

**Java Developers
&
Solaris Administrators**

DAY



ORACLE[®]



Vývoj portálových řešení v Javě

Pavel Kubal

Program

- Úvod do problematiky portálů
 - Co je to Portál
 - Jak se vyvíjejí portlety
 - Softwarová podpora vývoje
 - Výhody vývoje portálů
- Praktické zkušenosti a zajímavosti z projektů
 - Kde jsou limity technologií (ano zde existují)
 - Chyby, které byste po nás nechtěli opakovat
- Doporučení a praktické rady
 - Objevování kola je nesmírně nákladné

Zkušenosti autora

- 4 roky vývoje portálů na JSR 168 a JSR 286
- Vývoj na hlavních platformách WebLogic a WebSphere
- Portál B2B Škoda AUTO
 - Portál pro všechny dealery a importéry Škoda AUTO
- Portál B2E Škoda AUTO
 - Portál pro 20 000 zaměstnanců
- Portál ÚVN INIS
 - Portál pro zdravotnický personál nemocnice

Úvod do Java portálů



Co je to Portlet?

- Javová webová komponenta, jejíž běhové prostředí je portlet kontejner
- Stejně jako Servlet generuje HTML kód (fragment)
 - Portlet je větší abstrakce od HTTP protokolu
 - Výstupy portletů se agregují a vytváří portálovou stránku
 - Portlet má celou řadu omezení (nesmí generovat <head>...)
- Portlet standardy
 - JSR 168 Portlet specifikace 1.0
 - JSR 286 Portlet specifikace 2.0
- Celá řada dalších specifikací WSRP, Portlet Bridge atd.

Jak vypadá portlet?

- Podobně jako servlet

```
import javax.portlet.*;
import java.io.*;

public class HelloWorldPortlet extends GenericPortlet {
    public void doView(RenderRequest request, RenderResponse response) throws
        PortletException, IOException {

        response.setContentType(request.getResponseContentType());
        PrintWriter writer = response.getWriter();
        writer.write("Hello from portlet.");
    }
}
```

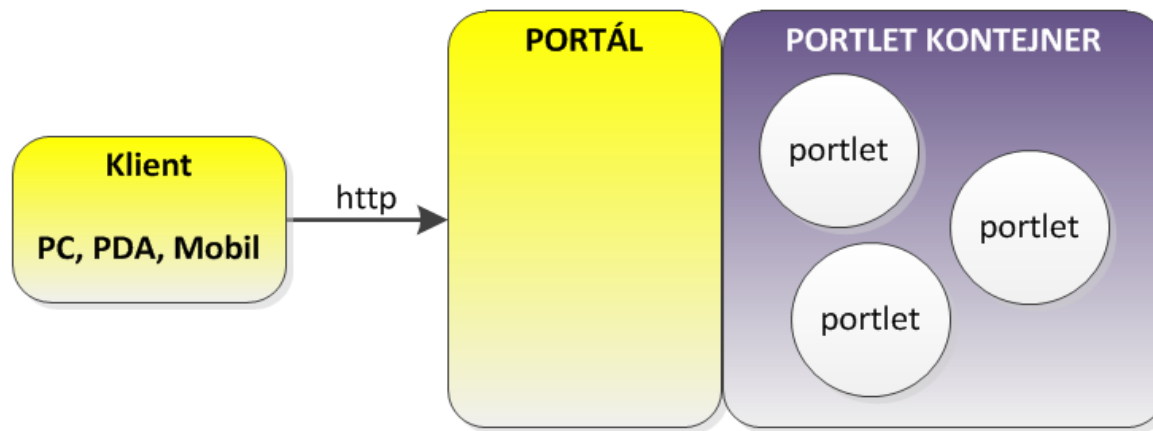
- Má svůj zvláštní deployment deskriptor portlet.xml
 - Každá webová aplikace má jeden, který může deklarovat více portletů

JSR 168 a JSR 286

- Kontrakt mezi komponentou (portletem) a kontejnerem (portálem)
- Řeší celou řadu problémů
 - Zpracování požadavku, stavové informace (session), životní cyklus, deployment, preference, paralelní rendering, meziportletovou komunikaci, módy portletu...
- A celou řadu problémů vůbec neřeší
 - Upload, download, AJAX, layout stránky, sdílení informací mezi portlety v jiných webových aplikacích, některé aspekty deploymentu

