



Vývoj portletů

František Hartman



- Agenda
 - Představení portálu
 - Portlety
 - Liferay
 - Úvod do vývoje portletů
 - Portletová specifikace
 - J SR-168 / J SR-286
 - API
 - Portlet gag library
 - Praktické cvičení
 - J jednoduchý portlet

- Integrovaná platforma
 - Integrace existujících aplikací

- Vývojová platforma
 - Standardizované prostředí
 - Správa uživatelů, rolí, oprávnění, autorizace, autorizace, single sign-on
 - Správa vzhledu, rozložení a hierarchie stránek
 - Cache
 - Personalizace obsahu
 - Ukládání nastavení

- Kdy portál nepoužít?

- Proč – situace před 10 lety
 - Různí dodavatelé mají různá API
 - Není možná spolupráce mezi portlety a různými portálovými servery
 - Zákazníci uzamčeni do konkrétního portálového řešení
 - Neexistuje jednoduchá cesta jak integrovat aplikaci do portálu

- Dnes
 - Standardy J SR-168 a J SR-286

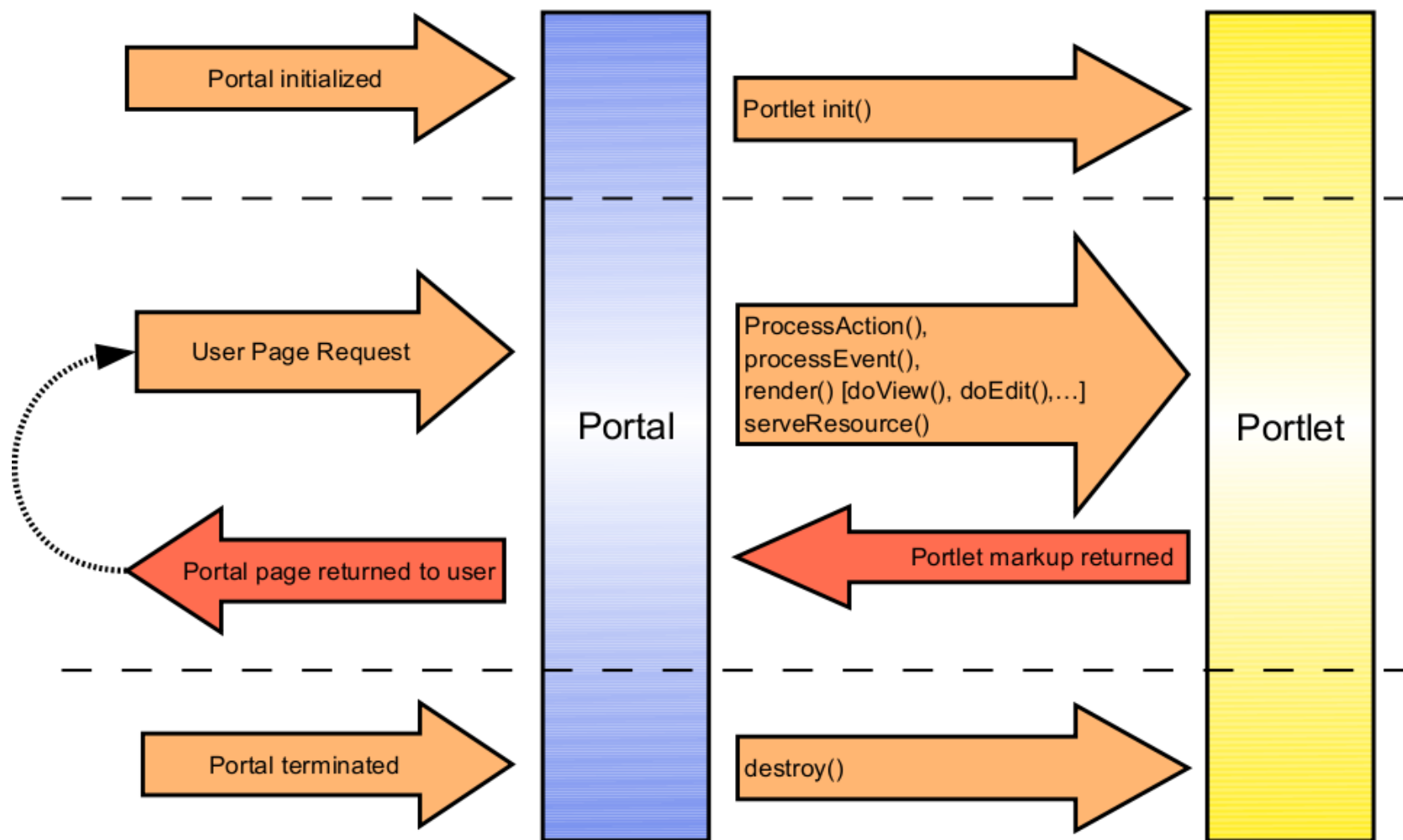
- Portletový standard pro lokální java portlety
 - Dokončeno říjen 2003
 - <http://jcp.org/en/jsr/detail?id=168>
- Cíle
 - J ednoduchý programovací model
 - Přenositelnost
 - Využití a kooperace s technologiemi J 2EE
 - Interoperabilita s dalšími J ava technologiemi

