

[Ejb3 e Flex 2: Installare il server JBoss - Parte 1](#)

(, 02-03-2007 08:48)

Visto l'enorme interesse suscitato da Flex nella comunit  degli sviluppatori Java, abbiamo pensato di introdurre una serie di articoli sull'integrazione tra Flex 2 e l'ultima interessante specifica introdotta in J2ee 5: gli Ejb3. In questi articoli metteremo in luce le nuove interessanti novit  introdotte dagli Ejb3 e a come sfruttare con Flex.

Gli Enterprise Java Bean sono componenti remoti che furono introdotti nella piattaforma Java Enterprise Edition per semplificare lo sviluppo di applicazioni distribuite, robuste e scalabili nel tentativo di sostituire tecnologie esistenti abbastanza complesse come Corba. In realt  queste specifiche si sono evolute fino a racchiudere funzionalit  legate non solo alla possibilit  di invocare procedure remote ma anche alla sicurezza (Java Authentication and Authorization Service), alla messaggistica (Java Message Service), ai web services, alla persistenza (Java Persistence API), alla schedulazione di eventi (Timer Service), e altro ancora.

Mentre lo sviluppo di servizi di backend si sia molto semplificato con l'introduzione degli Ejb3, lo sviluppo del frontend grafico si   sicuramente potenziato grazie all'introduzione delle Java Server Faces e Ajax, ma non si pu  dire che si sia semplificato, anzi... Da qui l'enorme interesse nell'interfacciare la piattaforma J2ee con Flex che rappresenta lo stato dell'arte di un presentation tier.

Per interfacciare le applicazioni Flex e gli Ejb3   necessario per  installare i Flex Data Services poich  bisogna abilitare la comunicazione attraverso il protocollo binario AMF di Flash, che richiede l'installazione di componenti aggiuntivi sul server remoto. Tramite i Flex Data Services in realt  si possono sfruttare anche altre nuove interessanti funzionalit  , che cercheremo, nel proseguimento degli articoli di questa serie, di toccare con mano insieme alle nuove caratteristiche introdotte con gli Ejb3, vedendo come queste due piattaforme si integrino splendidamente nello sviluppo di complesse applicazioni enterprise.

Per cominciare dovremo installare un application server J2ee compliant necessario per far funzionare gli Ejb3. La scelta di JBoss   dovuta alla completezza e qualit  di questo application server Open Source, senza dimenticare in realt  che le applicazioni che scriveremo seguiranno le specifiche standard J2ee 5 per cui saranno pienamente portabili anche verso altri server J2ee 5 compliant sia Open Source (JOnAS, Apache Geronimo, ...) che commerciali (WebSphere, Oracle AS, WebLogic, ...).

Installazione di JBoss con Ejb3

Prima di installare JBoss   necessario aggiornare la nostra versione del JDK alla versione 5 o superiori, poich  gli Ejb3 sfruttano le annotazioni, novit  introdotta dalla versione 5.0. Questa caratteristica, tra le novit  degli Ejb3, permette di "annotare" direttamente le classi che implementiamo con metadati che ne descrivano il comportamento all'interno di un container Ejb, rendendo non pi  necessario definire la nostra applicazione con scomodi XML di configurazione.

Per chi non volesse perdere troppo tempo per l'installazione di JBoss con l'ultima versione degli Ejb3   disponibile sul sito ufficiale <http://www.jboss.org>, un installer online che prende il nome di Jems installer (vedi sul sito il link download Web Install sul progetto JBoss AS). Il Jems installer   anche scaricabile per un'installazione offline...vedete voi cosa preferite! :)

Avviata la procedura di installazione del Jems installer   sufficiente rispondere quanto segue per avere un JBoss perfettamente abilitato agli Ejb3:

- Selezionate la lingua "English" e premete OK

- Premete Next
- Selezionate "I accept the license...." e premete Next
- Scegliete una cartella per l'installazione e premete Next
- Scegliete il profilo "ejb3" e premete Next
- Premete Next
- Scegliete l'opzione "Standard" e premete Next
- Premete Next
- Premete Next
- Premete Done

