Hibernate

Hibernate is an ORM(object – relational map) tool in Java. It is what Java’s JPA (Java Persistence API) is framed around.

Dialect agnostic data persistence

Maps Java classes to databases tables.

No more SQL. We use HQL (hibernate Query Language).

Can use Hibernate with an existing DB or use hibernate mappings to create tables & other DB entities.

Interfaces of Hibernate:

Session: connection with DB (do not confuse with HttpSession)

Expose methods used to persist data… save, get, delete, merge…

SessionFactory – used to create session objects. Only need 1 of these… will use the singleton design pattern.

Configuration – used to create SessionFactory for a DB -> configured via XML -> hibernate.cfg.xml

Transaction – manages ACID – compliant interactions with DB. i.e. tx.commit();

Query – write HQL(or native SQL) query.

Criteria – complicated queries programmatically.

Hibernate Object States: describe the state of an object in Java with relation to a row in DB.

Transient – no session associated with this object. Just instantiate using “new” keyword. No persisted representation in DB no PK value. Dereferenced transient objects are eligible for garbage collection.

Persistent – object has representation in DB (and has a unique identifier), in the scope of hibernate session. Hibernate will depict any changes made to an object in this state and will reflect changes at the end of the transaction.

Detached: object was once persisted, but session has been closed, still has representation in DB, but changes to this object will not be reflected in DB, unless it is reattached to a session.

Session methods: GET vs LOAD, MERGE vs UPDATE, SAVE vs PERSIST

Transactional write – behind.

Automatic dirty checking

Fetching strategies.