# Java Hibernate配置

1. 新建maven项目, 在pom.xml下添加如下

<properties>

<hibernate.version>4.3.5.Final</hibernate.version>

</properties>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-orm</artifactId>

<version>${org.springframework.version}</version>

</dependency>

<!-- Hibernate4 -->

<dependency>

<groupId>org.hibernate</groupId>

<artifactId>hibernate-core</artifactId>

<version>${hibernate.version}</version>

</dependency>

<!-- for JPA, use hibernate-entitymanager instead of hibernate-core -->

<dependency>

<groupId>org.hibernate</groupId>

<artifactId>hibernate-entitymanager</artifactId>

<version>${hibernate.version}</version>

</dependency>

<!-- 为了让Hibernate使用代理模式，需要javassist -->

<dependency>

<groupId>org.javassist</groupId>

<artifactId>javassist</artifactId>

<version>3.18.1-GA</version>

</dependency>

<dependency>

<groupId>org.jboss.logging</groupId>

<artifactId>jboss-logging</artifactId>

<version>3.1.3.GA</version>

</dependency>

<dependency>

<groupId>org.jboss.spec.javax.annotation</groupId>

<artifactId>jboss-annotations-api\_1.2\_spec</artifactId>

<version>1.0.0.Final</version>

</dependency>

1. 在resources资源文件夹下添加beans包，添加配置文件Hibernate-config.xml（spring bean文件）， 内容如下

<import resource="classpath\*:/common-beans.xml" />

<bean id="sessionFactory"

class="org.springframework.orm.hibernate4.LocalSessionFactoryBean">

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

<property name="packagesToScan">

<list>

<!-- 可以加多个包, 实体所在包名 -->

<value>entities</value>

</list>

</property>

<property name="hibernateProperties">

<props>

<prop key="hibernate.hbm2ddl.auto">${hibernate.hbm2ddl.auto}</prop>

<prop key="hibernate.dialect">${hibernate.dialect}</prop>

<prop key="hibernate.show\_sql">${hibernate.show\_sql}</prop>

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

</props>

</property>

</bean>

<!-- 配置Hibernate事务管理器 -->

<bean id="transactionManager"

class="org.springframework.orm.hibernate4.HibernateTransactionManager">

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

</bean>

<!-- 声明式容器事务管理 ,transaction-manager指定事务管理器为transactionManager -->

<tx:advice id="txAdvice" transaction-manager="transactionManager">

<tx:attributes>

<tx:method name="add\*" propagation="REQUIRED" />

<tx:method name="get\*" propagation="REQUIRED" />

<tx:method name="\*" read-only="true" />

</tx:attributes>

</tx:advice>

<!-- 配置事务异常封装 -->

<bean id="persistenceExceptionTranslationPostProcessor"

class="org.springframework.dao.annotation.PersistenceExceptionTranslationPostProcessor" />

<aop:config expose-proxy="true">

<!-- 只对业务逻辑层实施事务 -->

<!--服务命名空间，如在java源文件包下的services包下的java文件（多级用.隔开） -->

<aop:pointcut id="txPointcut" expression="execution(\* services..\*.\*(..))" />

<!-- Advisor定义，切入点和通知分别为txPointcut、txAdvice -->

<aop:advisor pointcut-ref="txPointcut" advice-ref="txAdvice" />

</aop:config>

1. 添加hibernate服务操作注入配置， resources下的beans下Hibernate-beans.xml（spring bean）, 内容如下

<import resource="classpath\*:/beans/Hibernate-config.xml" />

<!-- 业务服务类, 构造bean -->

<bean id="userInfoDao" class="daos.UserInfoDao">

<property name="sessionFactory">

<ref bean="sessionFactory"></ref>

</property>

</bean>

<bean id="userInfoService" class="services.UserInfoService">

<property name="userInfoDao">

<ref bean="userInfoDao"></ref>

</property>

</bean>

1. 添加entity

package entity;

import javax.persistence.Column;

import javax.persistence.Entity;

import javax.persistence.GeneratedValue;

import javax.persistence.GenerationType;

import javax.persistence.Id;

@SuppressWarnings("deprecation")

/\* 数据库表名 \*/

@Entity(name="users")

public class MyUser {

public MyUser(){

super();

}

@Id

@GeneratedValue(strategy=GenerationType.AUTO)

@Column(name="id")

private Integer id;

@Column(name="user\_name",length=32)

private String user\_name;

@Column(name="age")

private Integer age;

@Column(name="nice\_name",length=32)

private String nice\_name;

public Integer getId() {

return id;

}

public void setId(Integer id) {

this.id = id;

}

public String getUser\_name() {

return user\_name;

}

public void setUser\_name(String user\_name) {

this.user\_name = user\_name;

}

public Integer getAge() {

return age;

}

public void setAge(Integer age) {

this.age = age;

}

public String getNice\_name() {

return nice\_name;

}

public void setNice\_name(String nice\_name) {

this.nice\_name = nice\_name;

}

}

1. 添加dao

package dao;

import java.util.List;

import org.hibernate.Query;

import org.hibernate.Session;

import org.hibernate.SessionFactory;

import entity.MyUser;

public class UserDao {

private SessionFactory sessionFactory;

public void setSessionFactory(SessionFactory sessionFactory) {

this.sessionFactory = sessionFactory;

}

public SessionFactory getSessionFactory() {

return sessionFactory;

}

public List<MyUser> getAllUser(){

System.out.println("123");

/\* 数据库表名 \*/

String hsql="from users";

Session session = sessionFactory.getCurrentSession();

Query query = session.createQuery(hsql);

return query.list();

}

}

1. 增加service

package service;

import dao.\*;

public class UserService {

private UserDao userDao;

public int userCount(){

return userDao.getAllUser().size();

}

public UserDao getUserDao() {

return userDao;

}

public void setUserDao(UserDao userDao) {

this.userDao = userDao;

}

}

1. 在调用的地方注入服务, 如通过属性注入

@Resource(name="userService")

private UserService myservice;

在方法中调用服务

int c = myservice.userCount();

# 其它配置