

## Design of Information Systems Using the Portlet Technology

Victor BĂRCAN

Universitatea Babeş-Bolyai din Cluj-Napoca  
Facultatea de Ştiinţe Economice

*Portlets are Java-based Web components, managed by a portlet container, that process requests and generate dynamic content. Portals use portlets as pluggable user interface components that provide a presentation layer to information systems. The next step, after servlets in Web application programming, portlets enable modular and user-centric Web applications. In this paper are described the principal characteristics and the functionalities of the portlets. There are presents some notions of administration of a J2EE server which supports this technology (IBM WebSphere Application Server).*

**Keywords:** portlets, java components, application server

Scopul acestui articol este prezentarea generală a tehnologiei portletelor, precum și a particularităților legate de implementarea acestora de către produsul IBM WebSphere Portal Server.

### 1. Conceptul de portlete

#### ➤ Ce este o portletă?

O portletă este de fapt o servletă care însă este gestionată de către un container de portlete. Sunt deci necesare câteva lămuriri asupra ceea ce presupune un container de servlete.

#### ➤ Containerul de servlete

În momentul creării unei portlete aceasta este definită mai întâi în descriptorul de desfășurare ca și servletă.

Denumirea de servletă vine de la cuvântul server, deoarece servletele sunt considerate ca și extensii ale serverului, de unde denumirea care a fost adoptată.

În primul rând trebuie observat modul în care containerul de servlete gestionează o servletă.

Servletele au apărut ca și tehnologie ca alternativă java optimizată la programele CGI. În cadrul unei aplicații bazate pe acestea din urmă un apel HTTP presupunea crearea unei instanțe a programului CGI apelat, altfel spus, dacă 100 de utilizatori cereau în același timp un program CGI, serverul crea în memorie 100 de instanțe ale aceluși program pentru a face față cererii.

Nu același lucru se întâmplă în cazul folosirii servletelor. Servletele sunt gestionate de că-

tre containerul de servlete într-un mod diferit:

La demararea aplicației containerul de servlete parcurge fișierul web.xml, și pentru fiecare servletă definită în cadrul acestui fișier care are opțiunea de încărcare la lansarea activată, creează o instanță. Opțiunea de încărcare la pornire este specificată cu ajutorul unei balize xml:

```
<load-on-startup>1</load-on-startup>
```

Numărul este folosit pentru specificarea ordinii de încărcare a servletelor.

Dacă servleta nu este definită în cadrul descriptorului de desfășurare, sau dacă este definită dar nu este activată opțiunea de încărcare la pornire, atunci instanța este creată în momentul primului apel.

Indiferent de felul în care a fost creată instanța, toate cererile pentru servleta respectivă vor fi servite de către această instanță.

Acest lucru este posibil deoarece servletele sunt componente de tip multi-thread. După crearea instanței pentru fiecare cerere este creat un nou thread care se execută pe instanța respectivă.

Deoarece ceea ce consumă timp și resurse sistem, este tocmai crearea de instanțe, folosirea alternativei client-server pe baza de tehnologie servlete conferă câștiguri foarte importante de performanță.

Există totuși două excepții bine definite în care containerul de servlete poate crea mai multe instanțe ale aceleași clase:

#### a) Interfața SingleThreadModel

Aceasta este o interfață de tip „tagged interface”, sau interfață etichetă, adică este o interfață care nu conține nici o metodă, și nici o constantă, ceea ce înseamnă că orice clasă care implementează o astfel de interfață nu trebuie să implementeze nici o metodă. Consecința implementării de către o servletă a acestei interfețe, este că, containerul de servlete garantează că un apel client va fi deservit de o instanță dedicată. Aceasta nu înseamnă că vor exista atâtea instanțe câte apeluri sunt, deoarece, containerul de servlete poate să pună anumite apeluri în așteptare.

#### b) Definierea a două sau mai multe servlete având aceeași clasă.

În cadrul descriptorului de desfășurare web.xml se pot defini una sau mai multe servlete care se bazează pe aceeași clasă. Care sunt avantajele unei astfel de practici?

O primă consecință este că pot exista în Java Virtual Machine, la un moment dat, tot atâtea instanțe ale clasei respective câte servlete sunt definite pe clasa respectivă.

Servletele respective pot avea parametri de inițializare diferiți și pot fi mapate pe URL-uri diferite.

Această practică este avantajoasă în situația în care în funcție de parametri diferiți se execută acțiuni diferite, servlete care de altfel sunt apelate prin URL-uri diferite.

Pe de altă parte, codul din cadrul clasei respective, este reutilizat cu succes în toate servletele definite pe baza acestei clase în cadrul descriptorului de desfășurare web.xml.

Această tehnică va fi utilizată în cadrul proiectului de integrare ce va fi prezentat în continuare.

#### ➤ Containerul de portlete

Portalul este considerat ca fiind containerul de portlete. Parametrizarea comportamentului unei portlete se face cu ajutorul descriptorului de portlete care este fișierul *portlet.xml* urmând același mod de parametrare ca și în cazul fișierului *web.xml*.

