Hibernate 框架的搭建

1. 在开发环境中导入相关的hibernate包和连接池包和数据驱动包



1. 在根路径下配置基本配置文件hibernate.cfg.xml

<?xml version='1.0' encoding='UTF-8'?>

<!DOCTYPE hibernate-configuration PUBLIC

"-//Hibernate/Hibernate Configuration DTD 3.0//EN"

"http://hibernate.sourceforge.net/hibernate-configuration-3.0.dtd">

<!-- Generated by MyEclipse Hibernate Tools. -->

<hibernate-configuration>

<session-factory>

<!--Oracle 配置 -->

<property name="dialect">

org.hibernate.dialect.Oracle9iDialect

</property>

<property name="connection.url">

jdbc:oracle:thin:@192.168.7.\*\*\*:1521:DatabaseName

</property>

<property name="connection.username">OracleName</property>

<property name="connection.password">OraclePassword</property>

<property name="connection.driver\_class">

oracle.jdbc.driver.OracleDriver

</property>

<!-- mysql 数据配置-->

<!--<property name="dialect">

org.hibernate.dialect.MySQLDialect

</property>

<property name="connection.url">

jdbc:mysql://127.0.0.1:3306/DatabaseName

</property>

<property name="connection.username">mysqlName</property>

<property name="connection.password">mysqlPassword</property>

<property name="connection.driver\_class">

com.mysql.jdbc.Driver

</property>-->

<!-- 在后台打印sql语句 -->

<property name="show\_sql">true</property>

<!-- hibernate自动建表 -->

<property name="hibernate.hbm2ddl.auto">update</property>

<!-- 使用c3p0连接池 -->

<property name="hibernate.connection.provider\_class">org.hibernate.connection.C3P0ConnectionProvider</property>

<!--初始化时获取10个连接，取值应在minPoolSize与maxPoolSize之间。Default: 3 -->

<property name="c3p0.initialPoolSize">10</property>

<!-- 连接池中JDBC连接的最大数量。Hibernate默认为100 -->

<property name="hibernate.c3p0.max\_size">100</property>

<!-- 连接池中JDBC连接的最小数量。Hibernate默认为1 -->

<property name="hibernate.c3p0.min\_size">5</property>

<!-- 何时从连接池中移除一个空闲的连接（以秒为单位）时。Hibernate默认为0，永不过期 -->

<property name="hibernate.c3p0.timeout">300</property>

<!-- 被缓存的预编译语句数量。用来提高性能。Hibernate默认为0，缓存不可用-->

<property name="hibernate.c3p0.max\_statements">1000</property>

<!-- 若数据库中连接不足时，一次向数据库服务器申请多少个连接-->

<property name="hibernate.c3p0.acquire\_increment">5</property>

<!--hibernate将session 的线程配置 -->

<property name="current\_session\_context\_class">thread</property>

<!--连接关闭时默认将所有未提交的操作回滚。Default: false -->

<property name="autoCommitOnClose">false</property>

<!--两种指定映射文件-->

<!-- 注解方式-->

<!--<mapping class="com.eakom.\*"/> -->

<!-- xxx.hbm.xml配置文件方式 -->

<!--<mapping resource="org/mxg/UserInfo.hbm.xml"> -->

</session-factory>

</hibernate-configuration>

1. 在跟项目的新建工具类

**package** com.eakom.utils;

**import** java.io.Serializable;

**import** org.hibernate.Session;

**import** org.hibernate.SessionFactory;

**import** org.hibernate.Transaction;

**import** org.hibernate.cfg.AnnotationConfiguration;

**public** **class** HibernateUtils {

**private** **static** SessionFactory *sessionFactory*;

**private** HibernateUtils() {

}

**static** {

/\*\*

\* configure()参数为空默认查找classes目录下hibernate.cfg.xml

\* configure("文件名")也有重载方法，参数名为配置文件名

\*/

*sessionFactory* = **new** AnnotationConfiguration().configure().buildSessionFactory();

}

**public** **static** SessionFactory getSessionFactory() {

**return** *sessionFactory*;

}

/\*\*

\* 如果想使用sessionFactory.getCurrentSession()来获得Session时，需要在配置文件中添加一句：

\* <!-- 本地事务 防止使用sessionFactory.getCurrentSession()时报错："org.hibernate.HibernateException: No CurrentSessionContext configured!"-->

\* <property name="hibernate.current\_session\_context\_class">thread</property>

\* **@return**

\*/

**public** **static** Session getCurrentSession() {

**return** *sessionFactory*.getCurrentSession();

}

**public** **static** Session getNewSession() {

**return** *sessionFactory*.openSession();

}

**public** **static** **void** add(Object entity) {

Session s = **null**;

Transaction tx = **null**;

**try** {

s = HibernateUtils.*getNewSession*();

tx = s.beginTransaction();

s.save(entity);

tx.commit();

} **finally** {

**if** (s != **null**)

s.close();

}

}

**public** **static** **void** update(Object entity) {

Session s = **null**;

Transaction tx = **null**;

**try** {

s = HibernateUtils.*getNewSession*();

tx = s.beginTransaction();

s.update(entity);

tx.commit();

} **finally** {

**if** (s != **null**)

s.close();

}

}

**public** **static** **void** delete(Object entity) {

Session s = **null**;

Transaction tx = **null**;

**try** {

s = HibernateUtils.*getNewSession*();

tx = s.beginTransaction();

s.delete(entity);

tx.commit();

} **finally** {

**if** (s != **null**)

s.close();

}

}

**public** **static** Object get(Class clazz, Serializable id) {

Session s = **null**;

**try** {

s = HibernateUtils.*getNewSession*();

Object obj = s.get(clazz, id);

**return** obj;

} **finally** {

**if** (s != **null**)

s.close();

}

}

}

1. 在项目中使用hibernate连接

Session session = HibernateUtils.*getCurrentSession*();

Transaction tx = session.beginTransaction();

String sql=”select \* from tablename”;

SQLQuery query=session.createSQLQuery(sql);

Query.list();

tx.commit();

1. 创建完毕