XUL templates

Laurent Jouanneau
Course on Mozilla Education and Technologies @ Evry
December, 2011

Définition

- It allows to generate XUL content, from a datasource
- Datasource :
  - XML content
  - SQLite database
  - RDF content
  - Any other datasource if you create specific XPCOM components
- Generated content will be create as children of an element with the « datasources » attribute
- The « querytype » attribute indicate the type of datasource: « xml », « storage » (sqlite)

Template with an XML datasource

- The <template> element indicate the content to generate for each « record » of the datasource

```
A sample.xml file:
<people>
  <person name="Napoleon Bonaparte" gender="male"/>
  <person name="Cleopatra" gender="female"/>
</people>


In XUL:
<listbox datasources="sample.xml" querytype="xml" ref="*">
  <template>
    <listitem label="?name"/>
  </template>
</listbox>
```

  - For each <person> element, a listitem element is created, with a label which have the value of the attribute « name » of the xml element of the datasource

Template with a dynamic XML datasource

- If the XML datasource is not in a file, use builder.datasource to indicate the root element of the XML DOM
- The attribute « datasources » should be empty (and present)
- Example with XML content created « from scratch »

```
var parser = Components.classes["@mozilla.org/xmlextras/domparser;1"]
     .createInstance(Components.interfaces.nsIDOMParser);

var myXml = '<people>person name="Napoleon Bonaparte" gender="male"/>';
myXml += '<person name="Cleopatra" gender="female"/></people>';

// myDatasource is a DOMDocument
var myDatasource = parser.parseFromString(myXml,"text/xml");


// assign the datasource to the template builder
theListBox.builder.datasource = myDatasource.documentElement;
```

Query and action

- A template can contain a <query> element, indicating which part of the datasource will be used to generate the content
  - For XML: a Xpath expression selecting xml nodes
  - For SQLite: the SQL query to retrieve records
- Queries are prepared during the creation of the container element. Generated content is built when the container is visible.

```
<listbox datasources="profile:userdata.sqlite" ref="*" querytype="storage">
  <template>
    <query>
      select name from myfriends
    </query>
    <action>
      <listitem uri="?" label="?name"/>
    </action>
  </template>
</listbox>
```

XML datasource and var assignement

- With a query for XML datasource, you can create kind of variables with the <assign> element

```
<vbox datasources="people.xml" ref="*" querytype="xml">
  <template>
    <query expr="person">
      <assign var="?namelength" expr="string-length(@name)"/>
      <assign var="?siblings" expr="count(../*) - 1"/>
    </query>
    <action>
      <hbox uri="?" align="center">
        <button label="?name"/>
        <label value="?gender"/>
        <label value="?namelength"/>
        <label value="?siblings"/>
      </hbox>
    </action>
  </template>
</vbox>
```

SQLite datasource and parameters

- With the <param> elements, you can indicate parameters to a prepared SQL query

```
<listbox id="friends" datasources="profile:userdata.sqlite" ref="*" querytype="storage">
  <template>
    <query>
      select name, from myfriends where age >= :minage && age <= :maxage
      <param id="minage" name="minage" type="integer">30</param>
      <param id="maxage" name="maxage" type="integer">40</param>
    </query>
    <action>
      <listitem uri="?" label="?name"/>
    </action>
  </template>
</listbox>
```

```
document.getElementById("minage").textContent = min;
document.getElementById("maxage").textContent = max;
document.getElementById("friends").builder.rebuild();
```