

FAQ: How to load the courses at a later time in the application?

FAQ: How load the courses at a later time in the application?

Question

I've watched your 2 solutions for loading related data after session closing. Both, either getting related courses before closing session and using JOIN FETCH seem to be negating of lazy loading (using those solutions we completely resign of lazy loading).

Is there any good solution to load these data somewhere else in the app? Should I open new session?

Answer

Yes, you can load it later with using a new session, just make use of HQL

Here's the code snippet. Make note of HQL in bold

```
session = factory.getCurrentSession();

session.beginTransaction();

// get courses for a given instructor
Query<Course> query = session.createQuery("select c from
Course c "
                                     + "where
c.instructor.id=:theInstructorId",
                                     Course.class);

query.setParameter("theInstructorId", theId);
```

```
List<Course> tempCourses = query.getResultList();

System.out.println("tempCourses: " + tempCourses);
```

Here's the full example.

:-)

```
package com.luv2code.hibernate.demo;

import java.util.List;

import org.hibernate.Session;
import org.hibernate.SessionFactory;
import org.hibernate.cfg.Configuration;
import org.hibernate.query.Query;

import com.luv2code.hibernate.demo.entity.Course;
import com.luv2code.hibernate.demo.entity.Instructor;
import com.luv2code.hibernate.demo.entity.InstructorDetail;

public class GetCoursesLater {

    public static void main(String[] args) {

        // create session factory
        SessionFactory factory = new Configuration()
            .configure("hibernate.cfg.xml")
            .addAnnotatedClass(Instructor.class)
            .addAnnotatedClass(InstructorDetail.class)
            .addAnnotatedClass(Course.class)
            .buildSessionFactory();

        // create session
        Session session = factory.getCurrentSession();

        try {
```

```
// start a transaction
session.beginTransaction();

// get the instructor from db
int theId = 1;
Instructor templInstructor = session.get(Instructor.class,
theId);

System.out.println("luv2code: Instructor: " + templInstructor);

// commit transaction
session.getTransaction().commit();

// close the session
session.close();

System.out.println("\nluv2code: The session is now closed!\n");

//
// THIS HAPPENS SOMEWHERE ELSE / LATER IN THE
PROGRAM

// YOU NEED TO GET A NEW SESSION
//

System.out.println("\n\nluv2code: Opening a NEW session \n");

session = factory.getCurrentSession();

session.beginTransaction();

// get courses for a given instructor
Query<Course> query = session.createQuery("select c from
Course c "
                                     + "where
c.instructor.id=:theInstructorId",
                                     Course.class);

query.setParameter("theInstructorId", theId);
```

```
List<Course> tempCourses = query.getResultList();

System.out.println("tempCourses: " + tempCourses);

// now assign to instructor object in memory
tempInstructor.setCourses(tempCourses);

System.out.println("luv2code: Courses: " +
tempInstructor.getCourses());

session.getTransaction().commit();

System.out.println("luv2code: Done!");
}
finally {

    // add clean up code
    session.close();

    factory.close();
}
}
}
```