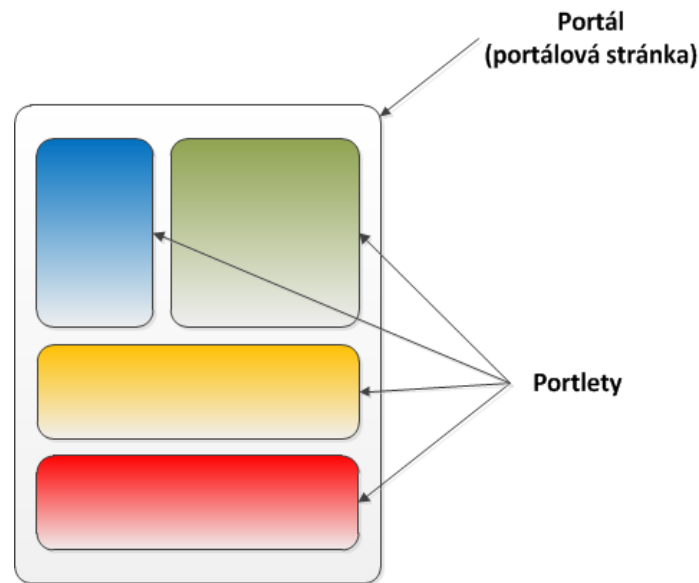
Co je to portlet kontejner?

- Portlet kontejner je běhové prostředí pro portlety
 - Běhové prostředí portlet kontejneru je servlet kontejner
- Například open source projekt Pluto

