An implementation of Tree Panel component in EXT JS 4.0

By Hamid M. Porasl

This implementation contains an HTML file that t is used to invoke used EXT JS java script files and our implemented java Script to show Tree Panel.

Data is prepared at backend by a Servlet/php file that reads file hierarchy daya and passes it as JSON data to EXT JS classes. Format of passed data is stored in a Node structures.

1. HTML file:

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
    <title>Tree Example</title>
    <link rel="stylesheet" type="text/css" href="../extjs/resources/css/ext-
all.css" />
    <link rel="stylesheet" type="text/css" href="../shared/example.css" />

    <script type="text/javascript" src="../extjs/bootstrap.js"></script>
    <script type="text/javascript" src="panelList.js"></script>
</head>
<body>

    <div id="tree-div"></div>
</body>
</html>
```

Html file used to invoke EXT JS and implementation java script file

2. `panelList.js`

```
Ext.require([
    'Ext.tree.*',
    'Ext.data.*',
    'Ext.tip.*'
]);

Ext.onReady(function() {
    Ext.QuickTips.init();

    var store = Ext.create('Ext.data.TreeStore', {
        proxy: {
            type: 'ajax',
            url: '/TreeLoader/TreeServlet?node=11' // location of Data
            //url: 'get-nodes.php'
        },
        root: {
            text: 'Root Directory',
            id: 'src',
            expanded: true
        },
        folderSort: true,
        sorters: [{
            property: 'text',
            direction: 'ASC'
        }]
    });

    var tree = Ext.create('Ext.tree.Panel', {
        store: store,
        viewConfig: {
            plugins: {
                ptype: 'treeviewdragdrop'
            }
        },
        renderTo: 'tree-div',
        height: 300,
        width: 250,
        title: 'Files',
        useArrows: true,
        dockedItems: [{
            xtype: 'toolbar',
            items: [{
                text: 'Expand All',
                handler: function(){
                    tree.expandAll();
                }
            }, {
                text: 'Collapse All',
                handler: function(){
                    tree.collapseAll();
                }
            }]
        }]
    });
});
```

Implementation of JS Tree Panel

3. Tree Servlet

```
package com.extjs.treeLoader;

import java.io.IOException;

import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

import org.apache.tomcat.util.json.JSONArray;
import org.apache.tomcat.util.json.JSONException;

public class TreeServlet extends HttpServlet
{
	/**
	 *
	 */
	private static final long serialVersionUID = 1L;


	protected void doPost(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {
		int node = Integer.parseInt(request.getParameter("node"));
		System.out.println("node: " + node); // just for debug purposes

		try
		{
			String jsonResult = createTree(node).toString();
			System.out.println("result: " + jsonResult);
			response.getOutputStream().write(jsonResult.getBytes());
		}
		catch (JSONException e)
		{
			throw new ServletException(e);
		}
	}

	protected void doGet(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {
		doPost(request, response);
	}
```

```
	private JSONArray createTree(int node) throws JSONException {
		if (node == 0)
		{
			JSONArray array = new JSONArray();
			Node guitars = new Node(1, "Guitars");
			guitars.setLeaf(false);
			array.put(guitars.toJson());

			Node violins = new Node(2, "Violins");
			array.put(violins.toJson());

			return array;
		}

		if (node == 1)
		{
			JSONArray array = new JSONArray();

			Node acoustic = new Node(11, "Acoustic");
			acoustic.setLeaf(false);
			array.put(acoustic.toJson());

			Node electric = new Node(12, "Electric");
			electric.setLeaf(false);
			array.put(electric.toJson());

			Node semihollow = new Node(13, "Semihollow");
			array.put(semihollow.toJson());

			return array;
		}

		if (node == 11)
		{
			JSONArray array = new JSONArray();

			Node taylor = new Node(110, "Taylor");
			array.put(taylor.toJson());

			Node martin = new Node(111, "Martin");
			array.put(martin.toJson());

			Node alvarez = new Node(112, "Alvarez Yairi");
			array.put(alvarez.toJson());

			return array;
		}
```

```
                if (node == 12)
                {
                        JSONArray array = new JSONArray();

                        Node fender = new Node(120, "Fender");
                        array.put(fender.toJson());

                        Node gibson = new Node(121, "Gibson");
                        array.put(gibson.toJson());

                        return array;
                }

                return null;
        }
```

This servlet returns JSON objects that are passed to client Java Script

4. Modification in web.xml file

   Add followings to web.xml file

```
<servlet>
<servlet-name>TreeServlet</servlet-name>
<servlet-class>com.extjs.treeLoader.TreeServlet</servlet-class>
</servlet>
<servlet-mapping>
    <servlet-name>TreeServlet</servlet-name>
    <url-pattern>/TreeServlet</url-pattern>
  </servlet-mapping>
```

5. Class Node

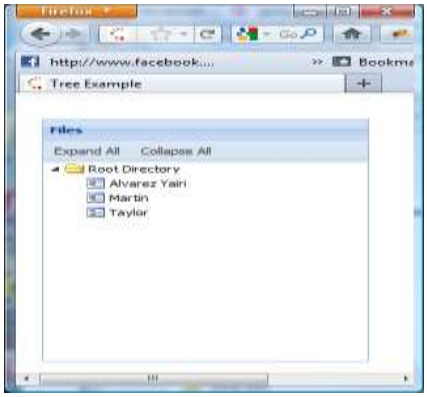
   This class is used to store file hierarchy data

