

14.-17.09.2009
in Nürnberg



Herbstcampus

Wissensmacher
per eLearning

Groß und mächtig Portale und Portlets (mit JavaServer Faces)

Andy Bosch

Unabhängiger Berater und Trainer

www.jsf-forum.de



Über mich

- Name: Andy Bosch
- Trotz perfektem Hochdeutsch: Schwabe aus Stuttgart
- Berater und Trainer mit Fokus auf Portlets und JavaServer Faces
- Buchautor "JavaServer Faces" und "Portlets und JSF"
- Betreiber von www.jsf-forum.de
- Herausgeber der JSF 2.0 Artikel auf www.jsf-forum.de
- Mitglied der Expert Group des JSR-301 (Portlet Bridge Specification for JSF)

Agenda

- Was ist ein Portal?
- Was leistet ein Portal / Portalsystem?
- Von Portalen zu Portlets
- Zwei Worte zu JavaServer Faces
- Portlets und JavaServer Faces

Agenda

- **Was ist ein Portal?**
- Was leistet ein Portal / Portalsystem?
- Von Portalen zu Portlets
- Zwei Worte zu JavaServer Faces
- Portlets und JavaServer Faces

Was ist eigentlich ein „Portal“?

Portalbegriff im weiteren Sinne:

- Ein Portal ist eine Webseite, auf der unterschiedliche Informationen zu einem Themengebiet zusammengefasst sind.
z.B. Anglerportal, Zauberer-Portal, Jodel-Portal
- Ein Portal ist eine zentrale Einstiegsseite in das WWW, oftmals mit redaktionell betreutem Inhalt wie bei Yahoo oder iGoogle
- „Ein Portal ist eine Webseite“

Beispiel iGoogle





Was ist eigentlich ein „Portal“?

Portalbegriff im engeren Sinne:

- Ein Portal ist eine Webapplikation, die mindestens folgende Eigenschaften besitzt:
 - Aggregation von (verschiedenen) Inhalten
 - Einheitliche Präsentation verschiedener Dienste
 - Personalisierung
 - Single Sign On
 - Security
 - ...



Portallösungen

- IBM WebSphere Portal
- Apache Jetspeed
- BEA WebLogic Portal
- SAP Netweaver Portal
- Oracle Portal
- Liferay Portal
- JBoss Portal
- Apache Pluto (eher nur PortletContainer)
- ...

Portale unterliegen keinem Standard. Dennoch bieten heutige Portallösungen ähnliche Funktionalität, allerdings ist die Umsetzung zum Teil komplett unterschiedlich.



Von Web 2.0 zu Enterprise 2.0

- Mit Web 2.0 steigen die Anforderungen des Kunden an eine Webanwendung
- Der Benutzer will aktiver eingebunden sein in Webanwendungen und diese auch mitgestalten können.
- Community-Gedanke wird immer stärker (Amazon-Empfehlungen, Xing, Hotelbewertungen, ...)
- Benutzer erwarten, dass Anwendungen auf sie „hören“ und auf sie eingehen.



Agenda

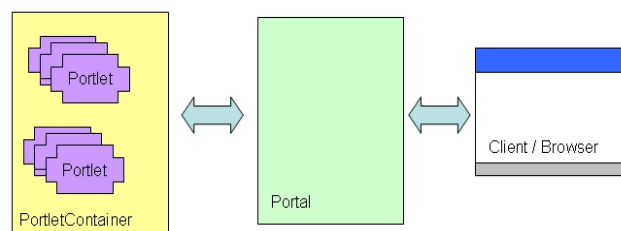
- Was ist ein Portal?
- **Was leistet ein Portal / Portalsystem?**
- Von Portalen zu Portlets
- Zwei Worte zu JavaServer Faces
- Portlets und JavaServer Faces

Aufgabe eines Portals / Portalsystems

- Ein Portal stellt bereits ein Grundgerüst zum Aufbau von Portalanwendungen bereit
 - Aggregation von Inhalten und Services
 - Single Sign On / Security
 - Personalisierung
 - Design- / Layouting Unterstützung
- Ein Portal ist der Lebensraum von Anwendungen (Portlets), die auf eine bestehende Infrastruktur aufsetzen können.
- Oftmals sind in einem Portalserver bereits viele Community-Features enthalten (Forum, Chat, ...)