Practic, o portletă nu este altceva decât o servletă care este gestionată de către containerul de portlete. Însăși ierarhia de clase folosită pentru a crea o portletă derivă din clasa *HttpServlet*.

Portleta face referire la acea servletă în cadrul fișierului de parametrare a portletelor: *portlet.xml*. Pentru a defini o portletă trebuie definită mai întâi servleta corespunzătoare în cadrul fișierului *web.xml*, iar apoi portleta și aplicația concretă a portletei în cadrul fișierului *portlet.xml*.<sup>1</sup>

#### ➤ Modalități de funcționare a portletelor.

Există mai multe moduri de funcționare ale unei portlete, după cum urmează:

1. *Afișare*; acesta este modul de bază de funcționare al unei portlete.
2. *Ajutor* folosit pentru afișarea unei pagini conținând indicații de folosire ale portletei.
3. *Editare* - folosit de exemplu de către utilizator pentru specificarea unor parametri de utilizare a portletei, specifici acestui utilizator.
4. *Configurare* - folosit de către administrator, sau de către utilizatorii acreditați pentru precizarea unor parametri de utilizare a portletei, specifici fiecărui utilizator.

Accesul la aceste moduri diferite de funcționare ale unei portlete este condiționat de atribuirea drepturilor corespunzătoare utilizatorului respectiv.

Toate aceste moduri sunt accesibile activând o pictogramă corespunzătoare<sup>2</sup>.





#### ➤ Metode

O portletă moștenește clasa *PortletAdapter* în cadrul căreia sunt definite mai multe metode. Metodele cele mai interesante pentru programator sunt :

• *InitConcrete(...)* lansată doar o singură dată în momentul creării instanței portletei respective, și folosită pentru executarea procedurilor de inițializare ale portletei.

<sup>1</sup> Detalii despre programarea și configurarea portletelor pot fi găsite în lucrarea: **IBM WebSphere Portal V4 Developer's Handbook**, disponibilă în mod gratuit la adresa: <http://www.redbooks.ibm.com/abstracts/sg246897.html?Open>

<sup>2</sup> Iconitele folosite pentru diferitele moduri sunt următoarele :

-  Afișare
-  Ajutor
-  Editare
-  Configurare

- **doView(...)** lansată ori de câte ori portleta este apelată în mod vizualizare și este folosită pentru afișarea conținutului.
- **doEdit(...)** lansată atunci când se folosește modul editare, și este folosită de exemplu pentru specificarea unor parametri.
- **doConfigure(...)** lansată atunci când se folosește modul configurare, și este folosită de exemplu pentru specificarea unor parametri.
- **doHelp(...)** lansată atunci când se folosește modul ajutor, și este folosită pentru afișarea unor indicații.

Containerul de portlete este cel care garantează apelul metodei java corespunzătoare modului de funcționare al portletei folosite.

## 2. Administrarea portalului

Produsul permite administrarea portalului de către persoane care nu au un nivel foarte avansat în informatică. Astfel se pot instala portlete, crea spații, pagini, acorda drepturi într-un mod foarte simplu și eficient.

Elemente organizatorice din punct de vedere funcțional sunt spațiile, paginile și portletele.

### ➤ Drepturi de acces

Drepturile de acces asupra acestor elemente pot fi acordate de către administrator pe patru nivele :

1. *Vizualizare*: presupune că utilizatorul căruia i s-a acordat acest drept poate doar vizualiza resursa în cauză
2. *Editare*: În cazul spațiilor și al paginilor, acest drept presupune modificarea conținutului și al modului de dispunere al elementelor componente. În cazul portletelor, acest drept presupune folosirea modului editare pentru a specifica de exemplu parametrii de funcționare ai portletei.
3. *Configurare*. În cazul portletelor, presupune folosirea modului de lucru **configure**, ce poate fi foarte util în cazul specificării unor parametri de funcționare ai portletei, necesari tuturor utilizatorilor.
4. *Delegarea* presupune acordarea acestor tipuri de drepturi asupra resurselor în cauză altor utilizatori.

### ➤ Elemente organizatorice utilizate la elaborarea proiectului de integrare

**Spațiile** sunt elementele organizatorice de nivel cel mai înalt în cadrul portalului. De

exemplu, se poate crea câte un spațiu pe departament și apoi se pot acorda dreptul de acces doar grupului de persoane din acel departament.

În momentul conectării pe portal, primul lucru pe care-l va vedea utilizatorul sunt spațiile asupra cărora are drepturi de acces.

**Paginile** sunt elemente componente ale spațiilor, drepturile de acces ale utilizatorilor pot fi acordate la nivel de pagină.

Pe o pagină pot fi dispuse una sau mai multe portlete.

**Portletele** sunt elementele funcționale finale, care sunt răspunzătoare pentru afișarea conținutului propriu-zis. În funcție de drepturile acordate, utilizatorul autentificat poate activa diferitele moduri de funcționare ale portletei.