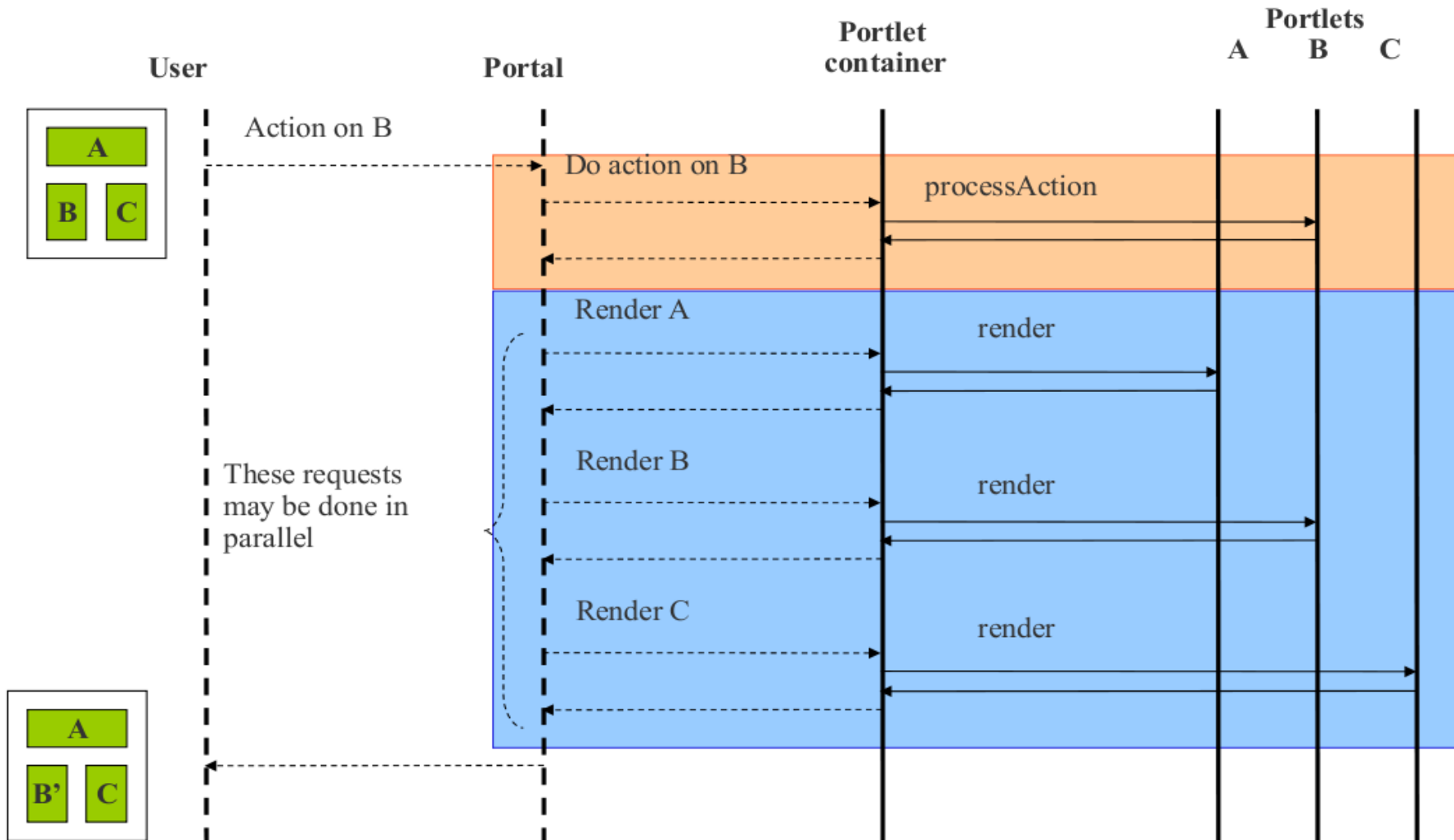
- Druhá verze standardu
 - Dokončeno říjen 2008
 - <http://jcp.org/en/jsr/detail?id=286>

