

Apache Turbine – economie de timp prin folosirea framework-urilor

Data: 05.02.2002

Autor: Ernst Leohard, SecureNet GmbH Munchen (el@secure-net.de), tradus și adaptat de Liviu Marinescu, SecureNet S.R.L. (lm@secure-net.ro)

Compania: SecureNet S.R.L., Craiova, tel.: 051/410555, <http://www.secure-net.ro>

În majoritatea proceselor ce implică un volum mare de date, timpul este o resursă foarte importantă. O modalitate de folosire cât mai optimă a acestei resurse o reprezintă folosirea framework-urilor. Turbine este o implementare practică a conceptului de framework care se bazează pe tehnologia Java Servlets pentru furnizarea de aplicații Web. Pentru programatorii experimentați ce folosesc limbajul Java, Turbine face posibilă dezvoltarea într-un timp foarte scurt a aplicațiilor web dorite. Compania SecureNet a folosit acest framework cu succes în proiecte ca McDialog.net, vizibil la adresa <http://www.mcdialog.net> și SN.Portal, vizibil parțial la adresa <http://www.bals.ro>

Caracteristici Turbine

- Integrarea cu sisteme de creare de șabloane Velocity, WebMacro, FreeMarker și JSP.
- Database Connection Pool, cu suport pentru majoritatea sistemelor de gestiune de baze de date.
- Parsarea parametrilor pentru GET/POST/PATH_INFO.
- Gestiunea acțiunilor pe bază de evenimente.
- Unelte de mapare a obiectelor relaționale: DatabaseMap/Peers, Village, Castor.
- IDBroker care abstractizează utilizarea secvențelor/autoincrement pe baze de date.
- Scheduler system - echivalentul funcției „Cron” din UNIX; un program care permite pornirea automată și regulată de programe la anumite date/ore.
- GlobalCache - mecanism persistent de stocare a obiectelor între requests.
- DateSelector - utilitar pentru construirea de meniuri pop-up HTML.
- File Upload API.
- Integrare XMLRPC.
- Servicii de localizare.
- Stocarea temporară sau permanentă a datelor sesiune pentru utilizatori.
- Clasa BrowserDetector pentru detectarea browserului client.
- Clasa GenerateUniqueId class pentru generarea de numere unice pentru sesiuni.
- Integrare cu JavaMail și cu Velocity/WebMacro pentru trimiterea de șabloane Velocity ca e-mail-uri.

Componente refolosibile & Tool-uri

Turbine pune la dispoziție un set de componente întâlnite în majoritatea aplicațiilor Web cum ar fi upload-area fișierelor, un Connection Pool pentru bazele de date, un program de E-mail bazat pe șabloane șamd. În plus, Turbine integrează un set de tehnologii și unelte cum ar fi XML/RPC, XSL/XSLT, JavaMail sau motoarele de șabloane WebMacro și Velocity. Aceste unelte sunt puse la dispoziția dezvoltatorului de aplicații prin așa-numitele "servicii Turbine" cu un API ce poate fi folosit ușor. În acest fel se scurtează timpul investit de către un programator în învățarea noilor tehnologii și API-uri.