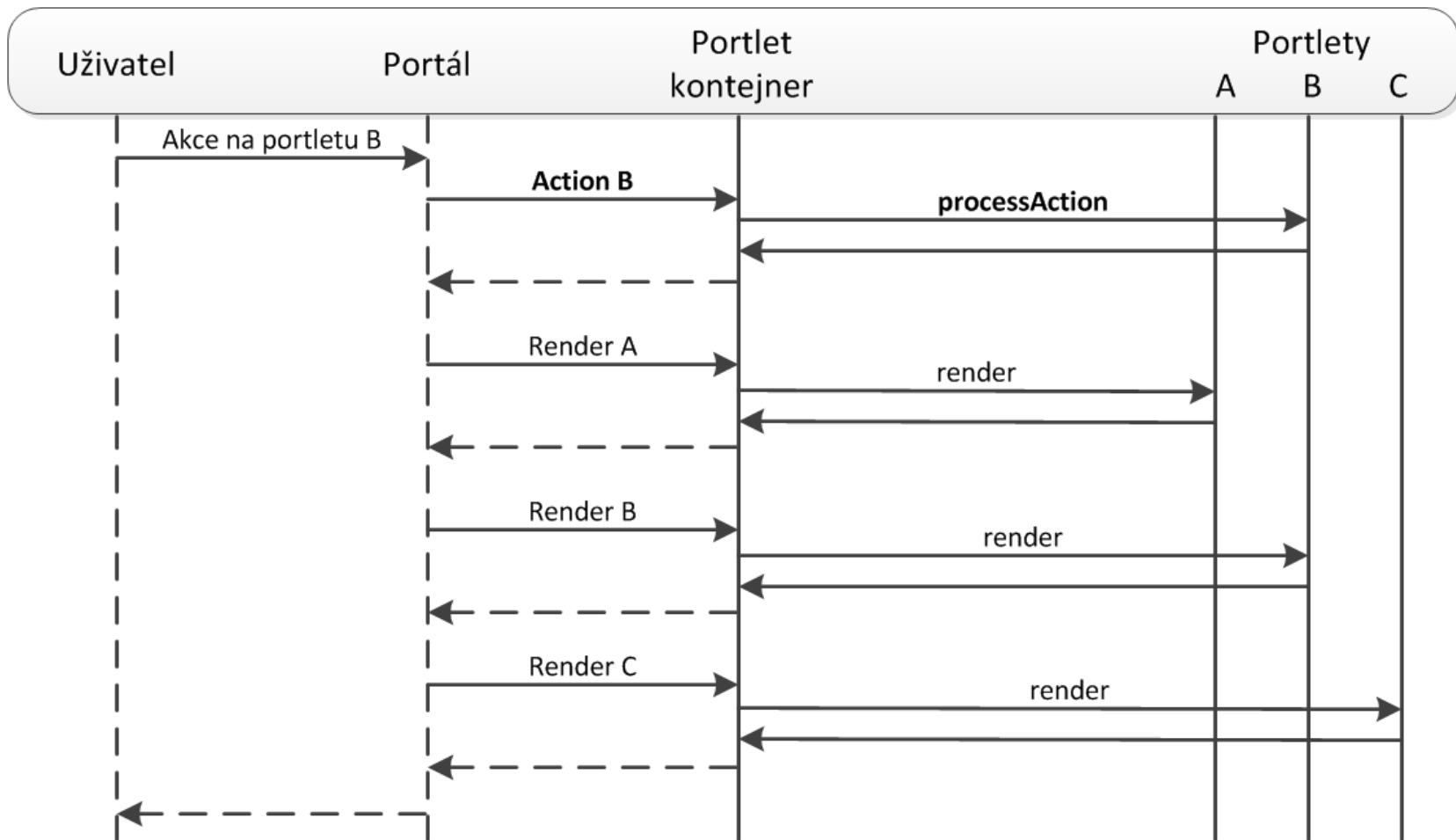


Co je to portál?

- Portál je webová stránka agregující obsah portletů
- Celá řada open source i komerčních
 - Pluto, WebSphere Portal, Jetspeed, JBoss Portal...



Interakce portletů a portálu



* převzato z prezentace Sang Shina 12/09/03

Proč portlety?

- Portlety představují další abstrakci od protokolu HTTP
- Žádná jiná technologie neagreguje výstup několika aplikací do jedné stránky
- Portlet můžeme nasadit do jakéhokoliv portálu a umístit kamkoliv na stránku
- Jednotlivé portlety mohou být založeny na rozdílných technologiích i vývojářských konvencích
- Portlety jsou na sobě navzájem nezávislé
- Portlety mají na rozdíl od servletů rozdělen požadavek na akci a rendering

Další výhody portálů

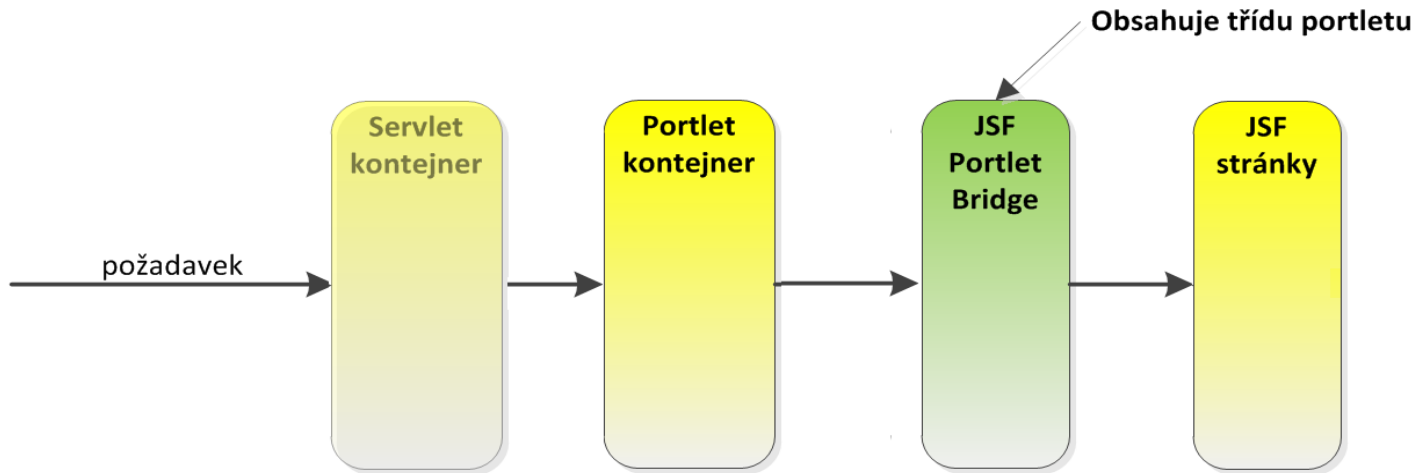
- Portál jako produkt přináší celou řadu služeb out-of-the-box:
 - Personalizaci (možnost upravit si portál podle potřeby)
 - Autentizaci a autorizaci
 - Agregaci obsahu z různých zdrojů
 - Vyhledávání
- Na svém portálu můžeme využívat portlety od jiných dodavatelů (např. počasí)
- Stávající aplikace můžeme převést na portlety
- Portlety mají stavy (min, max) a módy (VIEW, CONFIG)

