@Repository

@Transactional

**public** **class** RMDAOImpl **implements** iRMDAO {

**private** Log logger = LogFactory.*getLog*(**this**.getClass());

**protected** org.springframework.orm.hibernate3.HibernateTemplate

template = **null**;

@Resource(name = "ctcSessionFactory")

**protected** SessionFactory sessionFactory;

@Autowired

**public** **void** init(SessionFactory sessionFactory) {

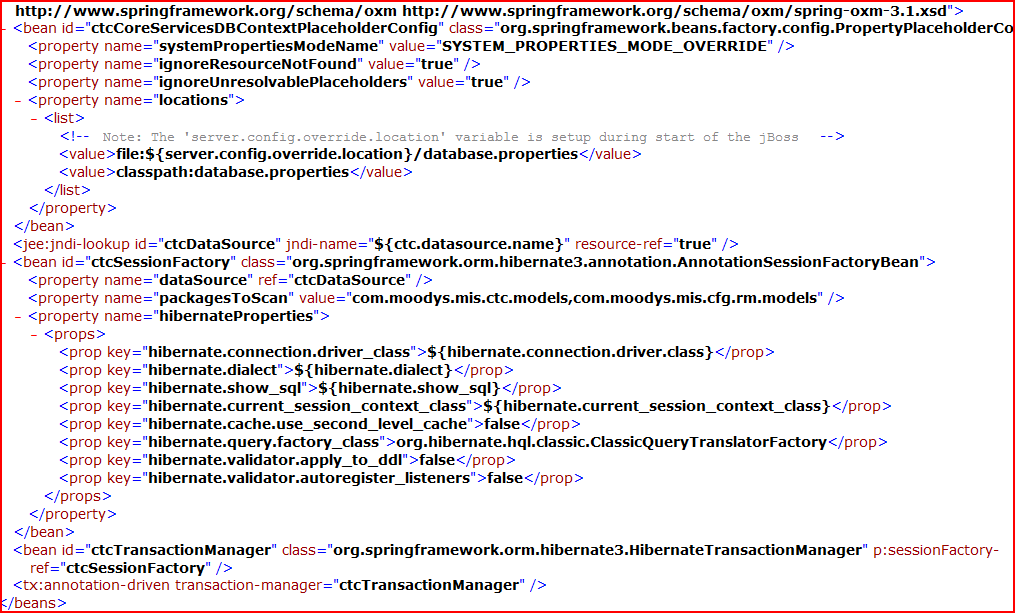
setSessionFactory(sessionFactory);

}

**public** **void** setSessionFactory(SessionFactory sessionFactory) {

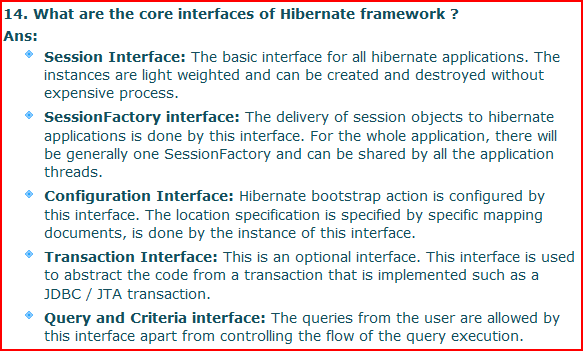
template = **new** org.springframework.orm.hibernate3.HibernateTemplate

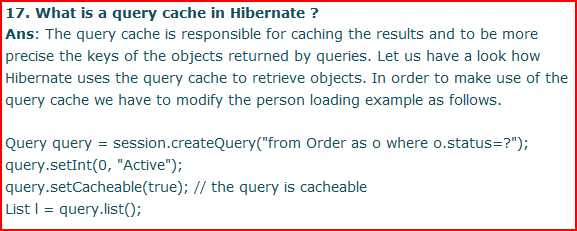
HibernateTemplate(sessionFactory);

