('g:i:s A'));

 }

 var runner = new Ext.util.TaskRunner();

 var task = runner.start({

     run: updateClock,

     interval: 1000

 }

The equivalent using TaskManager:

 var task = Ext.TaskManager.start({

     run: updateClock,

     interval: 1000

 });

**13.**   **Explain the life-cycle of a component in ExtJS?**

The most important stages in the life cycle of every class based on Component:

**1. Initialization:**

-The config object is applied

-The base Component events are created

-The component is registered in ComponentMgr

-The initComponent method is called

-Plugins are loaded (if applicable)

-State is initialized (if applicable)

-The component is rendered (if applicable)

**2.  Rendering:**

-The beforerender event is fired

-The container is set

-The onRender method is called

-The Component is "unhidden"

-Custom class and/or style applied

-The render event is fired

-The afterRender method is called

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

-Any state-specific events are initialized (if applicable)

**3. Destruction :**

-The beforedestroy event is fired

-The beforeDestroy method is called

-Element and its listeners are removed

-The onDestroy method is called

-Component is unregistered from ComponentMgr

-The destroy event is fired

-Event listeners on the Component are removed

**14.**   **Explain a little about component query class in ExtJS ?**

Ext.ComponentQuery is a class which is used for searching for components.

Provides searching of Components within Ext.ComponentManager (globally) or a specific Ext.container.Container on the document with a similar syntax to a CSS selector.

Components can be retrieved by using their xtype.

-component

-gridpanel

**15.**   **How does ExtJS is useful in detecting the DOM is ready ?**

Ext JS achieves cross browser compatibility by detecting which browser the code is executing on and manages the detection of the DOM ready state. Some browser fires some event on DOM ready or some browser sets some flag when DOM is ready.

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(“Congratulations! You have Ext configured correctly!”);

});

**16.**   **How can we define a constructor in a class in ExtJS 4?**

The constructors are special method that are executed when a class is instantiated. You can use constructor to prepare the object in any way required. For example, to set up the default property values.

eg:

Ext.define('Test.Emp', {

   config: {

        name: 'Raja',

        gender: 'Male'

   },

   constructor: function(config){

       // initialize our config object

       this.initConfig(config);

    },

    getDetails: function(){

      alert('My name is ' + this.name);

    }

});

var honey = Ext.create("Test.Emp",{

name: 'Mary',

       gender: 'Female'

});

**17.**   **How can you create a singleton class in ExtJS?**

To create a singleton class,just set to true, the class will be instantiated as singleton.

For example:

Ext.define('Logger', {

    singleton: true,

    log: function(msg) {

        console.log(msg);

    }

});

Logger.log('Hello');

**18.**   **What is the purpose of Layout Managers in ExtJS?**

Layouts fall under this package Ext.layout.\*

Layout managers are basically responsible for the visual organization of widgets onscreen. This involves keeping track of how the individual child items are placed to each other.

**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.

**19.**   **How can you create HashMap in ExtJS?**

It represents a collection of a set of key and value pairs. Each key in the HashMap must be unique, the same key cannot exist twice. Access to items is provided via the key only. Sample usage:

var map = new Ext.util.HashMap();

map.add('key1', 1);

map.add('key2', 2);

map.add('key3', 3);

map.each(function(key, value, length){

    console.log(key, value, length);

});

The HashMap is an unordered class, there is no guarantee when iterating over the items that they will be in any particular order. If this is required, then use a Ext.util.MixedCollection.

**Refer links for EXTJS4 Features:**

http://www.sencha.com/blog/introducing-ext-js-4-2/  
  
http://www.sencha.com/blog/ext-js-4-preview-faster-easier-more-stable/  
  
https://sites.google.com/site/mynotepad2/developer-notes/javascript/extjs/extjs-component-life-cycle

1.Why we need javascript Library?