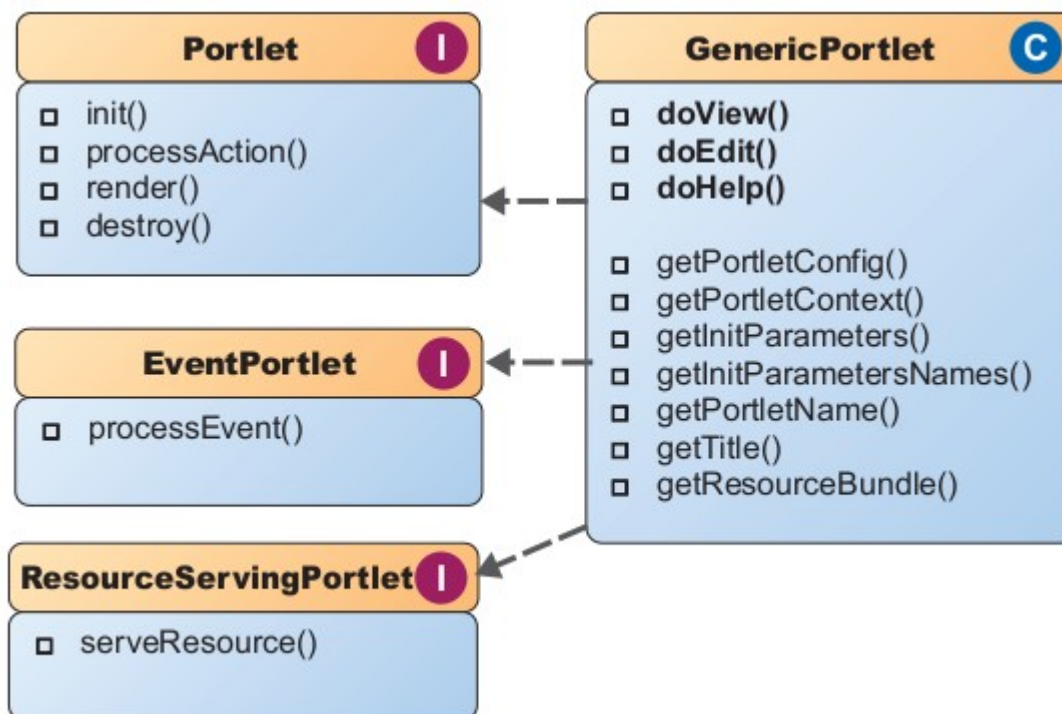
- Rozšíření JSR-168
 - Zpětná kompatibilita
 - Meziportletová komunikace
 - Veřejné parametry (public render parameters)
 - Události (events)
 - Poskytování obsahu
 - Obrázky, datové soubory
 - AJAX
 - Filtry
 - Zjednodušení integrace webových rámců

- portlet.xml
 - Obsahuje konfigurační informace přímo pro portál
 - Podporované módy a stavy oken
 - Podporované jazyky (lokalizace)
 - Popis a označení portletu
 - Konfigurace preferencí
 - Veřejné parametry
 - Události
 - Filtry

- A další







- GenericPortlet
- Standardní implementace rozhraní Portlet
- Abstraktní třída, jenž je nejčastěji rozšiřována
- Vlastní portlety by měly přepisovat
- doView() pro zpracování požadavků ve VIEW módu
- processAction() pro zpracování požadavků a změnu stavu portletu
- doEdit()
- doHelp()
- init() a destroy() pro práci se zdroji
- Běžně se nepřepisují metody
- render() a
- doDispatch(), která volá doView(), doEdit() nebo doHelp() na základě požadovaného portletového módu