Configurazione e avviamento di JBoss

A questo punto prima di lanciare JBoss per verificare l'installazione, analizziamo la struttura delle directory installate e modifichiamo la configurazione affinché possiamo avviarlo correttamente. Le cartelle piÃ¹ importanti di sono:

- JBOSS_HOME/bin - contiene gli script di lancio del server
- JBOSS_HOME/client - contiene le librerie jar per sviluppare un client che acceda a JBoss
- JBOSS_HOME/docs - contiene esempi di file di configurazione, DTD, XSD e applicazioni di prova
- JBOSS_HOME/lib - contiene le librerie necessarie all'avvio di JBoss
- JBOSS_HOME/server - contiene i profili di avvio di JBoss
- JBOSS_HOME/server/default - contiene la configurazione di avvio del profilo di default
- JBOSS_HOME/server/default/conf - contiene i file di configurazione
- JBOSS_HOME/server/default/deploy - contiene le applicazioni J2ee installate
- JBOSS_HOME/server/default/lib - contiene le librerie comuni alle varie applicazioni installate

Per avviare JBoss Ã¨ necessario modificare lo script di avvio JBOSS_HOME/bin/run.bat ed aggiungere all'inizio dello script il settaggio della variabile d'ambiente JAVA_HOME e inoltre cercate e modificate la riga che indica il settaggio della variabile per abilitare il debug remoto tramite JPDA:

```
set JAVA_HOME=../../jdk5.0
set JAVA_OPTS=-Xdebug -Xrunjdw:transport=dt_socket,address=8000,server=y,suspend=n
%JAVA_OPTS%
```

Il secondo settaggio prepara JBoss ad ascoltare sulla porta 8000 un'eventuale richiesta di debug remoto. Approfondiremo l'utilizzo del debug remoto in qualche articolo piÃ¹ avanti ma ricordatevi perÃ² che se la porta 8000 Ã¨ giÃ impegnata, Ã¨ necessario cambiarla su un'altra disponibile. Inoltre assicuratevi che ci sia l'opzione "suspend=n" perchÃ© altrimenti il server non parte.

A questo punto per lanciare JBoss Ã¨ sufficiente eseguire lo script run.bat e il server sarÃ pronto quando sulla console apparirÃ il messaggio:


```
Terminal — java — 80x24
:service=DataSourceBinding,name=DefaultDS' to JNDI name 'java:DefaultDS'
12:07:59,863 INFO [A] Bound to JNDI name: queue/A
12:07:59,867 INFO [B] Bound to JNDI name: queue/B
12:07:59,868 INFO [C] Bound to JNDI name: queue/C
12:07:59,869 INFO [D] Bound to JNDI name: queue/D
12:07:59,870 INFO [ex] Bound to JNDI name: queue/ex
12:07:59,910 INFO [testTopic] Bound to JNDI name: topic/testTopic
12:07:59,913 INFO [securedTopic] Bound to JNDI name: topic/securedTopic
12:07:59,915 INFO [testDurableTopic] Bound to JNDI name: topic/testDurableTopic
12:07:59,919 INFO [testQueue] Bound to JNDI name: queue/testQueue
12:07:59,968 INFO [UILServerILService] JBossMQ UIL service available at : /0.0.
0.0:8093
12:08:00,010 INFO [DLQ] Bound to JNDI name: queue/DLQ
12:08:00,264 INFO [ConnectionFactoryBindingService] Bound ConnectionManager 'jb
oss.jca:service=ConnectionFactoryBinding,name=JmsXA' to JNDI name 'java:JmsXA'
12:08:00,318 INFO [TomcatDeployer] deploy, ctxPath=/jmx-console, warUrl=.../dep
loy/jmx-console.war/
12:08:00,725 INFO [Http11BaseProtocol] Starting Coyote HTTP/1.1 on http-0.0.0.0
-8080
12:08:00,884 INFO [ChannelSocket] JK: ajp13 listening on /0.0.0.0:8009
12:08:00,910 INFO [JkMain] Jk running ID=0 time=0/54 config=null
12:08:00,927 INFO [Server] JBoss (MX MicroKernel) [4.0.5.GA (build: CVSTag=Bran
ch_4_0 date=200610162339)] Started in 21s:799ms
```