```
package com.extjs.treeLoader;

import java.util.ArrayList;
import java.util.List;

import org.apache.tomcat.util.json.JSONArray;
import org.apache.tomcat.util.json.JSONException;
import org.apache.tomcat.util.json.JSONObject;

public class Node {

        private Integer id;
        private String text;
        private boolean leaf = true;
        private List<Node> children = new ArrayList<Node>();

        public Node(Integer id, String text) {
```

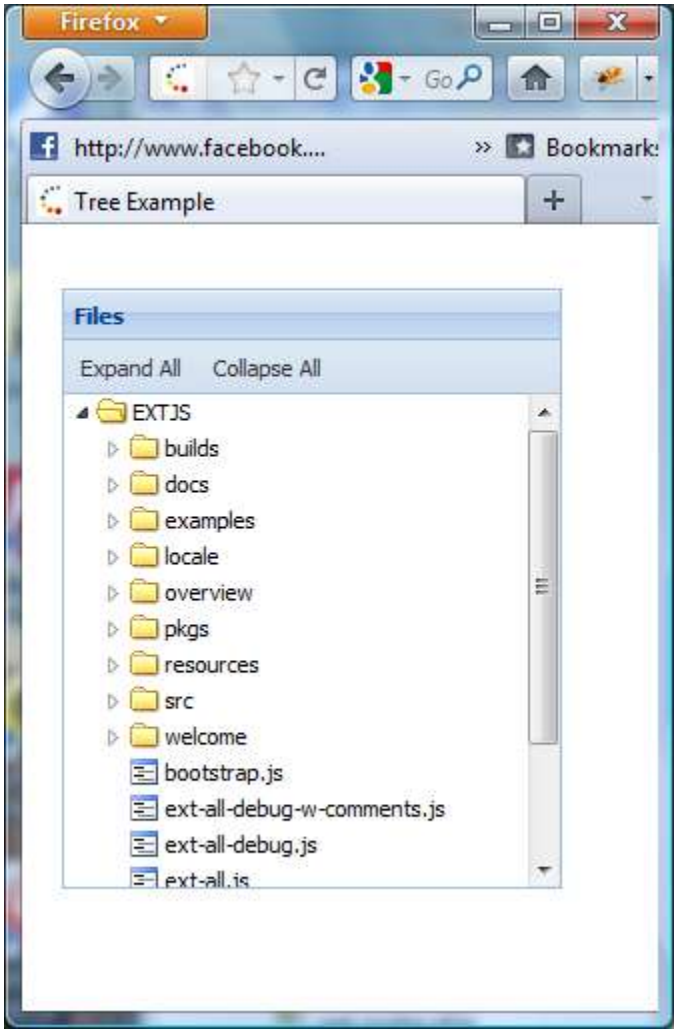
```
                this.id = id;
                this.text = text;
        }

        public JSONObject toJson() throws JSONException {
                JSONObject json = new JSONObject();
                json.put("id", id);
                json.put("text", text);
                json.put("leaf", leaf);

                if (children.isEmpty())
                        return json;

                JSONArray jsonArray = new JSONArray();
                for (Node child : children)
                        jsonArray.put(child.toJson());

                json.put("children", jsonArray);
                return json;
        }
        public void addChild(Node node) {
                leaf = false;
                children.add(node);
        }

        public Integer getId() {
                return id;
        }
        public void setId(Integer id) {
                this.id = id;
        }
        public String getText() {
                return text;
        }
        public void setText(String text) {
                this.text = text;
        }
        public boolean isLeaf() {
                return leaf;
        }
        public void setLeaf(boolean leaf) {
                this.leaf = leaf;
        }

}
```

Class Node is used to store Data

And if I run my code in JBOSS I get following tree panel displayed:



Following displayed EXT JS Tree Panel demonstrates EXT JS Tree.



Ext JS Tree Panel

Above mentioned tree is displayed using following Jason Data:


```
[{"text":"ext-debug.js","id":"\/ext-debug.js","leaf":true,"cls":"file"},
{"text":"resources","id":"\/resources","cls":"folder"},
{"text":"ext-all.js","id":"\/ext-all.js","leaf":true,"cls":"file"},
{"text":"locale","id":"\/locale","cls":"folder"},
{"text":"bootstrap.js","id":"\/bootstrap.js","leaf":true,"cls":"file"},
{"text":"welcome","id":"\/welcome","cls":"folder"},
{"text":"examples","id":"\/examples","cls":"folder"},
{"text":"ext-all-debug.js","id":"\/ext-all-debug.js","leaf":true,"cls":"file"},
{"text":"docs","id":"\/docs","cls":"folder"},
{"text":"src","id":"\/src","cls":"folder"},
{"text":"overview","id":"\/overview","cls":"folder"},
{"text":"ext-all-debug-w-comments.js","id":"\/ext-all-debug-w-
comments.js","leaf":true,"cls":"file"},
{"text":"license.txt","id":"\/license.txt","leaf":true,"cls":"file"},
{"text":"index.html","id":"\/index.html","leaf":true,"cls":"file"},
{"text":"pkgs","id":"\/pkgs","cls":"folder"},
{"text":"release-notes.html","id":"\/release-notes.html","leaf":true,"cls":"file"},
{"text":"ext.js","id":"\/ext.js","leaf":true,"cls":"file"},
{"text":"builds","id":"\/builds","cls":"folder"}]
```


References:

http://dev.sencha.com/deploy/ext-4.0.2a/examples/tree/reorder.html