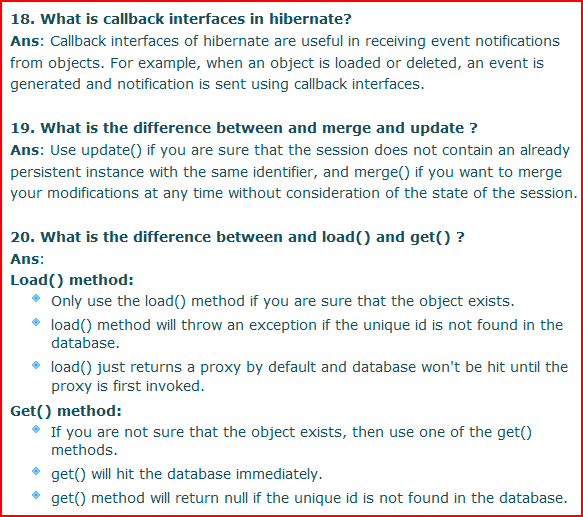
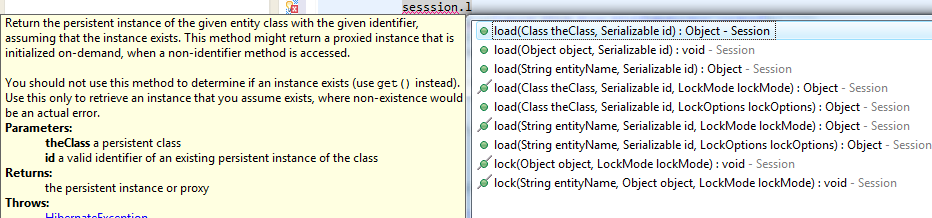


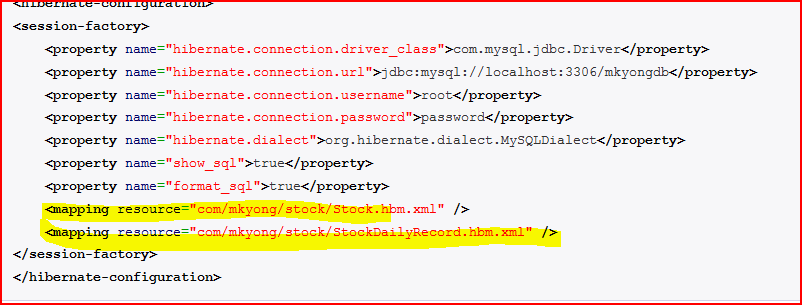
Use of dialect

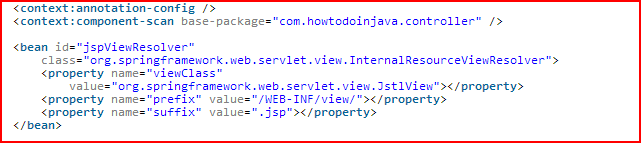
Databases implement subtle differences in the SQL they use. Things such as data types for example vary across databases (e.g. in Oracle I might put an integer value in a number field and in SQL Server use an int field). Or database specific functionality - selecting the top n rows is different depending on the database. The dialect abstracts this so you don't have to worry about it.

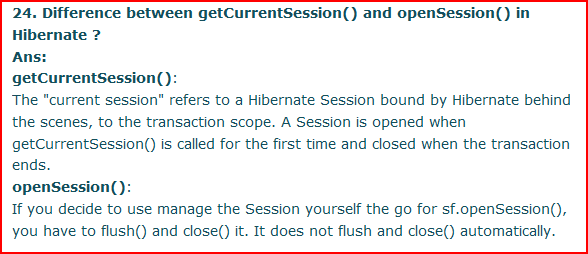


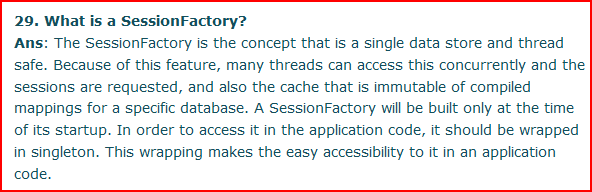


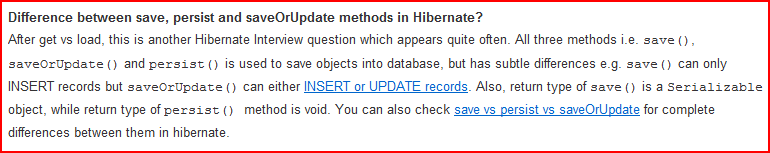












Thread questions:

It is never legal to start a thread more than once. In particular, a thread may not be restarted once it has completed execution.

Furthermore:

Throws:  
[IllegalThreadStateException](http://java.sun.com/javase/6/docs/api/java/lang/IllegalThreadStateException.html) - if the thread was already started.

So yes, a Thread can only be started once.

If so than what do I do if I want to run the thread again?

If a Thread needs to be run more than once, then one should make an new instance of the Thread and call start on it.