

An Introduction to EXT JS

Hamid M. Porasl

A Simple HelloWorld application in Ext JS	3
1. Download Ext JS source	3
2. Download Sencha SDK Tools.....	3
3. Create folder "helloWorld" on your root application Server or http Server	3
4. Start your httpServer or application server	4
Deployment.....	5
5. Create jsb3 object	5
6. Build your application	5
7. Create deployment index.html file	5

A Simple HelloWorld application in Ext JS

1. Download Ext JS source

<http://www.sencha.com/products/extjs/download> and unzip it under root of your http Server, e.g. for JBOSS: \$JBOSS_HOME\server\default\deploy\ROOT.war

2. Download Sencha SDK Tools

from <http://www.sencha.com/products/sdk-tools/download> . Run the executable then put

C:\Program Files (x86)\SenchaSDKTools-1.2.3\command in your class path

3. Create folder “helloWorld” on your root application Server or http Server

We need to create 2 files “myHelloWorld.js” and “index.html”

Index.html:

```
<html>
  <head>
    <title>Hello World</title>
    <link rel="stylesheet" type="text/css" href="../extjs/resources/css/ext-all.css">
    <script type="text/javascript"
      src="../extjs/ext-debug.js"></script>
    <script type="text/javascript" src="myHelloWorld.js"></script>
  </head>
  <body></body>
</html>
```

myHelloWorld.js

```
Ext.application({
  name: 'HelloWold',
```

```
        launch: function() {
Ext.create('Ext.container.Viewport', {
    layout: 'fit',
    items: [
        {
            title: 'Hello World',
            html : 'Hello! Welcome to my Hello World implementation in Ext JS.'
        }
    ]
});
    }
});
```

4. Start your httpServer or application server

JBOSS: <http://127.0.0.1:8081/helloWorld>

Apache Server : <http://localhost:8080/helloWorld>

You should see following text displayed:

“Hello! Welcome to my Hello World implementation in Ext JS.”

Is there a problem? Yes. You had to wait some second before application gets loaded. Why?

Your Ext JS application will load a lot of unnecessary scripts and libraries. We have to avoid it.

We have to tell which resources are required ahead. Put following statement on top of your javaScript file:

```
Ext.require('Ext.container.Viewport');
```

This will ensure that the file containing the code for Ext.container.Viewport is loaded before the application runs.

If you re-run application you see it is faster now, but not good-enough. Still application tries to load unnecessary resources. You have to deploy your Ext JS

Deployment

5. Create jsb3 object

```
sencha create jsb -a index.html -p myApplication.jsb3
```

Now a new jsb3 object is created in your application folder, i.e. myApplication.jsb3

6. Build your application

```
sencha build -p app.jsb3 -d .
```

Now you will find 2 new js files in your directory, i.e. **myHelloWorld-all.js** and **all-classes.js**

7. Create deployment index.html file

Now rename your index.html file to index_prod.html and use new generated js files as bellow

```
<html>
  <head>
    <title>Hello World</title>
    <link rel="stylesheet" type="text/css" href="../extjs/resources/css/ext-all.css">
    <script type="text/javascript"
      src="../extjs/ext.js"></script>
    <script type="text/javascript"
src="myHelloWorld-all.js"></script>
  </head>
  <body></body>
</html>
```

What is changed? Two new java script files are used in index_prod.html

Script ext.js is used instead of ext-debug.js which is used only for development

New myHelloWorld-all.js script is used instead of old myHelloWorld.js script that we wrote in the beginning of this introduction. Generated myHelloWorld-all.js script contains compiler generated low level javaScript code that can be run very fast on our web pages

Run production code on your server:

JBOSS: http://127.0.0.1:8081/helloWorld/index_prod.html

Apache Server : http://localhost:8080/helloWorld/index_prod.html

Wow it is getting loaded very fast