În secțiunea următoare se prezintă codul necesar pentru a face un servlet să opereze ca un XML/RPC - Server și a putea să accepte cereri XML/RPC:

```
/* xs este implementarea default a XmlRpcService*/
XmlRpcService xs =
  (XmlRpcService)((TurbineServices) TurbineServices.getInstance()).
    getService(XmlRpcService.SERVICE_NAME );

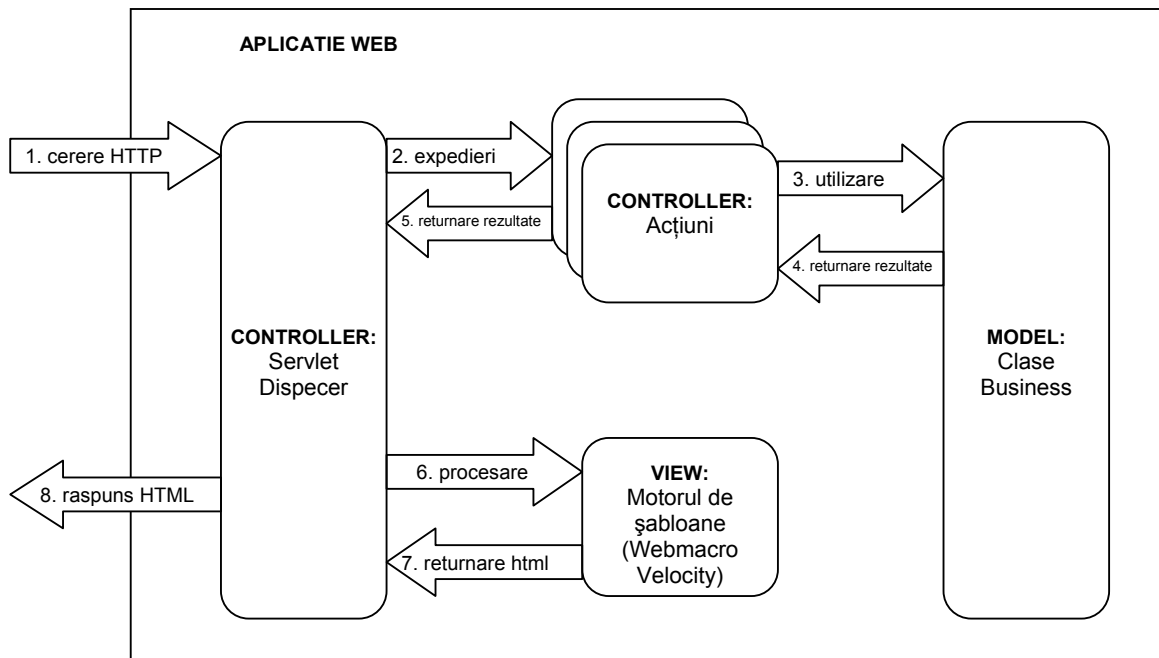
/* Cererea XMLRPC este selectată și implementată din InputStream-ul
 * obiectului HttpServletRequest. Rezultatul este scris în obiectul
 * OutputStream al HttpServletResponse.
 */
res.getOutputStream().write( xs.handleRequest(req.getInputStream()) );
```

În plus trebuie completată o intrare în fișierul de configurare al Turbine, care va defini ce clase vor fi disponibile pentru XMLRPC.

```
services.XmlRpcService.handler.someHandler=de.securenet.xmlrpc.SomeManager
```

Arhitectură: Model View Controller

Turbine nu este numai o colecție de componente și unelte folosite. Puterea sa se bazează mai mult pe arhitectura Framework-urilor orientate pe evenimente, ceea ce conduce dezvoltatorul software la o arhitectură multinivel corespunzătoare modelului MVC. Figura următoare prezintă această arhitectură și descrie modul de prelucrare a unei cereri HTTP, respectiv obiectul Request:



Servletul central numit și dispecer, transmite fiecare cerere HTTP către o clasă specifică, ce va prelucra evenimentul respectiv. Din interiorul acestei clase, (*action class*), se face legătura la stratul Model prin folosirea unuia sau mai multor obiecte ce conțin logica aplicației (*business logic*). În funcție de rezultat, clasa acțiune decide mersul aplicației și pune datele rezultate într-un obiect comun atât servletului central cât și clasei respective. În acest fel, servletul dispecer și clasa acțiune alcătuiesc stratul numit generic "CONTROLLER". Stratul "VIEW" este implementat folosind, de exemplu, tehnologia Velocity. Clasa acțiune decide ce șablon va fi folosit. Procesul este realizat de servletul dispecer.

