

JPA

— Problems with JDBC:

- More configuration
- Datatype mismatch between the model attributes and the table fields
- No encapsulation
- hard to implement MVC
- difficult to implement in large applications
- mismatch problems with
 - Granularity
 - Inheritance
 - Identity
 - Associations
 - Navigation

What is JPA?

— The Java Persistence API (JPA) is a Java specification for accessing, persisting, and managing data between Java objects / classes and a relational database.

- JPA is now considered the standard industry approach for Object to Relational Mapping (ORM) in the Java Industry.
- JPA is specification, and requires an implementation.
- JPA also defines a runtime EntityManager API for processing queries and transaction on the objects against the database.
- Caching also done at ORM layer
- Transaction management and automatic key generation
- Easy to switch over the tools(hibernate, abates, topplink etc)

— PersistenceContext

EntityManagerFactory
EntityManager

ORM (Object Relational Mapping)

Programming technique to convert java objects into relational database

Java Objects with data are mapped to the database

Class name - Table name

Class variables - table columns

NOTE:

Hibernate , iBatis, TopLink are the tools to implement the ORM applications where

the data is mapped between Java objects and the databases.

With the help of JPA, it is easy to switch between these tools if needed based on the requirements.

Because JPA is a specification where the app is built and the implementation can be any of the tools Hibernate or iBatis or TopLink.

— hibernate configuration properties

<https://docs.jboss.org/hibernate/stable/core.old/reference/en/html/configuration-optional.html>