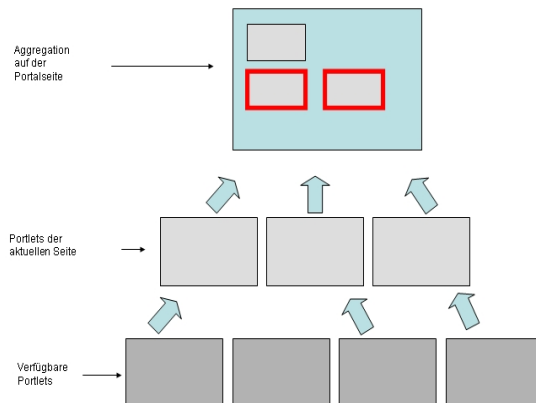
Portal und PortletContainer

- Das Portal ist die „Engine“ des kompletten Portalauftritts.
- Der PortletContainer ist der Lebensraum für Portlets
- Das Portal triggert den PortletContainer, sich darzustellen oder Aktionen auszuführen.
- Das Portal ist zuständig für die Aggregation.



Aggregation im Portal

- Das Portal sammelt die verschiedenen Fragmente ein und baut daraus eine komplette Portalseite.

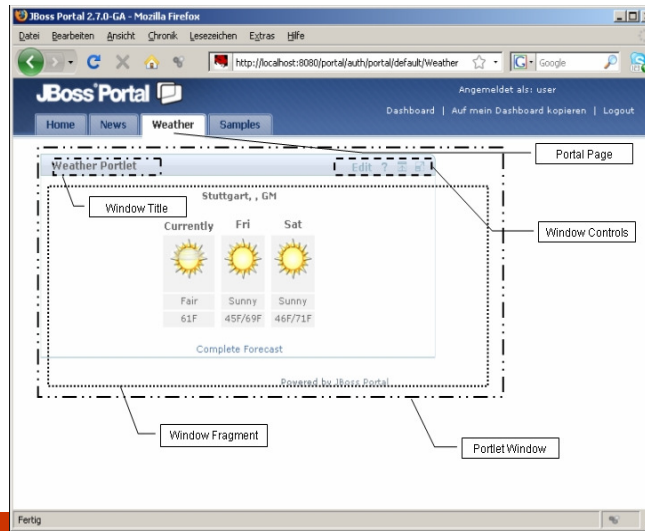


WSRP: Das Ganze auch Remote

- WSRP = Web Services for Remote Portlets
- Standard der OASIS
- Portlets „laufen“ in einem PortletContainer, werden aber in einem anderen Portal angezeigt. (=Remote)
- Beschränkt sich nicht nur auf Java, sondern technologieübergreifend
- Kann z.B. auch eingesetzt werden, um die Last eines Portals zu verteilen (Funktionalität übernimmt ein Portal, die Darstellung ein anderes)



Portalobjekte am Beispiel JBoss Portal



Personalisierung im Portal

- Eine Darstellung im Portal erfolgt z.B. Zielgruppenspezifisch.
- Sogar auf Einzelbenutzer-Ebene kann eine Personalisierung erfolgen.
- Es kann über Rechte oder User Settings eine Steuerung erfolgen.
- Eine Personalisierung kann steuern, welche Anwendungen und Inhalte sichtbar sind
- Oder: Es kann innerhalb der Anwendung / des Portlets eine bestimmte Funktionalität ermöglicht werden.

Agenda

- Was ist ein Portal?
- Was leistet ein Portal / Portalsystem?
- **Von Portalen zu Portlets**
- Zwei Worte zu JavaServer Faces
- Portlets und JSF?

