

Kompozit alkalmazások fejlesztése

IBM WebSphere Portal Server

Portletek - összefoglaló

- Portlet: portálba integrálható kisalkalmazás
- Szabványok
 - JSR 168 – Portlet 1.0
 - (IBM API)
 - JSR 286 – Portlet 2.0
- Dióhéjban
 - doView() / doEdit() / ..
 - processAction()
 - (processEvent())

Portletek közötti kommunikáció

- **Állapot-megosztás**
 - Shared Session (JSR 168)
 - Public Render Parameters (JSR 286)
- **Esemény-alapú**
 - JSR 286 Events
 - (IBM API Events)
 - Live Text & Click-to-action (WebSphere Portal)
 - Drag and Drop (WebSphere Portal)
- **JavaScript**

Shared Session

- Már JSR 168-ban is
- Csak egy .war-on belüli portletek közt
- Java kód:
 - `request.getPortletSession().setAttribute("searchResults", customers, PortletSession.APPLICATION_SCOPE);`
 - `Object searchString = request.getPortletSession().getAttribute("searchString", PortletSession.APPLICATION_SCOPE);`
 - `request.getPortletSession().removeAttribute("searchString", PortletSession.APPLICATION_SCOPE);`
- JSP Scriptlet:
 - `<%=renderRequest.getPortletSession().getAttribute("searchString", PortletSession.APPLICATION_SCOPE) %>`
- JSP EL: közvetlenül nem sikerült; megoldás:
 - Portlet-kód: `request.setAttribute("searchString ", request.getPortletSession().getAttribute("searchString"));`
 - JSP EL: ``${requestScope.searchString}`

Public Render Parameters

- Csak JSR 286-tól
- Akár különböző .war-okból származó portletek között is
- String paraméterek, a következő action törli őket.
- Kétféle képpen állítható be
 - url-ben megjelenik (bookmarkolható, linkelhető), a `<portlet:renderURL>` jsp tag segítségével:

```
<a href="<portlet:renderURL>  
    <portlet:param name="customerId" value="${customer.id}" />  
    </portlet:renderURL>">${customer.familyName} ${customer.firstName}</a>
```
 - Az url-ben nem jelenik meg: action-t hívunk és az azt kezelő java kódban `actionResponse.setRenderParameter(„customerId”, customer.getId());`
- Lekérdezés
 - Pre-render portlet kódban: `String id = request.getParameter("customerId");`
 - JSP EL: `${param.customerId}`

JSR 286 Events

- Csak JSR 286-tól
- Az portletek eseményeket generálhatnak, melyekre más portletek reagálhatnak: processEvent() fázis
- A portál adminisztrátornak össze kell drótoznia hogy melyik portletek melyik más portletek melyik eseményeire reagáljanak (akár oldalak közt is)
- Szükséges xml definíciók a portlet.xml-be:

```
<event-definition>
  <name>AirportCodeSelect</name>
  <value-type>com.example.EventPayload</value-type>
</event-definition>
<portlet>
  ...
  <supported-publishing-event>
    <name>AirportCodeSelect</name>
  </supported-publishing-event>
</portlet>
```

```
<portlet>
  ...
  <supported-processing -event>
    <name>AirportCodeSelect</name>
  </supported-processing -event>
</portlet>
```

JSR 286 Events

- Event objektum
 - Lehet String, vagy bármilyen Szerializálható osztály – összetett adatokat is tartalmazhat
 - ha WSRP-n keresztül is akarjuk használni a portleteket akkor JAXB szerializálhatóság is kell
- Küldés
 - `processAction()/processEvent(): response.setEvent(name,payload)`
- Fogadás
 - `processEvent(..) { Event payload = request.getEvent(); ... }`
- Trükk
 - ha .war-ok közt használunk egy eventet, akkor érdemes lehet shared lib-be tenni az event osztályt, különben a war-onként különböző classloader miatt szerializálást kell végrehajtani, ami lassú.

WPS Live Text & Click-to-action

- WPS V6.1 óta
 - http://publib.boulder.ibm.com/infocenter/wpdoc/v6r1/index.jsp?topic=/com.ibm.wp.ent.doc_v6101/dev/w2_smtg.html
- Működéséhez kell a Live Object Framework app
- Kommunikáció a kliensen JavaScriptben, nem kell a portlet.xml-be semmi
- Live Text: Semantic Tagging
 - Bármilyen (nem csak portlet!) html tartalom tag-elhető Live Text-ként (c2a:source class)
 - Beépített JavaScript felismeri a tag-et és kattintásra felugró ablakban prezentálja a megadott tartalmat
- Click-to-action:
 - Html formokat tag-elhetünk click-to-action targetként (c2a:target class)
 - Elég ha a form ugyanazon a képernyőn van; lehet másik portlet vagy tetszőleges html tartalom (content vagy a téma része is akár..)
 - A passzoló targetek felsorolásra kerülnek a source-ok felugró kisablakában, egy kattintás aktiválja őket az aktuális source-elemnek megfelelő paraméterrel – hívhatnak portlet action-t, tetszőleges szervletet, vagy akár javascriptet is.

