Annotations:

Two types of annotation:

1. Basic annotation- implemented JPA annotations
2. Hibernate annotations- extended specific to hibernate

Basic annotations>

* @Entity
* @Table

component persistence is called entity in EJB

JAVA EE EJB – cost heavily because of heavy server

In 2003, JSR 220 proposed CMP(container managed persistence requires light weight framework )

Java5EE have JPA java persistence API specification

Coming back to spring

Spring have 3 persistence mechanism

1. JDBC
2. ORM Technologies
3. JAVA DAO

SessionFactory sessionFac= new Configuration().configure().buildSessionFactory();

Session session = sessionFac.openSession();

Transaction tx=null;

Try{

Tx= session.beginTransaction();

List list= session.createQuery(“from Message”).list();

Tx.commit();

Tx= null;

Return list;

}catch(Hibernate Exception){

Tx.rollback();

}finally{

Session.close();

}

**Hibernate Mapping file:**

<hibernate-mapping>

<class name=”pojoname” table=”tableName”>

<id type=”int” column=”id”>

<generator class=”native” />

</id>

<property name ,column,type/>

</class>

</hibernate-mapping>

Hibernate configuration file:

Configuration =>

Hibernate.cfg.xml

DB configuration, mapping.xml

DB configuration

New Configuration().addResource(“title.hbm.xml”)

New Configuration().configure(“hibernate.cfg.xml”)

Hibernate.hbm.xml

Detached objects—pojo will be detached from session or session is closed. So changes in obj is not reflected in DB

Can be formed by- either

Session.close()

Session.evict(user)

Persisting detached objects needs to be connected to new session to regain their persistence property

Session factory: immutable and threadsafe- one for each DB

Commit is alternative of transaction commit

ACID

Atomicity- rollback()

Consistency- normal DB rules should be maintained(like foreign key must be present and all

Isolation- :The effects of the transaction will be completely invisible to all other transactions until it has completed successfully

Durability- he data should be retained intact