Von Portalen zu Portlets

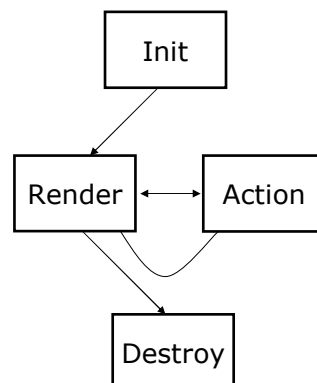
- Ein Portal **aggregiert** verschiedene Dienste, Services und Inhalte.
- Dazu wird eine Portalseite in verschiedene **Fragmente** zerlegt.
- Ein **Fragment** ist ein **Portlet**.

→ Portlets sind im Ggs. zu Portalen **standardisiert**. Im **JSR-168** ist die Funktionsweise eines Portlets genau beschrieben.
Im aktuellen **JSR-286** wurde eine weitere Version des Portletstandards erarbeitet.

Funktionsweise eines Portlets

- Ein Portlet bekommt durch den PortletContainer den Trigger, sich darzustellen (**Render-Phase**)
- Ausgaben mehrere Portlets einer Portalseite werden durch das Portal aggregiert und als (Html-)Seite ausgegeben.
- Erfolgt eine Aktion in einem Portlet, wird in diesem einen Portlet eine **Action-Phase** angestoßen. Danach werden alle Portlets neu gerendert.

Lifecycle eines Portlets (JSR-168)



Der JSR-286 (Portlet V2) wird diesen Lifecycle erweitern.

Ein triviales Portlet ...

```
public class HelloWorld extends GenericPortlet {  
  
    protected void doView( ... ) {  
        PrintWriter writer = response.getWriter();  
        writer.println( "Hallo neue Portlet Welt" );  
        writer.println( "<br><br>" );  
        writer.println( "Hallo Welt !!!" );  
        writer.println( "<br><br>" );  
    }  
  
}
```

Aufgaben des JSR-168

- Festlegung des Portlet Lifecycles
- Interfaces und Basisklassen (GenericPortlet und javax.portlet.Portlet)
- Window States und Portlet Modes
- Request- und Renderparameter
- PortletSessions
- PortletPreferences
- Einbindung von JSPs
- Portlet Tag Library

→ Fokus auf dem Portlet selbst

Aktuell: JSR-286

- Alles aus JSR-168 ist weiterhin gültig
- Public Render Parameter
- Portlet Events
- Resource Serving
- Portlet Filter
- Basis für Ajax-Integration

→ Fokus auf dem Zusammenspiel von Portlets

Fazit Portal-/Portlet-Entwicklung

- Portale bieten bereits eine Vielzahl von Basisfunktionalitäten, um die sich eine Anwendung nicht mehr kümmern muss.
- Es gibt einen Standard für Portlets (JSR-168 bzw. JSR-286)
- Viele Portale liefern bereits ein großes Set an Anwendungen mit aus (Community-Features etc)
- Dennoch ist eine Portlet-Entwicklung direkt an der API sehr aufwendig und wenig „komfortabel“.
- Ideal wäre es, die Portlet-Technologie mit einem modernen UI-Framework zu kombinieren ...

Agenda

- Was ist ein Portal?
- Was leistet ein Portal / Portalsystem?
- Von Portalen zu Portlets
- **Zwei Worte zu JavaServer Faces**
- Portlets und JSF?

Was ist JavaServer Faces?

- JavaServer Faces ist ein UI-Framework
- JavaServer Faces ist komponentenorientiert
- JavaServer Faces ist ein Standard innerhalb von JavaEE
- Für JavaServer Faces existieren mittlerweile eine (fast unüberschaubare) Anzahl von ergänzenden Komponenten, AddOns und Erweiterungen

Was bietet JavaServer Faces?

- Model-View-Controller
- Navigationskonzept
- Bean-Management
- Eventhandling
- Datenvalidierung und -konvertierung
- Große Auswahl an (OpenSource-) UI-Komponenten
- Ajax-Integrationsbibliotheken
- Templating-Engines
- ...