Per verificare il corretto funzionamento di JBoss Ã" sufficiente collegarsi con il browser all'indirizzo <http://localhost:8080> dove comparirÃ" la pagina iniziale di JBoss Server:



Per gli utenti che installano JBoss non in un'ambiente Windows, lo script da modificare per l'avvio Ã¨ il run.sh e le variabili d'ambiente da settare sono le stesse ma fatte con i comandi linux (ricordatevi di rendere lo script run.sh eseguibile).

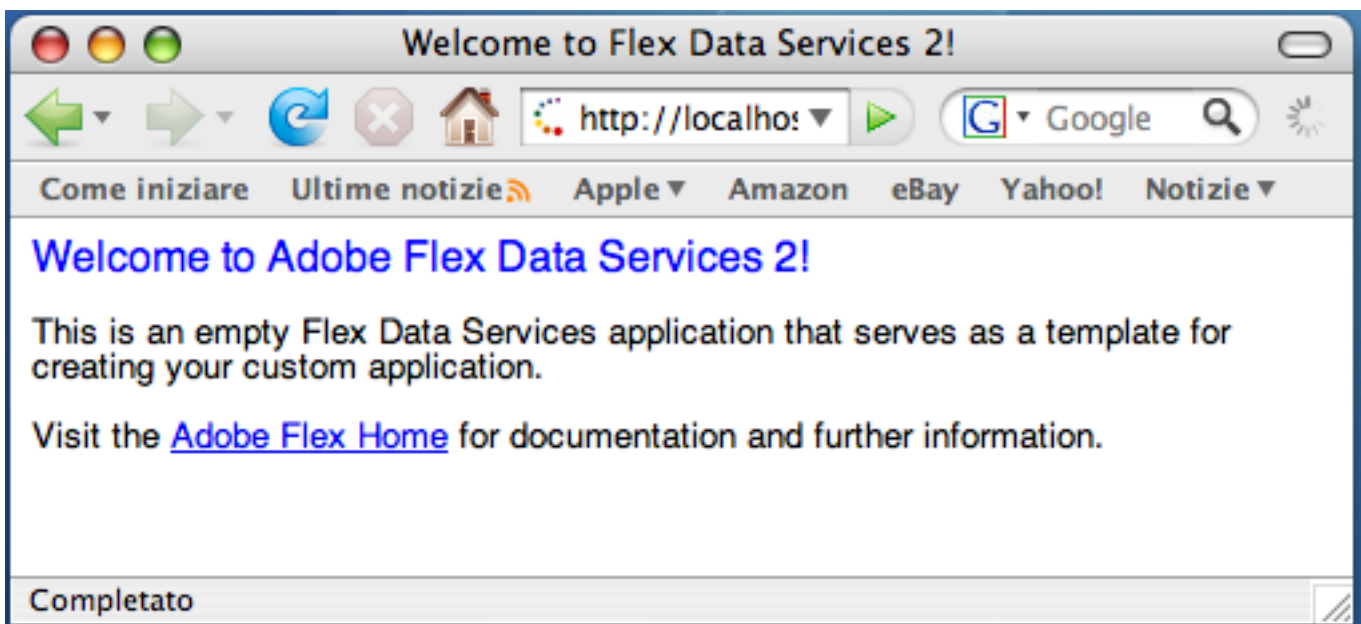
Avviamento di Flex Data Services

A questo punto Ã¨ necessario installare i Flex Data Services sulla nostra macchina se non l'abbiamo giÃ fatto. Per l'installazione di Flex Data Services rimandiamo all'articolo del collega Marco Casario pubblicato su Flex Gala il 14/12/2006: "Applicazioni Flex in ambiente J2EE: installare i Flex Data Services sotto Tomcat" (vedi i riferimenti in fondo all'articolo).

