Implementation of Hibernate with Spring

Steps:

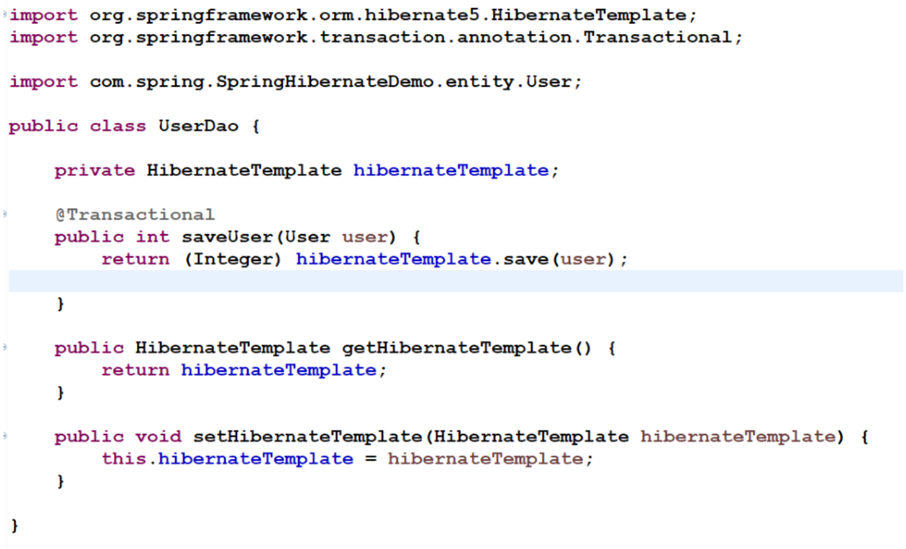
First:

1. Create POGO- class, with required annotations.

Second:

1. Create a Dao package and create a java class within it associated with POGO
   Eg: if pogo class is named Employee then Dao class will be EmployeeDao.
2. Define a HibernateTemplate attribute with setter and getter methods.
3. Implement method to perform CRUD operations.

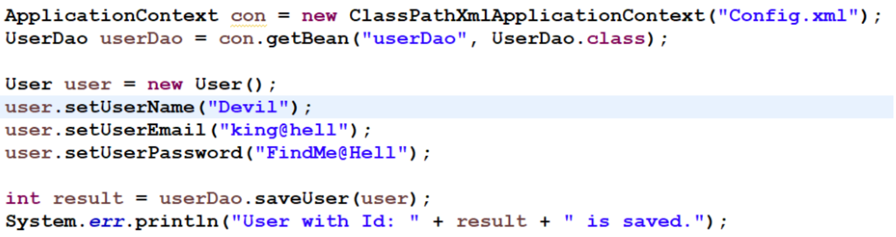
For instance:



```
import org.springframework.orm.hibernate5.HibernateTemplate;
import org.springframework.transaction.annotation.Transactional;

import com.spring.SpringHibernateDemo.entity.User;

public class UserDao {

    private HibernateTemplate hibernateTemplate;

    @Transactional
    public int saveUser(User user) {
        return (Integer) hibernateTemplate.save(user);

    }

    public HibernateTemplate getHibernateTemplate() {
        return hibernateTemplate;
    }

    public void setHibernateTemplate(HibernateTemplate hibernateTemplate) {
        this.hibernateTemplate = hibernateTemplate;
    }

}
```

Third:

1. Create a driver class with main method.



```
ApplicationContext con = new ClassPathXmlApplicationContext("Config.xml");
UserDao userDao = con.getBean("userDao", UserDao.class);

User user = new User();
user.setUserName("Devil");
user.setUserEmail("king@hell");
user.setUserPassword("FindMe@Hell");

int result = userDao.saveUser(user);
System.err.println("User with Id: " + result + " is saved.");
```

Forth:

1. Create a Configuration file. A xml file, and define the beans and enable required annotations.
2. Configure Dao bean, for instance, UserDao, which has a property hibernateTemplate. To set property of UserDao, you have to define a Hibernate Template beans, which has its own property such as sessionFactory.
3. Thus, you have to define a bean LocalSessionFactoryBean, and which has its properties, such as **first**: dataSource which takes dataSource object, **second**: HibernateProperties [where you will provide dialect, hbm2ddl.auto, etc] and **third**: annotatedClasses—which is a Entity class you have defined.
4. To provide the dataSource object, you have to define the bean DriverManagerDataSource which has it's own properties such as, driverClass, url, username, password. [set these properties according to your database]

Note:

To enable all the annotation, you have to import the schema

```
xmlns:tx="http://www.springframework.org/schema/tx"
http://www.springframework.org/schema/tx
http://www.springframework.org/schema/tx/spring-tx.xsd
```

And, use a tag:

```
<!-- Enabling all annotations -->
<tx:annotation-driven />
```

This will enable all the annotation used in the project.