JSF und Komponenten

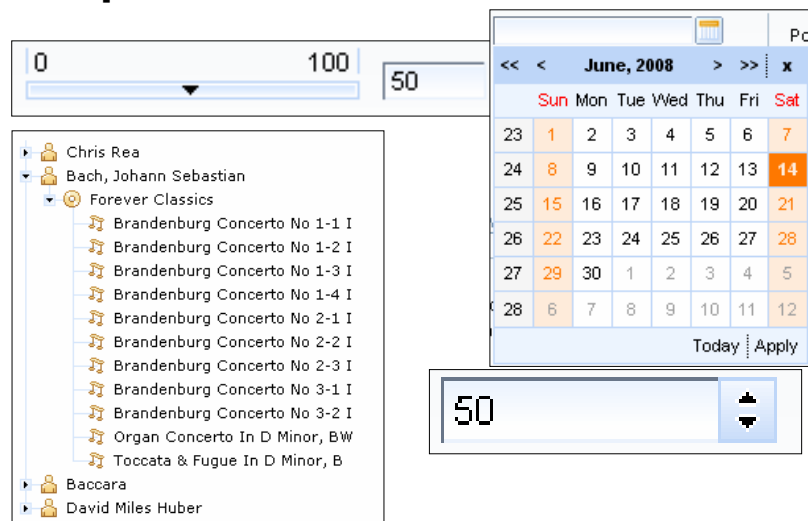
```
<h:form>
  <rich:calendar popup="true"/>

  <rich:inputNumberSpinner
    minValue="0"
    maxValue="100" step="1"/>
</h:form>
```

JSF und Ajax

- Ajax hat zunächst einmal nichts mit JSF zu tun, sondern ist eine eigenständige „Technologie“
- Es gibt jedoch Frameworks für JSF, die eine Integration von Ajax in JSF erleichtern.
- Z.B. bei ajax4jsf muss der Anwendungsentwickler keinerlei JavaScript-Code mehr schreiben
- Viele Komponentenbibliotheken im JSF-Umfeld sind bereits „ajaxifiziert“ und nutzen somit die Vorteile von PPR und asynchroner Verarbeitung.

Komponentenmarkt



The screenshot displays a JSF component market interface. At the top, there is a slider control with a range from 0 to 100 and a current value of 50. Below the slider is a tree view of components, including folders for Chris Rea, Bach, Johann Sebastian, and David Miles Huber, with sub-items like Brandenburg Concerto No 1-1 I through 3-2 I and Organ Concerto In D Minor, BWV 532. To the right of the tree view is a calendar for June 2008, showing the date 14 (Saturday) selected. Below the calendar is a numeric input field with the value 50 and a spin button.

Fazit JSF und Ajax

- JavaServer Faces bieten einen großen Komponentenmarkt
- JSF liefert viele Konzepte für die UI-Entwicklung
- JSF hat sich für die UI-Entwicklung etabliert
- JSF ist auch ein Standard
- Ajax läßt sich sehr gut in JSF integrieren.
- Es gibt Tools und IDE's, um eine JSF-Entwicklung zu unterstützen

Agenda

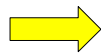
- Was ist ein Portal?
- Was leistet ein Portal / Portalsystem?
- Von Portalen zu Portlets
- Zwei Worte zu JavaServer Faces
- **Portlets und JSF**



Warum soll beides kombiniert werden?

Portlets (bzw. die Portlet-API) ist recht „basisnah“. Trotz des JSR-286 ist eine direkte Verwendung der API nicht gerade komfortabel.

Konzepte wie Pageflow, Konvertierung, Validierung, Eventhandling, UI-Komponenten fehlen komplett.



Es muss somit die Portlet-Technologie mit einem „reichhaltigen“ UI-Gerüst vermengt werden.



Die Lösung: JavaServer Faces

JSF kann all das! 😊

... wußten wir ja schon immer ...



JSF + Portlets?

Das Problem jedoch ist der „Glue-Code“.