Javascript is an awesome language. It’s super flexible.Browsers are the modern [UI](http://madesu.eu/get-to-know-ext-js/) paradigm.   
The javascript Libraries now must provide a rich set of [UI](http://madesu.eu/vaadin/) Widgets.  
javascript libraries:  
\* JQuery  
\* Qooxdoo  
\* Dojo  
\* Prototype.[js](http://madesu.eu/why-extjs/)  
\* mootools  
\* [extjs](http://madesu.eu/extjs4-interview-questions-answers/" \o "See also EXTJS4 Interview Questions & Answers)

2. Why did you choose Ext [JS](http://madesu.eu/extjs-overriding-the-inbuilt-methods-for-higher-customization/)?

The overall design of extjs is exemplary.One can learn a lot from it’s unified architecture – no matter which language one is programming in.  
Extjs requires you to start with one of their base classes – 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](http://madesu.eu/scala-exercise-3-decorator-and-composite-design-patterns/) elements provided. These elements include forms, dialog boxes, tabs, trees, and grids.   
e.The Ext JS framework includes support for [Ajax](http://madesu.eu/jquery-tutorial-for-beginners/) 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

3. What are major Web browsers supported by Ext JS framework?  
\* 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).

4. Integration of Web development server-side frameworks with Ext JS?  
You can use Ext JS with other common Web development server-side frameworks, including [PHP](http://madesu.eu/highlights-of-cakephp-an-efficient-php-framework/), the Java™ language, Microsoft® .[NET](http://madesu.eu/latest-top-jquery-interview-questions-and-answers-2013/" \o "See also Latest Top jQuery Interview Questions and Answers 2013), Ruby on Rails, and ColdFusion.

5. Where Extjs extended from ?  
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.

6. Extjs [Ajax](http://madesu.eu/jquery-ajax/) implementation?  
A typical Ext JS [Ajax](http://madesu.eu/six-best-web-frameworks-in-java/) implementation: an [HTML](http://madesu.eu/jquery-selectors-examples-to-find-elements-in-dom-id-class-element-descendent-child-multiple-and-pseudo-selector/) text field and button element that posts data in the text field to a Web server when the button is clicked.

7.Do you have any advice for developers using Ext for the first time?  
Ext can be used by Web Application developers who are familiar with [HTML](http://madesu.eu/selecting-elements-returned-from-jquery-ajax-response-strings-explanation-with-an-example/) 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.

8. How to access [Dom](http://madesu.eu/extjs-component-life-cycle/) element using EXTJS?  
The Element [API](http://madesu.eu/latest-top-javascript-interview-questions-and-answers-2013/) is fundamental to the entire Ext library.  
Using traditional Javascript, selecting a [DOM](http://madesu.eu/jquery-document-ready-handler-with-example/) node by [ID](http://madesu.eu/getting-tweets-to-fade-onto-the-screen/) is done like this:  
var myDiv = document.getElementById(‘myDiv’);  
Using Extjs:  
Ext.onReady(function() {  
var myDiv = Ext.get(‘myDiv’);  
});

9. what is the purpose of Element Object in Extjs?  
->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](http://madesu.eu/latest-jquery-interview-questions-and-answers/) classes, add/remove event handlers, positioning, sizing, animation, drag/drop, etc.)

10. what is syntax for Extjs Button click event?  
Ext.onReady(function() {  
Ext.get(‘myButton’).on(‘click’, function(){  
alert(“You clicked the button”);  
});  
});

11. what is use of Ext.onReady() function ?  
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(“Congratulations! You have Ext configured correctly!”);  
});

12. For example, to show our message when any paragraph in our test page is clicked, what is the extjs code on paragraph click?  
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);  
});

13. List out the extjs library files to include in JSP page?  
ext-base.js  
ext-all-debug.js or ext-all.js  
ext-all.[css](http://madesu.eu/apply-css-to-controls-using-jquery/)  
base.[css](http://madesu.eu/extjs-creating-custom-xtypes-example-1-img-with-easy-to-use-controls/) or examples.css

14. List out the css file required to apply Extjs Theme property?  
xtheme-gray.css  
ext-all.css

15. what is purpose of MessageBox?  
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();

16. write syntax for MessageBox show() method?  
Ext.MessageBox.show({  
title: ‘Paragraph Clicked’,  
msg: ‘User clicked on Paragraph’,  
width:400,  
buttons: Ext.MessageBox.OK,  
animEl: paragraph  
});

17. what is method to Update the message box body text for MessageBox?  
updateText( [String text] ) : Ext.MessageBox

18. what is a widget?  
A widget is a tiny piece or component of functionality.

