



## Esercitazione JMX

Alma Mater Studiorum - Università di Bologna  
CdS Laurea Magistrale in Ingegneria Informatica  
I Ciclo - A.A. 2017/2018  
**Sistemi Distribuiti M (8 cfu)**

Proposta di esercizio:  
**Gestione di Parametri Operativi  
tramite JMX**

A cura di:

Samuele Pasini [samuele.pasini@gmail.com](mailto:samuele.pasini@gmail.com)



## Obiettivo

Data l'applicazione Web di esempio, per la gestione di "Libri", "Autori" e "Editori" in una ipotetica biblioteca...

E data una versione "riveduta e corretta" dell'implementazione di esempio degli oggetti DAO basati su JDBC, **ora in grado di sfruttare meccanismi di "pooling" delle connessioni** forniti dalle librerie del progetto *Apache Commons*...

- Introdurre un **componente JMX** in grado di
  - variare a runtime le **dimensioni del pool di connessioni**
  - permettere a runtime di **scegliere il tipo di database** utilizzato (MySQL o Hsqldb)
- **Utilizzare la Java Console** per
  - **individuare** i componenti JMX esposti dalla JVM in uso
  - **modificare i parametri** del componente JMX sviluppato



## Dettagli...

- Il progetto contenente il codice su cui basare lo sviluppo...
- ❑ Contiene una **applicazione Web** minimale, una **suite di test** e un insieme di classi per l'inizializzazione della base di conoscenza
  - ❑ Permette, attraverso ANT, di eseguire il deployment della applicazione su un'installazione del Web Server Tomcat
  - ❑ Contiene una **versione modificata dell'implementazione JDBC degli oggetti DAO** per MySql e Hsqldb
    - Le factory di entrambi i tipi estendono ora una comune superclasse che **realizza il meccanismo di pooling**
    - Le factory di entrambi i tipi espongono **opportuni metodi di gestione** che il componente JMX dovrà richiamare per svolgere i propri compiti
    - La factory inizialmente in uso è indicata nel descrittore **web.xml** attraverso un parametro il cui valore è memorizzato in **ServletContext**
  - ❑ Specifica l'interfaccia di gestione **it.unibo.sdls.sampleproject.jmx.ManagedDAOFactoryMBean** che il componente JMX deve implementare



## Criticità (1)

### Registrazione dei componenti JMX

- È necessario recuperare *mbeanserver* di default della JVM su cui esegue *Tomcat* e registrare su di esso il nuovo componente
- È necessario, all'avvio del server, fornire opportune opzioni alla JVM affinché permetta il management JMX attraverso la Java Console

**Nota:** per agganciare la JavaConsole ai PID dei processi di tipo Java in esecuzione sulla macchina, occorre assicurarsi che il nome dell'host coincida con l'indirizzo di loopback o con un indirizzo di rete valido

In Ubuntu, a default, */etc/hosts* lega il nome "localhost" a IP 127.0.1.1 anziché 127.0.0.1



## Criticità (2)

### **Integrazione tra componenti di management e componenti Web**

- ❑ È necessario inizializzare il componente JMX **all'avvio dell'applicazione Web** perché esso decide e governa le modalità di accesso al DB utilizzate dalle altre parti dell'applicazione
  - > Recuperando da *ServletContext* il tipo di implementazione DAO di default
  - > Predisponendo tale implementazione per l'uso da parte degli altri componenti
- ❑ Tomcat, come la maggior parte dei server Web e degli AS, permette isolamento tra le diverse applicazioni ospitate attraverso un meccanismo di **class loading gerarchico**
  - > Tale meccanismo influenza le visibilità delle classi
  - > *mbeanserver* di default della JVM esegue nel class loader "radice", mentre le applicazioni Web in class loader "figli"



## Un suggerimento...

All'interno della pagina JSP *index.jsp*, la riga di codice 56...

```
DAOFactory daoFactory = (DAOFactory) application.getAttribute("daoFactory");
```

indica una sicura possibile soluzione per condividere l'oggetto Factory tra componente JMX e componenti Web

Fornire al componente JMX accesso al contesto dell'applicazione Web (**ServletContext** negli oggetti Java, **application** nelle pagine JSP) permette certamente di soddisfare le specifiche dell'esercizio

Altre soluzioni sono ovviamente possibili, anche modificando o riscrivendo di conseguenza il codice della pagina JSP!



## Riferimenti

Java Management Extension (JMX) Technology Tutorial

<http://java.sun.com/j2se/1.5.0/docs/guide/jmx/tutorial/tutorialTOC.html>

Esempi per Java Management Extension (JMX)

<http://java.sun.com/j2se/1.5.0/docs/guide/jmx/examples.html>

Abilitare l'agente JMX

<http://java.sun.com/j2se/1.5.0/docs/guide/management/agent.html#jmxagent>

MBeans e Tomcat: Howto Guide

<http://oss.wxnet.org/mbeans.html>