EWD tags + Sencha ExtJs 4

CSP-toolkit for tag-based development with the ExtJs framework

EWD ?

- EWD :
  - “Enterprise Web Development”
  - By Rob Tweed (mgateway.com)
- Different flavours :
  - EWD.js :
    - Node.js + Javascript
  - EWD tag-based development (“classic”)
    - EWD tags (enhanced CSP)
    - Custom tags : to integrate GUI-frameworks
    - Brings AJAX to CSP

EWD tags = “CSP Toolkit”

- EWD development
  - Import EWD routines
  - ***With ExtJS*** : copy ExtJS files to csp folder
  - Define settings (source folder, csp folder, …)
  - Create EWD files (with html-like tags)
  - Compile to CSP pages

EWD fragment workflow

- onBeforeRender
  - Caché routine or classmethod to prepare data
  - Put data in the CSP-Session
- CSP-page is rendered (server)
  - Session data is put in place
- CSP-page is shown (client)
  - Javascript handle events

Example EWD fragment

```
<ext4:fragment onBeforeRender="##class(itm.Item).fItemsGrid">
    <ext4:gridPanel  id="#tmp_GridId" title="#tmp_GridTitle" sessionName="gaItems"
    columnDefinition="gcItems" storeId="#tmp_GridStore">
          <ext4:toolbar id="#tmp_Id1" dock="top">
                    <ext4:button id="#tmp_Id2" text="#tmp_BtnText" iconCls="icon-add"
    scope="this" nextPage="fItemDetails" nvp="newItem=1&catId=<?= #catId ?>"
    addTo="mainTab0"/>
          </ext4:toolbar>
    </ext4:gridPanel>
    <ext4:js at="bottom">
          function DeleteItem(grid, rowIndex, colIndex,pGridId,pCategoryId) {
                    var ans=confirm('<?= #tmp_Question ?>');
                    if (ans==true) {
                              var rowNbr=EWD.ext4.getGridRowNo(grid,rowIndex);
                              var
    nvp='rowNbr='+rowNbr+'&tabId='+Ext.getCmp('mainTabs').getActiveTab()+'&gridId='+p
    GridId+'&parentId='+Ext.getCmp(pGridId).ownerCt.getId()+'&catId='+pCategoryId;
                              EWD.ajax.getPage({page:'fItemDelete',nvp:nvp})
                    };
          };
    </ext4:js>
</ext4:fragment>
```

Example Compiled to CSP

```
<csp:if condition="$g(Error)=&quot;&quot;" />
<script language="cache" runat="server">
 i $g(sessid)="" s sessid="unknown"</script>
<pre id="ewdscript">
   <script language="cache" runat="server">
 d
    writeGridStore^%zewdExt4Code("gaItems","gcItems",$$getSessionValue^%zewdAPI("tmp_GridId",sessid),$
    $getSessionValue^%zewdAPI("tmp_GridStore",sessid),"",sessid)   </script>
Ext.create("Ext.grid.Panel",{columns:EWD.ext4.grid['#($$getSessionValue^%zewdAPI("tmp_GridId",sessid))
    #'].cols,id:"#($$getSessionValue^%zewdAPI("tmp_GridId",sessid))#",store:#($$getSessionValue^%zewdA
    PI("tmp_GridStore",sessid))#,title:"#($$getSessionValue^%zewdAPI("tmp_GridTitle",sessid))#",docked
    Items:[{dock:"top",id:"#($$getSessionValue^%zewdAPI("tmp_Id1",sessid))#",xtype:"toolbar",items:[{i
    conCls:"icon-
    add",id:"#($$getSessionValue^%zewdAPI("tmp_Id2",sessid))#",scope:this,text:"#($$getSessionValue^%z
    ewdAPI("tmp_BtnText",sessid))#",xtype:"button",handler:function(me) {var
    nvp='newItem=1&catId=#($$getSessionValue^%zewdAPI("catId",sessid))#';nvp=nvp+'&ext4_addTo=mainTab0
    ';
EWD.ajax.getPage({page:'fItemDetails',nvp:nvp});}}
]
}
]
}
);var addTo='#($$getSessionValue^%zewdAPI("tmp_addTo",sessid))#';
var remove='#($$getSessionValue^%zewdAPI("tmp_removeAll",sessid))#';
if (remove === 'true')
    Ext.getCmp('#($$getSessionValue^%zewdAPI("tmp_addTo",sessid))#').removeAll(true);
if (addTo !== '')
```

Pros & Cons

- Pro
  - Structured
  - Javascript Code generation
- Contra
  - Rigid, not flexible
    (Component structure at compile time, reduced Javascript possibilities,way of refering to components)
  - Session variables do not fit everywhere
  - Some configuration properties have incorrect CamelCase (Javascript = case-sensitive!)
  - No use of Sencha ExtJS tools and structure

My (current) solution

- Caché code generates Javascript
  - EWD to bring AJAX to csp
    - Delivers Javascript that is evaluated & runned
    - Delivers JSON objects
  - Caché classes and objects to define ExtJs components
    - Components + configuration settings
    - Structure

My (future) solution

- Sencha ExtJS framework
  - Use existing tools, structure, components
  - Define own components (“classes” in Javascript files)
  - Interact with Caché database by use of REST calls
- Caché generates Javascript
  - Components + configuration settings
  - Structure

  } Caché objects

  - Components as JSON objects

```
{xtype:”grid”, columns:[{name:”ID”,dataIndex;”Oid”},{ … }]}
```

Conclusion

In my humble opinion :

- Do not use “classic” tag-based EWD with a javascript framework
- Check out EWD.js and/or Caché REST functionality