

Getting Started with JPA using Hibernate 3

Reza Rahman

Author, *EJB 3 in Action*

Chief Architect, Tripod Technologies

Connecticut Java User Group

September 2007

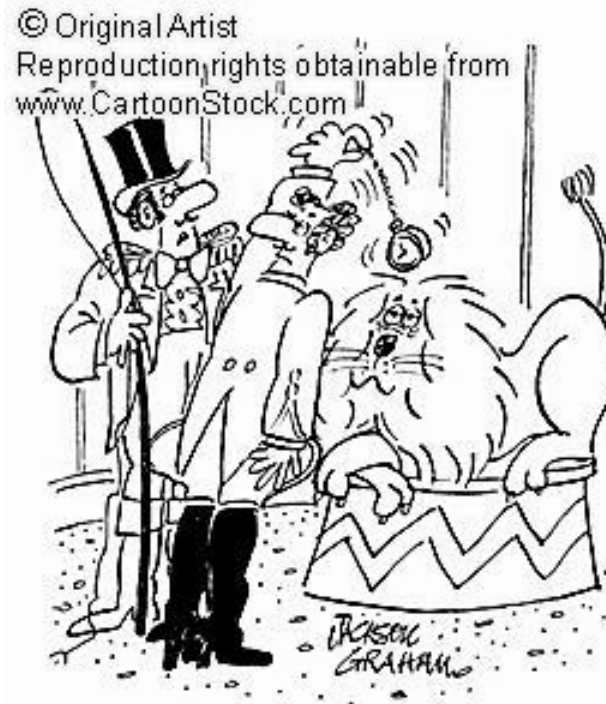
Java Persistence API (JPA): The Java ORM Standard

- The Java Persistence API (JPA) is the object relational mapping persistence standard for Java.
- JPA is part of EJB 3 and is a replacement for the faulty EJB Entity Beans standard.
- JPA can be used outside an EJB container anywhere a JDK can be deployed.
- JPA fully embraces POJO development, annotation-driven programming and the API paradigm as opposed to the container paradigm.
- If you have used Hibernate, TopLink or JDO, JPA will seem very familiar. Indeed, all these products are likely to be JPA compliant persistence providers.
- There are three major parts to JPA:
 - **Entities:** Persistent objects that are stored in a relational database.
 - **The EntityManager API:** The API for performing actual persistence operations such as saving, deleting, and retrieving entities.
 - **The Java Persistence Query Language (JPQL):** An SQL-like language used to retrieve and manipulate entities.

Hibernate 3: The Most Widely Deployed JPA Persistence Provider

- **Hibernate is the most widely used Java ORM tool.**
 - **The popularity of Hibernate was a major catalyst in the demise of Entity Beans and the adoption of JPA into the EJB standard.**
 - **The enhancements in Hibernate 3 makes Hibernate fully JPA compliant. The enhancements include:**
 - Support for annotations.
 - Support for the EntityManager API.
 - Enhanced support for native queries.
 - Enhanced entity, relation and inheritance mapping support.
 - Better support for JTA transactions.
 - Bulk update and delete queries.
 - **Hibernate 3 is likely to be the most popular JPA provider.**
 - **Hibernate is the default persistence provider for the JBoss Application Server.**
- This session is a very-hand on introduction to JPA using Hibernate 3!**

Let the Games Begin...



'JUST HOW LONG HAVE YOU BEEN
TAMING LIONS?'

You will need...

- Java SE 5 or higher

http://java.sun.com/javase/downloads/index_jdk5.jsp

- Hibernate 3 (Core and EntityManager)

<http://www.hibernate.org/6.html>

- HSQLDB

<http://www.hsqldb.org>

- NetBeans 5.5 or Eclipse Dali (included in the Europa release with WTP 2.0)

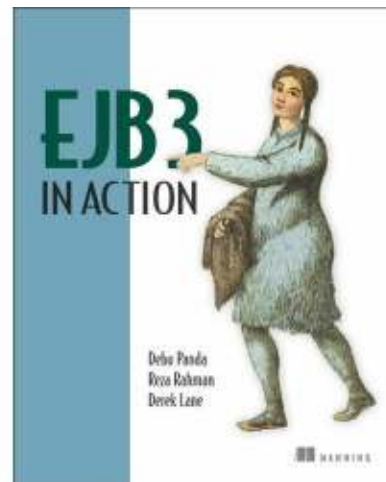
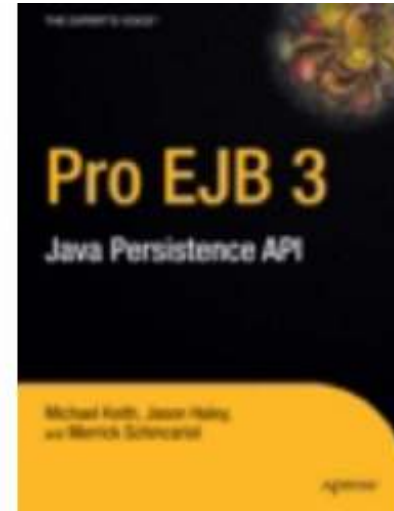
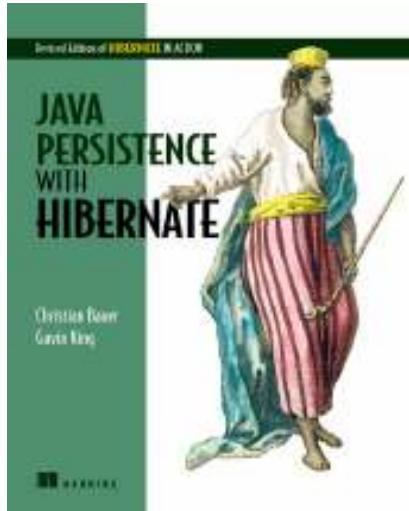
<http://www.netbeans.info/downloads/index.php>

<http://www.eclipse.org/downloads>

Some More JPA Features

- **Using JPA in managed environments (EJB 3 application servers):**
Injecting managed JPA objects, using JTA for transactions.
- **One-to-one and many-to-many relations:**
Annotations for mapping unidirectional and bidirectional relations.
- **Join tables:**
The most complex and flexible relationship mapping mechanism.
- **Compound primary keys:**
Mapping compound or natural primary keys.
- **Mapping inheritance:**
Three different ways of modeling class hierarchies.
- **Entity life-cycle listeners:**
The entity life-cycle and event callbacks.
- **JPQL DELETE and UPDATE statements:**
Bulk persistence operations through JPQL.
- **Native queries**
Using database specific SQL when needed.

References



Shameless plug alert!

Frequently Asked Questions

- **Why use JPA?**
 - It has a small, simplified API.
 - Has much better integration with EJB 3, the Spring framework and the like.
 - It is possible to easily switch persistence providers.
 - Advantages of standardization.
- **When should I use Hibernate 3 instead?**
 - If you cannot deploy to Java SE 5.
 - Supports stored procedures.
 - Supports iBATIS-style SQL-centric mapping of persistence operations.
 - Slightly more feature rich in mapping and object queries (query by example, query by criteria).
 - Advantages of a non-standard open source platform.
- **What other JPA persistence providers are available?**
 - TopLink, TopLink Essentials, BEA Kodo, OpenJPA