Hibernate Tip: Initialize lazy relationships

Question:

How can I initialize a lazy relationship within a query to avoid *LazyInitializationExceptions*?

Solution:

Hibernate throws a *LazyInitializationException* if you try to use the attribute of a lazily fetched relationship outside of an active Hibernate Session.

```
EntityManager em = emf.createEntityManager();
em.getTransaction().begin();

Author a = em.createQuery(
                    "SELECT a FROM Author a WHERE id = 1",
                    Author.class).getSingleResult();

em.getTransaction().commit();
em.close();

log.info(a.getFirstName() + " " + a.getLastName()
                + " wrote " + a.getBooks().size() + " books.");
```

You can avoid that by initializing the relationship before you close the session. Hibernate and JPA support different ways to do that. The easiest one is a *JOIN FETCH* statement within a query, like the one in the following code snippet.

```
EntityManager em = emf.createEntityManager();
em.getTransaction().begin();

Author a = em.createQuery("SELECT a FROM Author a”
                        +“ JOIN FETCH a.books WHERE a.id = 1",
                        Author.class).getSingleResult();

em.getTransaction().commit();
em.close();

log.info(a.getFirstName() + " " + a.getLastName()
                + " wrote " + a.getBooks().size() + " books.");
```

The additional *FETCH* keyword tells Hibernate to not only join the entity for the query but also to fetch it from the database to initialize the attribute. This prevents *LazyInitializationExceptions* if you access the relationship attribute outside of an active Hibernate Session.

Learn more:

Join Fetch statements are only 1 option to initialize lazy relationships. Other interesting options are @NamedEntityGraphs and dynamic entity graphs which allow you to define a query independent graph of entities which shall be fetched with a query.

Initializing a required lazy relationship does not only prevent *LazyInitializationExceptions*, it also can improve the performance by avoiding n+1 select issues. I get into more details about that in this free mini-course and in the Hibernate Performance Tuning Online Training.