简单的Hibernate实例入门

# Hibernate简介

hibernate是一个开源的对象/关系映射（ORM）框架，它对JDBC进行了轻量级的封装。所谓ORM就是Object/Relationship Mapping，为什么要使用对象/关系映射呢？因为使用面向对象思想编写的数据库应用程序都是要把信息存储到关系型数据库中的，使用Hibernate就可以使用面向对象的方法来操作和使用数据库了。

# Hibernate简单实例

下面通过一个简单的例子来介绍一下使用Hibernate的基本步骤，本实例使用了MySQL数据库，并使用JUnit来进行测试。首先要引入使用的jar包，在本例中需要引入Hibernate使用的相关jar包、MySQL数据库连接jar包和JUnit测试jar包，在pom文件中添加以下依赖。

|  |
| --- |
| <dependencies>  <dependency>  <groupId>org.hibernate</groupId>  <artifactId>hibernate-core</artifactId>  <version>5.2.0.Final</version>  </dependency>  <dependency>  <groupId>mysql</groupId>  <artifactId>mysql-connector-java</artifactId>  <version>5.1.38</version>  </dependency>  <dependency>  <groupId>junit</groupId>  <artifactId>junit</artifactId>  <version>4.10</version>  <scope>test</scope>  </dependency>  </dependencies> |

引入了相关jar包之后就可以进行编码了。编写一个Hibernate例子分为以下几个步骤：

1. 创建Hibernate的配置文件（hibernate.cfg.xml）；
2. 创建持久化类；
3. 创建对象/关系映射文件；
4. 通过Hibernate的API来编写操作数据库的代码。

## 创建Hibernate配置文件（hibernate.cfg.xml）

首先我们需要创建Hibernate的配置文件hibernate.cfg.xml，配置文件的内容如下：

|  |
| --- |
| <?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">  <hibernate-configuration>  <session-factory>  <!-- 对应的数据库驱动 -->  <property name="hibernate.connection.driver\_class">com.mysql.jdbc.Driver</property>  <!-- 数据库密码 -->  <property name="hibernate.connection.password">123</property>  <!-- 数据库用户名 -->  <property name="hibernate.connection.username">root</property>  <!-- 数据库url，类似于JDBC中连接数据库的url -->  <property name="hibernate.connection.url">jdbc:mysql://172.16.129.212:3306/test?useUnicode=true&amp;characterEncoding=UTF-8</property>  <!-- 数据库使用的方言 -->  <property name="hibernate.dialect">org.hibernate.dialect.MySQLDialect</property>    <!-- 是否把输出到控制台的sql进行格式化 -->  <property name="format\_sql">true</property>  <!-- 是否将Hibernate运行时的sql输出到控制台 -->  <property name="show\_sql">true</property>  <!-- 自动创建|更新|验证数据库表结构 -->  <property name="hbm2ddl.auto">create</property>    <mapping resource="User.hbm.xml"/>  </session-factory>  </hibernate-configuration> |

## 创建持久化类

接下来我们就需要创建一个持久化类User，里面包含了一些基本的用户信息。

|  |
| --- |
| **public** **class** User {  **private** **int** id;  **private** String name;  **private** **int** age;  **private** String sex;  **private** Date birthday;  **private** String address;  //无参数构造器，set ,get ,toString方法... |

## 创建对象/关系映射文件

接下来我们就需要根据持久化类创建一个对象/关系映射文件User.hbm.xml。

|  |
| --- |
| <?xml version=*"1.0"*?>  <!DOCTYPE hibernate-mapping PUBLIC "-//Hibernate/Hibernate Mapping DTD 3.0//EN"  "http://hibernate.sourceforge.net/hibernate-mapping-3.0.dtd">  **<hibernate-mapping package=*"com.immoc.entity"*>**  <class name=*"User"* table=*"USER"*>  <id name=*"id"* type=*"int"*>  <column name=*"ID"* />  <generator class=*"assigned"* />  </id>  <property name=*"name"* type=*"java.lang.String"*>  <column name=*"NAME"* />  </property>  <property name=*"age"* type=*"int"*>  <column name=*"AGE"* />  </property>  <property name=*"sex"* type=*"java.lang.String"*>  <column name=*"SEX"* />  </property>  <property name=*"birthday"* type=*"java.util.Date"*>  <column name=*"BIRTHDAY"* />  </property>  <property name=*"address"* type=*"java.lang.String"*>  <column name=*"ADDRESS"* />  </property>    </class>  </hibernate-mapping> |

这个对象/关系映射文件其实就是将定义的持久化类与数据库表进行映射，这里将User类映射为一个名为USER的表，将对象的属性映射为数据库的列，将属性的类型映射为数据库列的类型。在创建了这个对象/关系映射文件之后，我们需要在Hibernate配置文件hibernate.cfg.xml中声明这个映射关系，在session-factory标签下声明如下内容。

<mapping resource="com/imooc/map/User.hbm.xml"/>

## 通过Hibernate的API来编写操作数据库的代码

在上面一系列的配置完成之后我们就可以使用JUnit来编写一个测试类UserTest来使用Hibernate了。

|  |
| --- |
| **package** com.immoc.test;  **import** java.util.Date;  **import** org.hibernate.Session;  **import** org.hibernate.SessionFactory;  **import** org.hibernate.Transaction;  **import** org.hibernate.cfg.Configuration;  **import** org.junit.After;  **import** org.junit.Before;  **import** org.junit.Test;  **import** com.immoc.entity.User;  **public** **class** UserTest {  **private** SessionFactory sessionFactory;  **private** Session session;  **private** Transaction transaction;  @Before  **public** **void** init() {  //创建配置对象  Configuration config = **new** Configuration().configure();  //创建会话工厂对象  sessionFactory = config.buildSessionFactory();  //创建会话对象  session = sessionFactory.openSession();  //开启事务  transaction = session.beginTransaction();  }  @After  **public** **void** destory() {  //提交事务  transaction.commit();  //关闭会话  session.close();  //关闭会话工厂  sessionFactory.close();  }  @Test  **public** **void** saveNewUser() **throws** Exception {  User s = **new** User(1, "张三", 12, "男", **new** Date(), "上海市");  session.save(s);  }  } |

# Hibernate的执行流程

