

Spring-Hibernate Training Program

Session 6

Hibernate Object States

- Following are the States of Objects in relation to Hibernate
 - **Transient** : *new* Object, not associated with Hibernate Session.
 - **Persistent** : Persistent, has Identifier, associated with an open Hibernate Session.
 - **Detached** : Persistent, has Identifier, NOT associated with an open Hibernate Session.

Making objects persistent

- `save()`
- `update()`
- `saveOrUpdate()`
- `merge()`
- `persist()`

Loading objects

- `load()` : unrecoverable exception if not found.
- `get()` : returns null if not found.
- `refresh()` : re-load the object and its collections.
- Fetching Strategies
 - Join Fetching : Single Select. Outer Join
 - Select Fetching : 2 select queries. Lazy.
 - Subselect Fetching : 1 select and 1 subselect query. Lazy.
 - Batch Fetching : limit the number of instances.

Modifying Objects

- Persistent Objects
 - `session.load()`
 - `session.flush()`
- Detached Objects
 - `session.update()` : not applicable in the loading session.
 - `session.merge()` : session state agnostic.
 - `session.lock()` : reassociate unmodified object.

Deleting Objects

- `delete()` - Delete from database. Application can still have a reference.

Cascading Operations

- create
- merge
- delete
- save-update
- evict
- replicate
- lock
- refresh

Querying Objects

- *org.hibernate.Query*
- `list()`
- `iterate()`
- tuples
- Scalar results
- Pagination. `setFirstResult()`, `setMaxResults()`

Read-only Entities

- Only Persistent entities can be made read-only.
- Entities of immutable class
- *Session.setReadOnly(true);*
- *Query.setReadOnly(true);*
- *Criteria.setReadOnly(true);*

HQL

- Hibernate uses a powerful query language (HQL) that is similar in appearance to SQL. Compared with SQL, however, HQL is fully object-oriented and understands notions like inheritance, polymorphism and association.
- Case Insensitive
- Simplest query is “*from Author*”

End of
Spring – Hibernate Training Program
Session 6