- Stav okna portletu (window state)
- Stav okna definuje, jak bude portlet zobrazen na stránce
- Normal – jako součást stránky
- Maximized – daný portlet je na stránce sám nebo zabírá většinu plochy
- Minimized – většinou pouze záhlaví portletu
- Existují další stavy specifické pro konkrétní portály
-
- Portletové módy (portlet mode)
- Portletové módy umožňují zobrazení různých pohledů
- View – základní mód
- Edit – většinou pro úpravu uživatelských nastavení
- Help – zobrazení nápovědy k portletu
- Existují další módy specifické pro konkrétní portály
- Configure – administrace portletu

- PortletConfig
- Poskytuje portletu jeho nastavení z portlet.xml
-
- PortletContext
- Definuje pohled portletu na portletový kontejner
- Inicializační parametry jsou pouze ke čtení a nastavují se v deskriptoru web.xml
- Důležité metody
- `getRequestDispatcher(java.lang.String path)`
- `getInitParameter(java.lang.String name)`
- `getInitParameterNames()`

- PortletRequest
- Vytvářeny a spravovány portletovým kontejnerem
- Předávány jako argument portletovým metodám
- Jsou využívány pro komunikaci mezi portletovým kontejnerem a portletem a zároveň k předávání dat v rámci zpracování požadavku
- Data
- Parametry
- Atributy
- Relace (sezení, session)
- Preference
- Informace o uživateli
- `getRemoteUser()` - login uživatele (nebo id)
- Atribut `USER_INFO`

- PortletResponse
- Rozhraní pro odesílání odpovědí na požadavek
- `encodeUrl()` - Vrací zakódované URL požadovaného zdroje (např. servlet, JSP, obrázky nebo jiný statický obsah)
- RenderResponse
- `create*URL()` - vytváření URL
- ActionResponse
- `sendRedirect()`
- `setPortletMode()`
- `setWindowState()`
- `setRenderParameter()`
- `setRenderParameters()`
- Parametry viditelné v `processAction()` se automaticky nepropagují do `render()`
-

- PortletPreferences
- Perzistentní úložiště dat spravované kontejnerem
- Výchozí hodnoty jsou definované v portlet.xml
- Pro každé portletové okno se nastavují samostatně
- Pro uložení dat je zapotřebí volat metodu store()
- Nelze volat během render fáze
- PortletSession
- Úložiště pro informace déle než na dobu jednoho požadavku
- Session se vytváří pro uživatele pro každou portletovou aplikaci
- Pozor na velikost objektů, které do session ukládáte – velké objekty mohou mít dopad na výkonnost
- `getAttribute(String name)`
- `getAttribute(String name, int scope)`
- `setAttribute(String name, Object value)`
- `setAttribute(String name, Object value, int scope)`
- `APPLICATION_SCOPE` a `PORTLET_SCOPE`

Umístění	Typ	Velikost	R/W	Doba života
Init parameters(PortletConfig)	String	KB	RO	Jako portlet
Init parameters(PortletContext)	String	KB	RO	Jako portlet
Attributtes	Object	MB	RW	Request
Parameters	String	KB	RO	Request
Session Attributes	Object	KB	RW	Session
Preferences	String	KB	RW/RO	Persistent

- Značky pro
- Generování odkazů
- Přístup k objektům v request a response
- Definované značky
- defineObjects
- renderURL
- actionURL
- resourceURL
- namespace
- param

- Vytváření odkazů
-
- `<portlet:renderURL var="detailUrl">`
- `<portlet:param name="view" value="detail"/>`
- `<portlet:param name="id" value="42"/>`
- `</portlet:renderURL>`
- `Show detail`
-
- Značka `<portlet:namespace>`
- Vrací jednoznačné označení portletu
- Používá se pro odlišení entit na stránce v případě, že je na ní více stejných portletů
- Využití
- Javascript
- Element ID
-

- Příště
- Meziportletová komunikace
- Obsluha zdrojů
- Filtry