

Mappare il DB senza XML

Introduzione alle annotation di Hibernate3

a cura di Lucio Benfante



Java User Group Padova



- Hibernate è un framework che realizza un object/relational mapping (ORM) fra oggetti Java e strutture di un database relazionale
- Permette allo sviluppatore di utilizzare oggetti Java che vengono resi persistenti sul database, svincolandolo dai problemi di utilizzo diretto del database
- E' una soluzione non intrusiva: gli oggetti persistenti sono dei semplici JavaBean, senza nessun requisito ulteriore

```

public static void main(String[] args)
    throws HibernateException {

    SessionFactory sessionFactory =
        new Configuration()

            .configure().buildSessionFactory();

    Session hSession = sessionFactory.openSession();
    Transaction tx = hSession.beginTransaction();
    Autore autore = new Autore();
    autore.setName("Lucio");
    autore.setCognome("Benfante");
    autore.setNote("Il primo autore del nostro DB");
    hSession.save-autore);
    tx.commit();
    hSession.close();
  }

```



- Castor (www.castor.org)
- Apache JDO, OJB e Torque(db.apache.org)
- JPOX (www.jpox.org)
- iBatis (www.ibatis.net)
- Ammentos (ammentos.biobytes.it)
- Entity EJB
- Java Persistence API (JPA)
- ...



- Il mapping fra gli oggetti Java e il database viene definito mediante dei file XML
- Con la versione 3 di Hibernate è possibile definire il mapping anche mediante delle annotation Java 5



**POJO
classes**

Plain Old Java Object

Le classi persistenti che implementano il domain model dell'applicazione. Sono dei semplici JavaBean.



**XML
Hibernate
mappings**

File XML (.hbm.xml) che definiscono il mapping delle classi Java sugli elementi del database.



**SQL
DDL**

Script SQL che definiscono il database.

annotated
POJO
classes

Plain Old Java Object

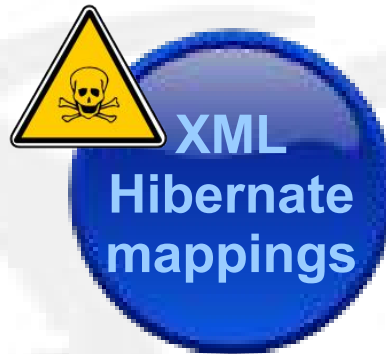
Le classi persistenti che implementano il domain model dell'applicazione.

Sono dei semplici JavaBean.

Il mapping è espresso dentro le classi stesse mediante annotazioni Java 5

SQL
DDL

Script SQL che definiscono il database.



SchemaExport
SchemaUpdate



Annotando le classi e usando i tool **SchemaExport** o **SchemaUpdate** di Hibernate gli script SQL di definizione del database possono essere generati automaticamente.



- sono metadati che vengono aggiunti al codice Java per descriverne caratteristiche particolari
- sono modificatori applicati a package, dichiarazioni di tipo, costruttori, metodi, attributi, parametri e variabili
- possono essere utilizzate a livello di codice sorgente o di classe compilate, e anche a runtime mediante reflection
- Le annotazioni predefinite nella JSE 5 sono: Override, Deprecated e SuppressWarnings
- E possibile, e molto semplice, definire delle proprie annotazioni

```

public class AnnotatedToStringHelperTestBean {
    @Exposed private char a = 'a';
    private byte b = 1;
    @Exposed private float f = 5.0F;
    private double g = 6.0;
    private boolean h = true;

    @Override
    public String toString() {
        ToStringHelper.setWorker(
            new AnnotatedFieldsToStringWorker());
        return ToStringHelper.toString(this);
    }
}

```

```

@Documented
@Target ({ElementType.FIELD})
@Retention (RetentionPolicy.RUNTIME)
@Inherited
public @interface Exposed {

    // eventuali definizioni di proprietà
    // ad es:
    //
    // String value() default "Esempio";
    //

}

```



- Annotazioni standard per il mapping O/R della specifica JSR 220 (Enterprise JavaBeans™ 3.0)
 - Estensioni specifiche per il mapping O/R di Hibernate
 - Validazione dei dati
- ...inoltre tramite l'Hibernate EntityManager viene implementato un completo EJB3 persistence provider (standalone)

- ...andiamo sul codice!





- Sito di Hibernate
www.hibernate.org
- Christian Bauer and Gaving King, “Hibernate in action”, 2005, Manning
- JSR 175: A Metadata Facility for the Java™ Programming Language
<http://jcp.org/en/jsr/detail?id=175>
- JSR 220: Enterprise JavaBeans™ 3.0
<http://www.jcp.org/en/jsr/detail?id=220>

- **Sito Web:**
 - <http://www.jugpadova.it>
- **Mailing List:**
 - http://groups.yahoo.com/group/JUG_Padova/
- **Persone di riferimento**
 - Dario Santamaria (dario.santamaria@jugpadova.it)
 - Lucio Benfante (lucio.benfante@jugpadova.it)
 - Paolo Donà (paolo.dona@jugpadova.it)