A questo punto Ã¨ sufficiente recuperare l'applicazione template "flex.war" dalla cartella d'installazione di FDS e scompattarla in una cartella il cui nome sia sempre "flex.war". La cartella "flex.war" rappresenta il progetto template FDS da cui cominceremo lo sviluppo, prima perÃ² installiamola in JBoss copiandola nella directory JBOSS_HOME/server/default/deploy. L'applicazione Flex dovrÃ quindi essere JBOSS_HOME/server/default/deploy/flex.war, facendo attenzione a mantenere l'estensione .war nel nome della directory per permettere a JBoss di identificare la cartella come applicazione Web. Grazie alla funzionalitÃ di deploy automatico sulla cartella deploy, JBoss installerÃ l'applicazione e sulla console di JBoss appariranno i messaggi:


```
Terminal — java — 80x24
oss.jca:service=ConnectionFactoryBinding,name=JmsXA' to JNDI name 'java:JmsXA'
12:08:00,318 INFO [TomcatDeployer] deploy, ctxPath=/jmx-console, warUrl=.../dep
loy/jmx-console.war/
12:08:00,725 INFO [Http11BaseProtocol] Starting Coyote HTTP/1.1 on http-0.0.0.0
-8080
12:08:00,884 INFO [ChannelSocket] JK: ajp13 listening on /0.0.0.0:8009
12:08:00,910 INFO [JkMain] Jk running ID=0 time=0/54 config=null
12:08:00,927 INFO [Server] JBoss (MX MicroKernel) [4.0.5.GA (build: CVSTag=Branc
h_4_0 date=200610162339)] Started in 21s:799ms
12:11:21,105 INFO [TomcatDeployer] deploy, ctxPath=/flex, warUrl=.../deploy/fle
x.war/
12:11:22,229 INFO [STDOUT] 25/02 12:11:22 INFO Loading configuration file /User
s/roberto/Applications/jboss-corso/server/default/deploy/flex.war/WEB-INF/flex/f
lex-webtier-config.xml
12:11:22,539 INFO [STDOUT] 25/02 12:11:22 INFO Loading configuration file /User
s/roberto/Applications/jboss-corso/server/default/deploy/flex.war/WEB-INF/flex/f
lex-config.xml
12:11:22,787 ERROR [STDERR] 25-feb-2007 12.11.22 org.apache.catalina.core.Applic
ationContext log
INFO: FlexMxmlServlet: Starting Adobe Flex Web Tier Compiler
12:11:22,792 ERROR [STDERR] 25-feb-2007 12.11.22 org.apache.catalina.core.Applic
ationContext log
INFO: FlexMxmlServlet: Adobe Flex Web Tier Compiler Build: 155539
```

A questo punto per verificare il corretto funzionamento di Flex su JBoss Ã” sufficiente collegarsi con il browser all'indirizzo <http://localhost:8080/flex> dove comparirÃ” la pagina iniziale di Flex Data Service:



Per stoppare il server Ã” sufficiente premere Ctrl^C dalla console o lanciare lo script `JBOSS_HOME/bin/shutdown.bat`.

Nel prossimo articolo vedremo come creare il nostro primo progetto Ejb3, come si effettua il deploy su JBoss

e a come sia possibile testarne il funzionamento. Per chi freme per l'interfacciamento con Flex perÃ², deve resistere ancora un pÃ² :).

Riferimenti

Sito ufficiale Ejb: <http://java.sun.com/products/ejb/>

Sito JBoss: <http://labs.jboss.com/portal/>

Sito JBoss application server: <http://labs.jboss.com/portal/jbossas>

Sito JBoss implementazione Ejb3: <http://labs.jboss.com/portal/jbossejb3>

Articolo sull'installazione dei Flex Data Services:

<http://augitaly.com/flexgala/index.php?cmd=newsreader&id=32>

Note sull'autore

Roberto Tassi (roberto.tassi@aminoweb.com) Ã¨ un consulente Sun Certificated Java Programmer & Developer, che da 8 anni si occupa di sviluppo di applicazioni web-oriented su piattaforma Java. La sua esperienza web Ã¨ partita da Cold Fusion, per poi consolidarsi su Java fino ad approdare al fantastico mondo di Flex.