

# 1、创建pojo实体类User

实体类对应数据库中的记录 uid 为其主键

```
1 package daiwei.learning.hibernate.pojo;
2
3 public class User {
4
5     private int uid;
6     private String username;
7     private String password;
8     private String address;
9
10    public int getUid() {
11        return uid;
12    }
13    public void setUid(int uid) {
14        this.uid = uid;
15    }
16    public String getUsername() {
17        return username;
18    }
19    public void setUsername(String username) {
20        this.username = username;
21    }
22    public String getPassword() {
23        return password;
24    }
25    public void setPassword(String password) {
26        this.password = password;
27    }
28    public String getAddress() {
29        return address;
30    }
31    public void setAddress(String address) {
32        this.address = address;
33    }
34
35 }
```

## 2、创建hibernate的映射文件hibernate.hbm.xml

一般情况下，hibernate的映射文件和 pojo 放在一个目录下,且名字固定为hibernate.hbm.xml

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <!DOCTYPE hibernate-mapping PUBLIC
3     "-//Hibernate/Hibernate Mapping DTD 3.0//EN"
4     "http://www.hibernate.org/dtd/hibernate-mapping-3.0.dtd">
5 <hibernate-mapping>
6     <!-- 配置类和表对应
7         class标签
8         name: 实体类全路径
9         table: 数据库中表名称
10    -->
11    <class name="daiwei.learning.hibernate.pojo.User" table="tb_user">
12        <!-- 配置实体类id和表id对应
13            hibernate要求实体类有一个属性唯一值
14            hibernate要求表字段作为唯一值
15        -->
16
17        <!-- id标签
18            name: pojo中id属性名
19            column: 生成的表字段名称
20            generator: 设置数据库id字段增长策略
21        -->
22        <id name="uid" column="id">
23            <generator class="native"></generator>
24        </id>
25        <!--
26            property标签
27            name: 实体类中属性名;
28            column: 生成的对应的表字段;
29        -->
30        <property name="username" column="username"></property>
31        <property name="password" column="password"></property>
32        <property name="address" column="address"></property>
33    </class>
34 </hibernate-mapping>
```

在写配置文件之前要导入，mapping 的 dtd 约束文件。[hibernate-mapping](#)（附件中有约束文件）

class标签

name: 实体类的全路径;

table: 数据库中的对应的表;

id 标签: 表主键

name: 实体类中属性对应主键字段;

column: 数据库中实体类主键字段;

generator 标签: 数据库中主键增长策略 (class=native 自增);

property 标签: 实体类中属性标签

name: 实体类中属性名

column: 数据库中属性名对应字段

### 3、hibernate核心配置文件 hibernate.cfg.xml

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <!DOCTYPE hibernate-configuration PUBLIC
3     "-//Hibernate/Hibernate Configuration DTD 3.0//EN"
4     "http://www.hibernate.org/dtd/hibernate-configuration-3.0.dtd">
5 <hibernate-configuration>
6     <session-factory>
7         <!-- 配置数据库信息，必须的-->
8         <property
9             name="hibernate.connection.driver_class">com.mysql.jdbc.Driver</property>
10        <property
11            name="hibernate.connection.url">jdbc:mysql://localhost:3306/hibernate_test</pr
12            operty>
13        <property name="hibernate.connection.username">root</property>
14        <property name="hibernate.connection.password">daiwei</property>
15
16        <!-- 配置hibernate有关配置，可选的 -->
17        <!-- 控制台输出发的sql语句 -->
18        <property name="hibernate.show_sql">true</property>
19        <!-- 输出的sql语句格式化 -->
20        <property name="hibernate.format_sql">true</property>
21        <!-- hibernate帮助建表
22            update: 如果已经创建了表，更新，如果没有创建，创建
23            -->
24        <property name="hibernate.hbm2ddl.auto">update</property>
25        <!-- 配置数据库方言 -->
26        <property
27            name="hibernate.dialect">org.hibernate.dialect.MySQLDialect</property>
28
29        <!-- 映射配置，必须的 -->
30        <mapping resource="daiwei/learning/hibernate/pojo/User.hbm.xml"/>
```

```
27
28     </session-factory>
29 </hibernate-configuration>
```

写配置之前要先引入 dtd 约束文件 `hibernate-configuration`（附件中有约束文件）

`property` 标签：配置属性文件

name：属性名

在<property>要配置的选项</property>

- 1、`hibernate.show_sql` 控制台是否展示底层发送的sql
- 2、`hibernate.format_sql` 控制展示的sql语句是否要格式化
- 3、`hibernate.hbm2ddl.auto` 是否需要hibernate帮助建表
- 4、`hibernate.dialect` 配置数据库方言

`mapping` 标签：链接映射文件

resource：映射文件位置

核心配置文件结构：

- 1、数据库配置部分（必须的）
- 2、核心配置部分（可选的）
- 3、映射配置部分（必须的）

hibernate的配置属性可以参考附件中的`hibernate.property`这个property文件，里面有hibernate.cfg.xml 的每个属性的配置选项。

## 4、测试Demo（不规范）

```
1 package daiwei.learning.hibernate.test;
2
3 import org.hibernate.Session;
4 import org.hibernate.SessionFactory;
5 import org.hibernate.Transaction;
6 import org.hibernate.cfg.Configuration;
7 import org.junit.Test;
8
9 import daiwei.learning.hibernate.pojo.User;
10
11 /**
12  * hibernate测试类
```

```

13  *<p>Title:HibernateTest</p>
14  *<p>description:</p>
15  * @author DW
16  * @date 2017年7月27日 下午1:47:45
17  * @version 1.0
18  */
19  public class HibernateTest {
20
21      @Test
22      public void test01() {
23          // 加载hibernate的核心配置文件
24          Configuration config = new Configuration();
25          config.configure();
26
27          // 创建sessionFactory
28          SessionFactory factory = config.buildSessionFactory();
29          Session session = factory.openSession();
30
31          //创建事务
32          Transaction transaction = session.beginTransaction();
33
34          //curd操作
35          User user = new User();
36          user.setUsername("daiwei");
37          user.setPassword("123456");
38          user.setAddress("nanchang");
39
40          session.save(user);
41          //提交事务
42          transaction.commit();
43
44          //关闭session
45          session.close();
46          factory.close();
47      }
48
49  }

```

测试步骤：

- 1、创建Configuration，加载 hibernate 核心配置文件
- 2、通过 Configuration 来创建SessionFactory接口
- 3、用 SessionFactory open 一个 session
- 4、使用 session 来开启事务（不是必须的）
- 5、执行数据库 curd 操作

6、提交事务

7、释放资源（关闭session, sessionFactory）