- Beide Technologien müssen integriert werden
- Jede Technologie soll ihre „ureigene“ Funktionalität beibehalten
- Integrationsaufwände sollen möglichst gering sein

Geht das überhaupt?

→ Ja, mit dem JSR-301

JSR-301: PortletBridge Specification for JavaServer Faces

- Start im Dezember 2006
- Spec Lead: Michael Freedman (Oracle)
- Ziel ist die Bereitstellung einer standardisierten Bridge, um JSF-Anwendungen im Portlet-Context ausführen zu können.
- Ziel ist es außerdem, die Handhabung für den Entwickler so einfach wie möglich zu gestalten.
- Achtung:
 - JSR-301 = JSF 1.2 und Portlet 1.0
 - JSR-329 = JSF 1.2 und Portlet 2.0

Die Welt vor JSR-301

- Jeder Application Server hatte eine eigene Bridge-Implementierung
- Keine Bridge-Implementierung funktionierte wirklich ☹
- Komponentenbibliotheken waren nur mit viel „Fummel-Aufwand“ zum Laufen zu bekommen (wenn überhaupt)
- Es gab viele Projekte im Umfeld, um den fehlenden Standards Abhilfe zu schaffen (Filter-Portlets, Tomahawk-Bridge, ...)
- Irgendwie machte dies keinen Spaß ...

Aufgaben des JSR-301

- **Standardisierung einer Bridge-Lösung, nach der sich möglichst alle Bridge-Implementierer richten können**
- Mapping der Lifecycles
- Einführung eines Bridge Request Scope
- URL-Adressierungen
- u.v.m.