WPS Live Text & Click-to-action

- Példa

```
<div class="c2a:source">
  <span class="c2a:typename" style="display:none">customer</span>
  <span class="c2a:value" style="display:none">${customer.id}</span>
  <%@ include file="/jsps/inc/liveTextHeaderForCustomer.jsp"%>
  <span class="c2a:anchor">more</span>
</div>
```

```
<form class="c2a:target" method="get"
  action="/myapp/do.something"
  style="display:none;">
```

```
<span class="c2a:typename">customer</span>
<p class="c2a:action-label">
  Egy szervletnek küldöm</p>
<input type="text" class="c2a:action-param"
  name="param" />
<input type="submit">
```

```
</form>
```

```
<form class="c2a:target" action="javascript:void(0)"
  style="display:none;"
  onsubmit="alert('customer id:
'+document.getElementById('clientTargetParam').value );
return false;">
```

```
<span class="c2a:typename">customer</span>
<p class="c2a:action-label">szimpla alert</p>
<input name="targetParam"
  id="clientTargetParam" type="text"
  class="c2a:action-param" />
<input type="submit">
```

```
</form>
```

WPS Drag and Drop

- WPS V6.0.1. óta
- A kommunikáció a kliensen (JavaScript)
- Nem kell közös .war, se portlet.xml deklaráció
- Használat

```
<%@ taglib uri="/WEB-INF/tld/dnd.tld" prefix="dnd" %>
```

```
<dnd:drag type=„matching" value=„value_to_send" dragClone="true">
```

```
  <dnd:additionalProperty type=„param2" value=„val" />
```

```
  ..content to be dragged..
```

```
</dnd:drag>
```

```
<dnd:drop name=„eee" action=„action_url_processing_params_eee_fff" type=„matching" >
```

```
  <dnd:additionalProperty name=„fff" type=„param2">
```

```
    .. content of the dropping target area..
```

```
</dnd:drop>
```

http://www.ibm.com/developerworks/websphere/tutorials/0707_fernandez/section2.html

Saját JavaScript

- Használhatunk különféle JavaScript framework-öket (pl. Dojo Events)
- Írhatunk saját JavaScript kódot
- Ha egyedi id-ra van szükség a **portleten belül**, akkor használjuk a portlet taglib **<portlet:namespace/>** tag-jét:
 - `<div id="<portlet:namespace/>_fontos_div"></div>`
 - Ezáltal akkor is helyesen fognak működni a javascript kódok ha a portletünk több példánya is jelen van egy adott oldalon.

Összehasonlítás

– Public render parameters

- Prezentáció összehangolása (master-detail, találatok szűrése, stb)
- Portlet 2.0 kell hozzá
- A portletleírókban meg kell említeni
- Csak Stringek
- Bekerül(het) az URL-be (bookmark-olható)

– Shared Session

- Csak .war-on belül működik
- Tetszőleges objektumok, portlet.xml-ben **nem** kell jelezni.

– JSR 286 Events

- Logika hívása (adatfeldolgozás, eseménykezelés)
- Portlet 2.0 kell hozzá
- A portletleírókban meg kell említeni
- Deploy után még össze kell drótozni a portleteket
- Tetszőleges szerializálható objektumok (WSRP-hez JAXB szerializálhatóság)
- Szerver oldali metódus hívódik az esemény hatására, hívhat további eseményeket is

– WPS Live Text – Click-to-action

- Kell a Live Objects Framework app és javascript
- További adatokat jeleníthet meg
- Hívhat szerver és kliens oldali kódot is
- Nem kell közös .war, se portlet.xml deklaráció
- Nem csak portletek közt, bármilyen html tartalomban!

– WPS Drag and Drop

- WPS V6.0.1. óta
- A drop zóna portlet-action-t hív a rádobott elemnek megfelelő paraméterekkel
- Nem kell közös .war, se portlet.xml deklaráció

– Saját JavaScript

- Korlátlan lehetőségek
- Magunkra vagyunk utalva