Vývoj portletů



Jak se píší portlety

- Portlety můžeme psát ve svém oblíbeném webovém frameworku, pokud pro něj existuje bridge (JSF, Struts)
- JSF Portlet Bridge – JSR 301, JSR 329



Portlet Bridge

- Portlet Bridge je abstrakce od technologie portletů
- Díky PB vypadá portletová aplikace jako běžná webová aplikace
- Portlet Bridge odstíní programátora od životního cyklu portletu
 - Výhoda: neřeším portlety a píšu aplikaci
 - Nevýhoda: nemám přístup k potomkovi třídy GenericPortlet
- Celá řada různých implementací
 - Apache MyFaces Portlet Bridge, JBoss Portlet Bridge, Open Portlet Container Bridge...

Softwarová podpora vývoje a provozu portálů

- Jaké softwarové komponenty potřebujeme pro vývoj
 - Framework pro tvorbu stránek
 - JSF, Struts, Spring, Velocity, Groovy, Perl a PHP!
 - Portlet Bridge
 - Apache, JBoss, Open Portal...
 - Portlet kontejner (portál)
 - Open Portal, WebSphere Portal, JBoss Portal, Pluto
 - Aplikační server
 - Tomcat, Jetty, WebSphere, JBoss, WebLogic

Zkušenosti s vývojem portálů

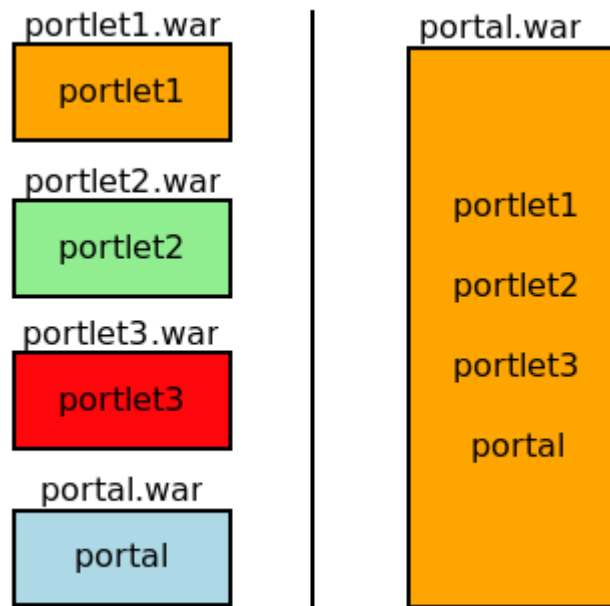


Na co si dát pozor

- Build a deployment
- Sdílení dat
- Ajax
- Upload dat
- Konfigurace portletu
- Problematické požadavky na portál
- Chyby, které byste po nás nechtěli opakovat

