链滴

Hibernate 第一天

作者：xiaoqinghong
原文链接：https://ld246.com/article/1552839867793
来源网站：链滴

Java EE三层结构

1. web层：struts2
2. service层：spring
3. dao层：hibernate

- 对数据库进行crud操作

Hibernate配置文件

- **映射配置文件**

1. 映射文件的名称和位置没有固定要求
2. class标签name属性值实体类全路径
3. id标签和property标签name属性值 实体类名称
4. id标签和property标签，column属性可以省略
5. property标签type属性，设置生成表字段类型，自动对应类型（可以不做设置）

```
//实体类User.java
public class User {
    private int uid;
    private String username;
    private String password;
    private String address;
    public int getUid() {
        return uid;
    }
    public void setUid(int uid) {
        this.uid = uid;
    }
    public String getUsername() {
        return username;
    }
    public void setUsername(String username) {
        this.username = username;
    }
    public String getPassword() {
        return password;
    }
    public void setPassword(String password) {
        this.password = password;
    }
    public String getAddress() {
        return address;
    }
    public void setAddress(String address) {
        this.address = address;
    }
}
```

```
//映射文件，User.hbm.xml
<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE hibernate-mapping PUBLIC
   "-//Hibernate/Hibernate Mapping DTD 3.0//EN"
   "http://www.hibernate.org/dtd/hibernate-mapping-3.0.dtd">

<hibernate-mapping>
   <class name="com.eric.demo.entity.User" table="t_user">
      <id name="uid" column="uid">
      <generator class="native"></generator>
      </id>
      <property name="username" column="username"></property>
      <property name="password" column="password"></property>
      <property name="address" column="address"></property>
   </class>
</hibernate-mapping>
```

● 核心配置文件

1. 数据库部分必须配置
2. hibernate部分可选
3. 映射文件必须配置
4. 核心配置文件名称和位置固定（hibernate.cfg.xml、位置在src下）
5. 该文件配置完成后，很少修改

```
<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE hibernate-configuration PUBLIC
   "-//Hibernate/Hibernate Configuration DTD 3.0//EN"
   "http://www.hibernate.org/dtd/hibernate-configuration-3.0.dtd">

<hibernate-configuration>
   <session-factory>
      <!-- 第一部分，配置数据库信息(必须) -->
      <property name="hibernate.connection.driver_class">com.mysql.jdbc.Driver</property>
      <property name="hibernate.connection.url">jdbc:mysql:///hibernate_test</property>
      <property name="hibernate.connection.username">root</property>
      <property name="hibernate.connection.password">0314</property>
      <!-- 第二部分，配置hibernate信息(可选) -->
      <property name="hibernate.show_sql">true</property>
      <property name="hibernate.fromat_sql">true</property>
      <!-- 自动生成表 -->
      <property name="hibernate.hbm2ddl.auto">update</property>
      <!-- 配置数据库方言
         让hibernate框架识别不同的数据库所持有的语句
      -->
      <property name="hibernate.dialect">org.hibernate.dialect.MySQLDialect</property>
      <!-- 第三部分，把映射文件放到核心文件中(必须) -->
      <mapping resource="com/eric/demo/entity/User.hbm.xml"/>
```

```
    </session-factory>
</hibernate-configuration>
```

Hibernate核心api的使用

- **Configuration**

1. 到src下找到名称为hibernate.cfg.xml配置文件，创建对象，把配置文件放到对象里面（加载核心置文件）

```
Configuration cfg = new Configuration();
cfg.configure();
```

- **SessionFactory（重点）**

1. 使用Configuration对象创建SessionFactory对象
    1. 根据核心配置文件中，有数据库配置，有映射文件部分，到数据库里面根据映射关系把表创建。

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

2. 创建SessionFactory过程中，这个过程需要创建表，这个过程很耗时
    1. 一个项目只创建一个SessionFactory对象。
3. 具体优化实现
    1. 写工具类，写静态代码块实现

```
public class HibernateUtils {
    private static final Configuration cfg;
    private static final SessionFactory sessionFactory;
    static{
        // 加载核心配置文件
        cfg = new Configuration();
        cfg.configure();
        sessionFactory = cfg.buildSessionFactory();
    }
    // 获取SessionFactory的引用
    public static SessionFactory getSessionFactory() {
        return sessionFactory;
    }
}
```

- **Session（重点）**

1. 类似于jdbc中的connection
2. 调用session中的不同方法实现crud操作
    1. 添加save
    2. 修改update
    3. 删除delete

4. 根据id查询get

3. session对象是单线程对象：session不能共用，只能自己使用

- **Transaction**

1. 开启事务

```
Transaction tx = session.beginTransaction();
```

2. 事务的提交tx.commit()
3. 事务的回滚tx.rollback()
4. 事务的四个特性
    1. 原子性：一组操作，一个失败相当于所有失败
    2. 一致性：操作之前之后，数据总量没变
    3. 隔离性：多个操作同时操作同一条记录，互不影响
    4. 持久性：提交数据到数据库保存

解决MyEclipse写配置文件没有提示的问题

1. 若是电脑可以上网，配置的约束可以有提示功能
2. 把约束文件引入到MyEclipse中