Hibernate 与 spring 整合

<bean id=*"mySessionFactory"* class=*"org.springframework.orm.hibernate3.annotation.AnnotationSessionFactoryBean"*>

<property name=*"dataSource"* ref=*"dataSource"* />

<property name=*"packagesToScan"* value=*"com.ssh.po"*>

</property>

<property name=*"hibernateProperties"*>

<props>

<prop key=*"hibernate.dialect"*>org.hibernate.dialect.MySQL5Dialect</prop>

<prop key=*"hibernate.show\_sql"*>true</prop>

<prop key=*"hibernate.format\_sql"*>true</prop>

<prop key=*"hibernate.hbm2ddl.auto"*>update</prop>

<prop key=*"hibernate.current\_session\_context\_class"*>thread</prop>

</props>

</property>

</bean>

<bean id=*"txManager"* class=*"org.springframework.orm.hibernate3.HibernateTransactionManager"*>

<property name=*"sessionFactory"* ref=*"sessionFactory"* />

</bean>

1. 配置数据连接池

<bean id=*"dataSource"* class=*"org.apache.commons.dbcp.BasicDataSource"* destroy-method=*"close"*>

<property name=*"driverClassName"* value=*"com.mysql.jdbc.Driver"* />

<property name=*"url"* value=*"jdbc:mysql://localhost:3306/test"* />

<property name=*"username"* value=*"root"* />

<property name=*"password"* value=*"root"* /> </bean>

1. 测试：

**import** java.sql.Connection;

**import** java.sql.SQLException;

**import** org.apache.commons.dbcp.BasicDataSource;

**import** org.springframework.context.ApplicationContext;

**import** org.springframework.context.support.ClassPathXmlApplicationContext;

**public** **class** Test {

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stub

ApplicationContext ac = **new** ClassPathXmlApplicationContext("applicationContext.xml");

BasicDataSource bds = ac.getBean("dataSource",BasicDataSource.**class**);

Connection conn = **null**;

**try** {

conn = bds.getConnection();

conn.createStatement().executeUpdate("insert into t\_user1 values(null,'admin','12345')");

} **catch** (SQLException e) {

// **TODO** Auto-generated catch block

e.printStackTrace();

}

}

}

2.

**public** **void** addUser(User user) {

// **TODO** Auto-generated method stub

ApplicationContext ac = **new** ClassPathXmlApplicationContext("applicationContext.xml");

SessionFactory sessionFactory = ac.getBean("mySessionFactory",SessionFactory.**class**);

Session session = sessionFactory.openSession();

Transaction ts = session.beginTransaction();

**try**{

Query query = session.createQuery("from User where username=?");

query.setParameter(0, "admin");

User user1 = (User) query.uniqueResult();

System.*out*.println(user1.getPassword());

}**catch** (Exception e) {

// **TODO**: handle exception

ts.rollback();

}

}