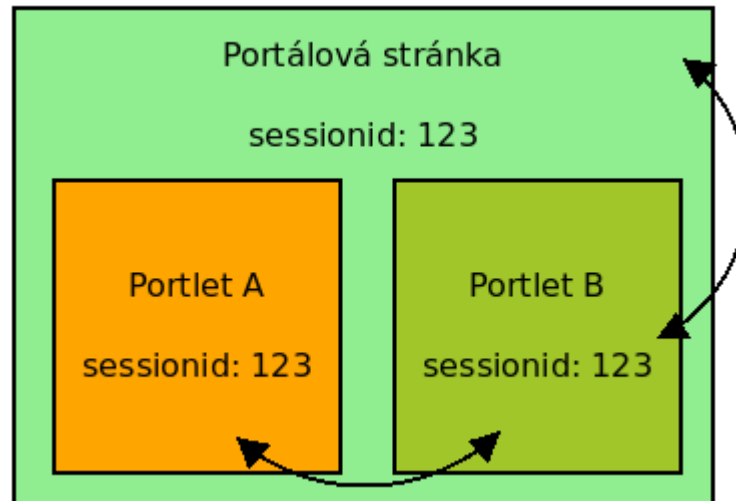
Build a deployment

- Dvě hlavní možnosti či hybrid
- Každý přístup má své výhody a nevýhody



Sdílení dat

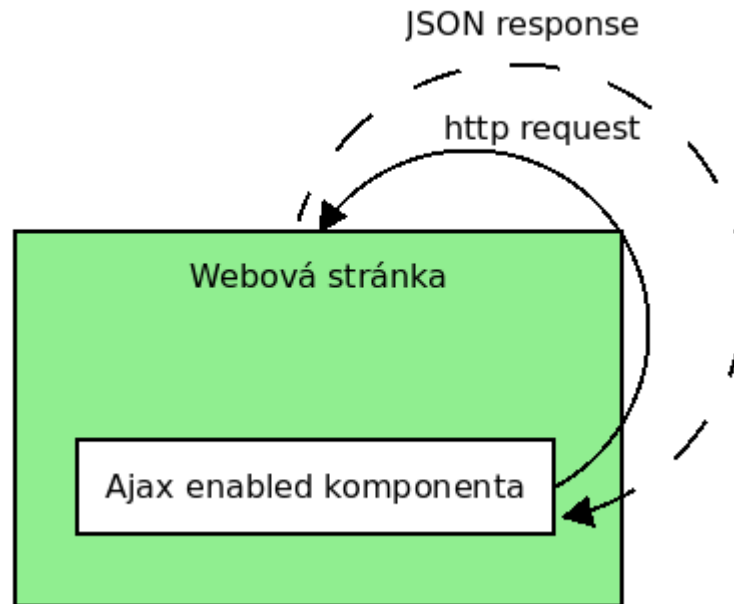
- Portlety sdílí session spolu se servlety a JSP stránkami
- Ale pouze pokud jsou ve stejné aplikaci



- Další možnosti sdílení dat jsou cache a application scope

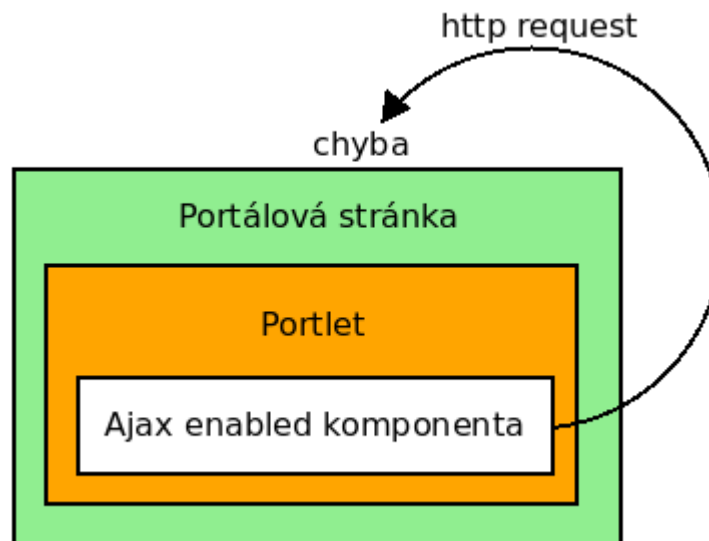
Ajax

- Ajax u standardních webových komponent bez portálu



Ajax v portálu

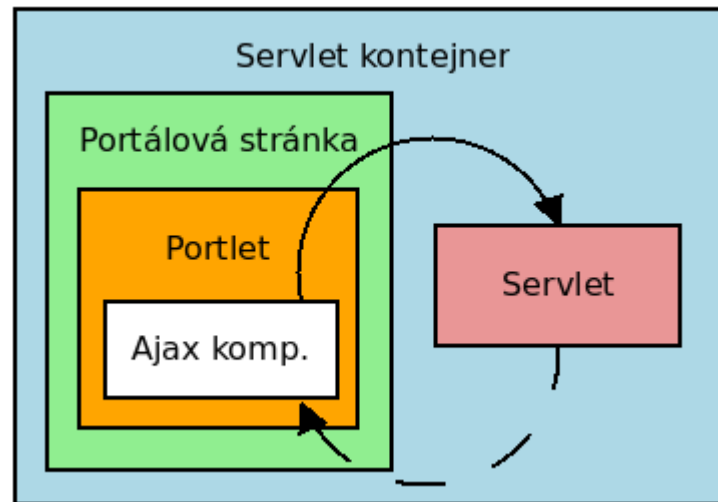
- V portálu standardní Ajaxové komponenty nefungují