Torque – utilitar de mapare pe obiecte a unei baze de date relaționale

Framework-ul Turbine poate fi folosit opțional împreună cu utilitarul Torque de mapare pe obiecte a unei baze de date relaționale. Pe baza modelului bazei de date (Entity Relationship Model, ERM) care se definește folosind sintaxa XML, sunt generate pentru fiecare entitate din modelul bazei de date câteva clase Java. Astfel, dezvoltatorul nu mai este nevoit să utilizeze direct SQL ci se va folosi de clasele generate.

```
<app-data>
  <database>
    <table name="category" idMethod="autoincrement">
      <column name="category_id" required="true" autoIncrement="true"
        primaryKey="true" type="INTEGER"/>
      <column name="name" size="100" type="VARCHAR"/>
    </table>
    <table name="item" idMethod="autoincrement">
      <column name="item_id" required="true" autoIncrement="true"
        primaryKey="true" type="INTEGER"/>
      <column name="name" size="100" type="VARCHAR"/>
      <column name="price" required="true" type="INTEGER"/>
      <column name="category_id" required="true" type="INTEGER"/>
      <foreign-key foreignTable="category">
        <reference local="category_id" foreign="category_id"/>
      </foreign-key>
    </table>
  </database>
</app-data>
```

Exemplu de folosire a instrucțiunii Sql INSERT:

```
// Inserarea unui nou obiect de tip Item
Item itm = new Item();
itm.setName ("New Item");
itm.setPrice (100);
itm.setCategoryId (1);
Object o = ItemPeer.doInsert (itm);
```

Exemplu de folosire a clauzei Sql SELECT:

```
// toate obiectele de tip Items ce au ID 2
Criteria crit = new Criteria();
crit.add (ItemPeer.CATEGORY_ID,2);
Vector v = ItemPeer.doSelect (crit);
```

Expansiunea

Fiecare serviciu pus la dispoziția utilizatorului de către Turbine constă în:

1. o interfață Java
2. o implementare implicită a fiecărui serviciu
3. un obiect Factory, care întoarce și livrează serviciile instanței

În funcție de configurarea Turbine, implementarea implicită poate fi oprită. Intrarea pentru serviciul XML/RPC poate arăta, de exemplu, în felul următor:

```
services.XmlRpcService.classname  
=org.apache.turbine.services.xmlrpc.TurbineXmlRpcService
```

Concluzii

Compania SecureNet a decis să utilizeze framework-ul Turbine în proiectul MC.Dialog (<http://www.mcdialog.net>). Unele dintre componente au fost rescrise pentru a permite integrarea unor extensii speciale în cadrul proiectului, rezultând o arhitectura proprie, denumită SN.Turbine. Această arhitectură se folosește momentan la dezvoltarea unui portal personalizat, pentru extinderea funcționalității site-ului <http://www.bals.ro>. O dată cu rescrierea servletului dispecer, dezvoltatorul aplicației a avut controlul total asupra modului de funcționare a aplicației, fără a avea nevoie de toate serviciile și componentele specifice Turbine. Folosirea interogărilor complexe SQL au dus la apariția anumitor erori în clasele generate. Acestea au trebuit rescrise. Pe ansamblu, framework-ul Turbine a fost evaluat foarte pozitiv iar timpul de implementare al aplicațiilor s-a scurtat simțitor prin folosirea lui. Deoarece acum există cunoștințe despre modul de lucru Turbine, slăbiciunile și punctele forte ale acestuia, aplicațiile ulterioare în cadrul firmei vor profita de acest fapt, oferind o arhitectura modernă, flexibilă și stabilă într-un timp record de implementare.

Resurse

1. Turbine: <http://jakarta.apache.org/turbine>
2. SecureNet GmbH: <http://www.secure-net.de>
3. MCDialog: <http://www.mcdialog.net>
4. WebMacro: Java-based Template Engine, <http://www.webmacro.org>
5. Velocity: Java-based Template Engine, <http://jakarta.apache.org/velocity>
6. XMLRPC: <http://www.xmlrpc.com>

Despre autor

Ernst Leonhard lucrează ca Senior Software-Engineer la firma SecureNet în München, o companie specializată pe dezvoltarea de aplicații și soluții software pentru Internet, Intranet și Extranet (<http://www.secure-net.de>). El are o experiență de 10 ani în dezvoltarea de software, dintre care 6 ani în folosirea tehnologiilor web.