19.what is parent class for all stores in extjs? how many stores exists?  
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

20. How to handle event for a extjs component?  
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 ){ ….. });

21. How to find no of records in a store?  
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.

22. How to handle exception while loading datastore?  
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();   
});

23. how to handle updates for store changes?  
use store.commitChanges()

24. what is the purpose of each() in store?  
Calls the specified function for each of the Records in the cache  
each( Function fn, [records Object] )

25. how to get modified records using store object?  
store.getModifiedRecords() : Gets all records modified since the last commit.

26. how to get record using index?  
store.getAt( Number index ) : Get the Record at the specified index.

27. how to get record using id?  
store.getById( String id ) : Get the Record with the specified id.

28. what is the purpose of load() in store?   
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});

29. what is purpose of loadData() in store?  
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.

30. How many types of layout managers exist in extjs?what are they?   
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](http://madesu.eu/prado-php-framework/) table as the layout container.you want to allow the contents to flow naturally based on standard browser table layout rules.

31. How we can apply pagination in grid panel ?  
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}});

32. what is xtype?  
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

33. what is vtype?  
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](http://madesu.eu/magento-steps-to-add-custom-tabs-to-the-product-admin/" \o "See also Magento - Steps to add Custom Tabs to the Product Admin) validation function returns false

34. store.getModifiedRecords(): returns array object

35.how to get record object from store:  
var record = grid.getStore().getAt(rowIndex);

36. purpose of Load mask?  
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();

37. purpose of renderer in grid panel?  
using config option,   
renderer: fnCellColor where fnCellColor is method to apply color to a cell.

38. how to get selection model used in a grid panel?  
using grid.getSelectionModel(); method

39. how to stop editing a record?  
newRecord.endEdit();

40. how to start editing a record?  
newRecord.beginEdit();

41. how to commit a record modification?  
newRecord.commit();

42. what is use of combo select event function?  
To get the selected value from a combo.using getvalue();  
var selectedComboValue = mycombo1.getValue();

43. how to get a value of textfield or combo box?  
using getvalue();  
var selectedValue = mytextfield.getValue();

44. how to apply css on select of combo box?  
using config option as  
emptyClass : ‘emptycss’, where emptycss is a css classname

45. what are components required for grid panel?  
store, columnmodel, id, width,height  
46. how to disable menu option for header in columnModel?  
using menuDisabled: true

47. how to hide the column in grid panel?  
using hidden : true

48. How to register callbacks to the load and exception events of the JsonStore?   
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  
},

[...]  
}

49. extjs decode() ?  
var [json](http://madesu.eu/extjs-gui-versions/" \o "See also EXTJS GUI  Versions) = Ext.decode(response.responseText);  
Ext.Msg.alert(‘Error’, json.error);

FAQ links:  
[http](http://madesu.eu/ember-js-mvc-tutorial/)://www.extjs.com/learn/Tutorial:Introduction\_to\_Ext\_2.0  
http://blogs.yellowfish.biz/tag/extjs/  
Data Grid:  
http://loianegroner.com/2009/12/getting-started-with-extjs-datagrid/  
http://loianegroner.com/2010/03/extjs-and-spring-[mvc](http://madesu.eu/commonly-asked-wpf-interview-questions-and-answers-part-1/)-framework-crud-datagrid-example/

Extjs FAQ:  
—————————————  
1.what is maximum size of http post request ?   
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.

2.Why did you choose Ext JS?  
http://blogs.yellowfish.biz/tag/extjs/  
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 ‘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.

3.What features could we add to Ext to make building a rich application like PLANet easier in the future?  
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.