- Portálová stránka neví jak přesměrovat požadavek

Ajax v portálu - řešení

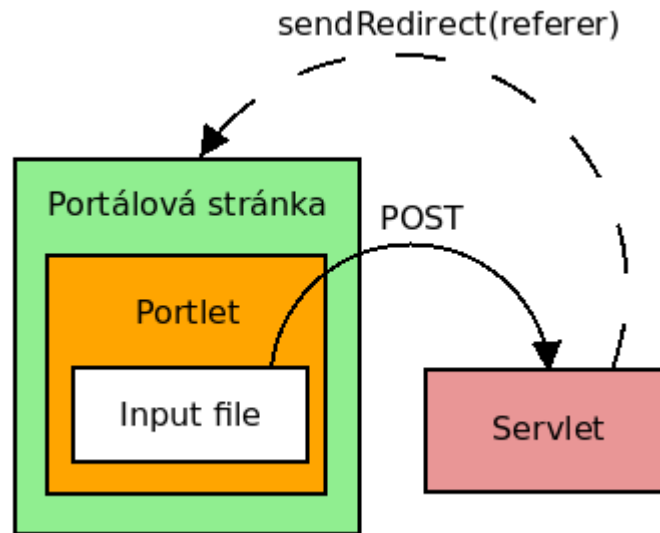
- Vytvořit servlet, který bude vracet JSON response



- Jedná se o reimplementaci zabudovaného Ajaxu pro portálové řešení

Upload dat

- Obvykle kombinace symptomů ajaxového problému a problému sdílení dat + návrat na stránku portálu



- Řešení je opět servlet + redirect na referera

Konfigurace portletu

- Pro konfiguraci portletu lze využít
 - web.xml (context-param)
 - Dostupný ze PortletContext
 - portlet.xml (init-param)
 - Dostupný z PortletConfig (nedostupný v PortletBridge)
 - portlet.xml (preferences)
 - Dostupný z PortletRequest (ne v MyFacesBridge)

Problematické požadavky portálu

- Portál nemá smysl pro každý typ aplikací
- Problematické požadavky na portál
 - Rozlišení 1024x768 (cca jako rozlišení iPhoneu)
 - Layout – ne každý si dokáže představit více než jeden portlet na stránce
 - Restart navigačního stavu portletu
 - Problematické kombinace softwarových komponent
 - MyFaces, MyFacesBridge, OPC, WLS → průšvih

Chyby, které byste po nás nechtěli opakovat

- Restart navigačního stavu portletu
 - Standardně se udržuje navigační stav portletu i při klikání do menu
 - Je nutno ověřit zda vybrané portálové řešení umí restart navigačního stavu
 - WebSphere Portal restart umí, ale ne pro JSF portlety
 - MyFaces Bridge restart neumí
 - OPC Bridge restart umí
 - ...
- Velmi problematická je situace, kdy zákazníkovi vysvětlujete, že restart stavu portletu nelze provést

Lessons learned

- Nekombinujte příliš mnoho softwarových balíčků
 - Preferujte JSF framework, Portlet Bridge, Portál a servlet kontejner od jednoho výrobce
- Kladte velký důraz na fázi proof of concept
- Drahé komerční portály obvykle nenabízí oproti open source řešením nic navíc (kromě uzavřeného kódu)
 - Ani podpora nemusí pomoci (viz WPS)
- Už od začátku se soustředte na možnosti sdílení funkcí mezi portálem a portlety
 - CSS styly, JavaScript, autorizace

Lessons learned

- Je lepší když si portálové stránky vytvoříte jako JSP než abyste je klikali v administrační konzoli
 - Výrazně větší flexibilita
- Nedovolte uživatelům pohybovat s portlety na stránce!
 - Podpora nebude řešit nic jiného
- Pozor na tuny JavaScriptu a CSS, které jsou v některých portálech by default

Co v prezentaci nebylo probráno

- Vývoj servletů a JSP stránek
 - SL-314-EE6
- Java Server Faces 2.0
 - DTJ-3108
- Bezpečnost v prostředí webových aplikací v Javě
 - DTJ-3109

Dotazy



pro prezentaci byly použity inspirace z volně dostupných materiálů

Joke at the end

Adapted from the beloved Java API

“Calling `getCause()` gets the cause that caused the throwable to be thrown and when you've got the cause that threw the throwable, you'll know what caused the exception .”