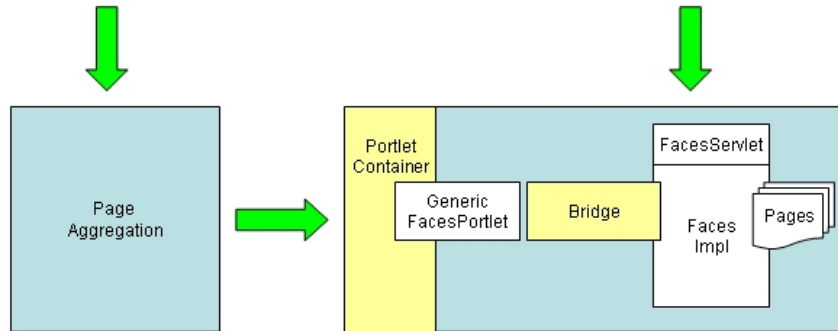




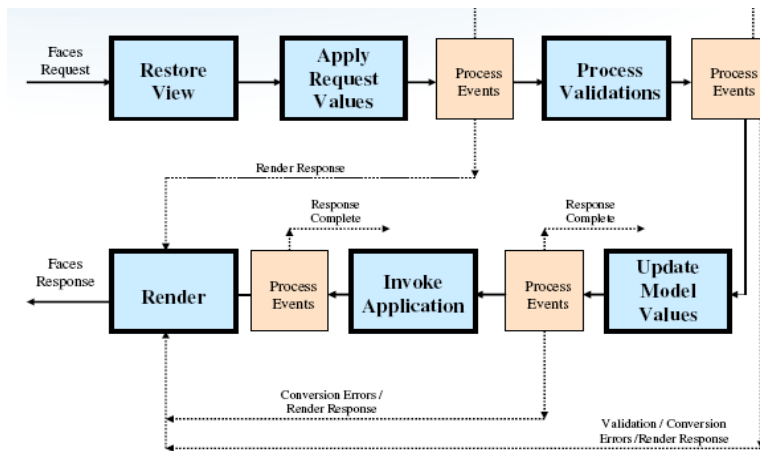
Überblick von „ganz oben“

Client Portal Request

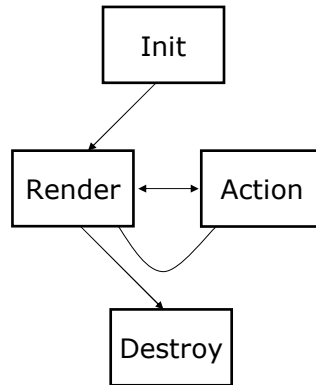
Client WebApp Request



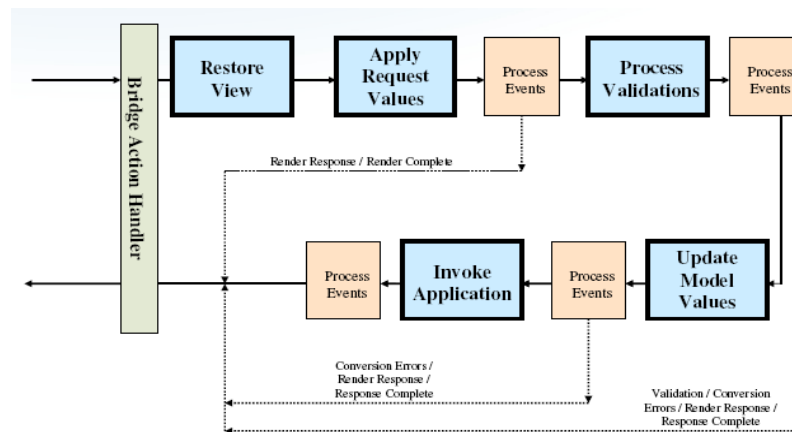
Mapping der Lifecycles: JSF



Zur Erinnerung: Portlet Lifecycle

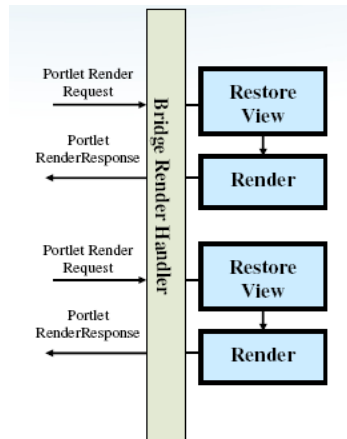


Mapping der Lifecycles: Bridge





Mapping der Lifecycles: Bridge



Bridge Request Scope

- JSF geht davon aus, dass der komplette Lebenszyklus in einem Request abgearbeitet wird.
 - Bei Portlets wird die Render- und die Action-Phase in zwei komplett unabhängigen Requests durchgeführt
- Die Bridge muss dafür sorgen, dass alle notwendigen Informationen für JSF in beiden Requests vorhanden sind.
- Dazu werden die Attribute der Action-Phase durch die Bridge zwischengespeichert und für jede Render-Phase bereitgestellt.



Einsatz der PortletBridge (1)

1. Entwicklung einer „normalen“ JSF-Anwendung
2. Entfernen „verbotener“ Tags wie <html>, <body>, <head> etc.
3. Einbinden der JSR-301 Libs
4. Einbinden der portlet.xml und Hinterlegen des GenericFacesPortlet
5. Deploy and have fun ☺



JSR-301 Implementierungen

- Referenzimplementierung innerhalb der MyFaces-Community
<http://myfaces.apache.org/portlet-bridge/index.html>
- Open Portal Implementierung
<https://jsfportletbridge.dev.java.net/>
- JBoss Implementierung
<http://www.jboss.org/portletbridge/>
- ...

Gesamt Fazit

- Ein Portal ist ein sehr mächtiges Framework zur Erstellung von Integrationslösungen auf UI-Ebene.
- Heutige Portale liefern viele Zusatzleistungen wie Personalisierung, Integration von Content, Single Sign On etc.
- JavaServer Faces ist **das** UI-Framework innerhalb von JavaEE. Mit JSF können sehr mächtige User Interfaces erstellt werden.
- Durch die Kombination von Portlets und JSF kann das **Beste aus beiden Welten** kombiniert werden.

Achtung: Werbeblock



Fragen?



Weitere Tutorials und
viel Material unter:

www.jsf-forum.de

oder auch:

www.jsf-portlets.net

Gerne auch:
andy.bosch@jsf-forum.de