## 3. Single Sign On

### ➤ Notiune

Conceptul de bază al produsului de integrare WebSphere Portal Server constă în accesul printr-un punct unic de intrare la totalitatea aplicațiilor întreprinderii.

În acest context capătă importanță noțiunea de **Single sign On**, care presupune o singură autentificare pentru a accesa totalitatea aplicațiilor. Există mai multe moduri de realizare ale acestei funcționalități, după cum urmează :

### ➤ Single Sign On Clasic

Acesta presupune faptul că toate aplicațiile accesate prin intermediul portalului și-au externalizat și centralizat autentificarea pe un singur server LDAP.

În momentul instalării produsului se alege opțiunea LDAP, și se face instalarea folosind anuarul LDAP în cauză.

Atunci când un utilizator se autentifică pe portal se autentifică pe serverul LDAP al întreprinderii. În momentul în care acest utilizator încearcă să acceseze o aplicație prin intermediul unei portlete, datele de autentificare denumite *credentials*, sunt transmise aplicației în cauză de către Portal, nemaifiind necesară autentificarea.

### ➤ Single Sign Extins

Atunci când aplicațiile de integrat folosesc mecanisme de autentificare proprii, transmiterea datelor de autentificare a utilizatorilor

pe portal a aplicației în cauză nu mai are nici un sens.

Pentru a păstra funcționalitate de Single Sign On, trebuie furnizate și păstrate datele de autentificare pe aplicația respectivă pentru fiecare utilizator. În plus aceste date trebuie accesate în condiții de maximă securitate.

În acest scenariu, utilizatorul se autentifică pe portal, iar în momentul în care încearcă să acceseze aplicația respectivă WPS, regăsește datele de autentificare pentru acces utilizator pentru această aplicație, realizând astfel autentificarea.

WPS oferă un răspuns tehnologic la această problemă prin mecanismul de credențial Vault.

#### ➤ **Notiunea de Vault**

Mecanismul de **credențial vault** se bazează pe noțiunea de **Slot** și permite stocarea datelor de autentificare ale unui utilizator pentru o aplicație într-un mod transparent din punct de vedere SQL/JDBC pentru programator.

Acesta apelează doar anumite metode specifice WPS, care se vor ocupa de persistarea și regăsirea slotului pentru acest utilizator și pentru această aplicație. Un exemplu concret al unor astfel de apeluri este prezentat în capitolele următoare ale lucrării.

#### ➤ **Tipuri de Sloturi**

Există mai multe tipuri de slot suportate de către WPS, după cum urmează :

1. **Private Slot** este folosit de către un utilizator în cadrul unei portlete. Slotul de acest tip nu este folosit de către alte portlete pentru acest utilizator sau pentru alții. Un astfel de slot va fi folosit în cadrul aplicației de integrare pentru stocarea datelor specifice utilizatorului.

2. **Shared Slot** este folosit de către un utilizator pentru mai multe portlete. Datele persistate în cadrul unui shared Slot pot fi accesate de către toate portletele aceluși utilizator.

3. **Administrative Slot** este folosit pentru a stoca date de configurare de către administrator. Aceste date vor fi pe urmă interpretate de către portletă pentru a accesa aplicația în cauză. Acest tip de slot va fi folosit în cadrul aplicației de integrare pentru stocarea datelor comune tuturor utilizatorilor.

4. **System Slot** este folosit pentru a persista informații accesibile tuturor utilizatorilor în cadrul tuturor portletelor.

#### **Bibliografie**

1. Juan Rodriguez, Jerome Curlier, Werner Frei, Denise Hatzidakis, Juergen Moetzel, Nancy Ting, Shawn VanRaay, **IBM WebSphere Portal V4 Developer's Handbook**, IBM Form Number: SG24-6897-00, 2003, ISBN: 0738427837
2. Michele Galic, Johan Edling, Chris Hisler, Paul Holm, Wesley King, Santosh K M, Remy Tiffon, **Access Integration Pattern using IBM WebSphere Portal Server**, IBM Form Number: SG24-6267-00, 2001, ISBN: 0738422657
3. Michele Galic, Lori Cramer, Martin Wolfe, Roshan Rao, **A Portal Composite Pattern Using WebSphere Portal V4.1**, IBM Form Number: SG24-6869-00, 2003, ISBN: 0738427446
4. Peter Kovari, Alex Barbosa Coqueiro, José Guilherme S.T. de Souza, Sergio Del Valle, Michele Galic, Luís Fernando Liguori, **Patterns: Pervasive Portals Patterns for e-business Series-** IBM Form Number: SG24-6876-00, 2003, ISBN: 0738427772
5. Rufus Credle, Denise Hendriks Hatzidakis, Sunil Hirannah, Gord Niguma, Dwight Norwood, Roshan Rao, Bernhard Stimpfle, **IBM WebSphere Portal V4.1 Handbook Volume 1**, IBM Form Number: SG24-6883-00, 2003, ISBN: 0738428094.