4.Do you have any advice for developers using Ext for the first time?  
Ext can be used by Web Application developers who are familiar with [HTML](http://madesu.eu/three-time-saving-tips-you-didnt-know-in-testing-wicket-pages/) 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:

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

9. ExtJS as a framework wrapping around widgets/constrols.  
Alternatives:  
Backbase Ajax Framework,jQuery,qooxdoo  
http://qooxdoo.org/demo  
http://yuilibrary.com/gallery/  
http://www.smartclient.com/

10.Extjs Vs jQuery:  
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.

11. Namespace importance:  
http://higher-order.blogspot.com/2008/02/designing-clientserver-web-applications.html  
http://jquery-howto.blogspot.com/2009/01/namespace-your-javascript-function-and.html  
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)

12. Handle load exception while datastore?  
13. How pagination works in grid panel?

14. extjs performance issues/ memory leak problems?

15.where to use anchor layout?

16. Do ie6 supports extjs2.2?  
17. A widget is a tiny piece or component of functionality.

About EXTJS ver.2.2 :   
—————————————————–  
- 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.

**Requirements for a simple EXT-JS**  
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.

**EXTJS Features**:

Quick tips  
status bar  
integration with google maps   
Grids

Yes No cancel prompt  
progress dialog  
alert  
icon dialog

Dialog  
- dependent dialog  
- independent  
- splash

Layouts  
Border Layout is drawn from nested layout panel.  
- Border Layout  
- Form Layout  
- Anchor Layout –> flexible form resizing.  
- Flow Layout –> Building of cool bar.

**EXTJS components:**  
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

Benefits of ExtJS:  
——————————————————  
Why did you choose Ext JS?  
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 ‘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.

Get to know Ext JS

Ext JS began as a project to extend the functionality that the YUI Library offered. A key aspect of the YUI Library is the cross-browser support, which you’ll also find in Ext JS. This support allows you to build Web applications without worrying about the target browser.

Ext JS provides excellent performance. The framework is fully object oriented and extensible. Because it’s written in the JavaScript language, Ext JS’s features are ready to use after you download and install it.

Ext JS browser support:

The Ext JS framework is supported in all the major Web browsers, including:

\* 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).

Design patterns and Ext JS

Developers should appreciate the well-thought-out design and implementation of Ext JS. Its object-oriented design patterns influence the relationship and interactions between objects. According to Ext JS, the design patterns used in their development were highly influenced by the book, Head First Design Patterns, by Freeman and Freeman (see Resources). Developers looking at the Ext JS source code will find creational patterns, including the singleton design pattern; structural patterns, including the flyweight design pattern; and behavioral patterns, including the observer pattern.

Build rich Internet applications with Ext JS:

Ext JS provides numerous UI elements that are essential to developing rich Internet applications (RIAs). Ext JS includes controls such as message boxes, combo boxes, data grids, and toolbars. In addition, layout managers allow you to specify how elements are displayed on a page. Additional features are available for working with forms and windows.

The include order for the JavaScript files can change if you are using other frameworks. However, Ext JS is typically included in your Web application, assuming that you’ve installed Ext JS in the lib/ext directory on your Web server.

UI elements:

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.

Integration of Ext JS;  
You can use Ext JS with other common Web development server-side frameworks, including [PHP](http://madesu.eu/extjs-creating-new-data-in-models-or-views/), the Java™ language, Microsoft® .[NET](http://madesu.eu/latest-mvc-interview-questions-and-answers-2013/), Ruby on Rails, and ColdFusion.

**Ext JS and Ajax:**

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.

Ext JS integration with other Web server frameworks:

You can use Ext JS with other common Web development server-side frameworks, including [PHP](http://madesu.eu/zend-framework-mvc-introduction-in-php/), the Java language, Microsoft .[NET](http://madesu.eu/asp-net-4-frequently-asked-interview-questions-answers/" \o "See also asp.net 4 frequently asked interview Questions Answers), Ruby on Rails, and ColdFusion. For integration specifics for each of these frameworks.

Ext JS development tools

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.

Conclusion

Web development frameworks often promise to simplify and speed application development, but many fall short of that goal. Ext JS keeps its promise with an easy-to-use development model. The latest release of Ext JS — version 3.0 — shows that it is committed to evolving and remaining a cornerstone of RIA development.