

Hibernate 数据关联 —— 1:1 关联

开发程序的架构：Struts + DAO + Hibernate

读者应该具备以下知识点：

- 熟练使用 Struts
- 了解 Struts 中动态 ActionForm 及验证框架
- Hibernate 基础入门
- Hibernate 数据关联技术
- MySQL 数据库
- Eclipse + Tomcat
- 分发 Action

Person → 身份证

开发分为两个步骤：

- 1、先开发 Hibernate 和 DAO 的后台程序
- 2、再去开发 Struts 及验证框架

功能：

实现对用户及身份证的管理

具备以下功能：

人员管理：

- 1、添加人员 → **如果输入了身份证号码，则要添加身份证信息**
- 2、修改人员信息 → 姓名和年龄
- 3、删除人员信息 → 删除人员的时候要同时将身份证数据删除掉
- 4、列表打印及检索 → 列出全部的人员

|- 详细列出：将人员及身份证的信息详细的列出

身份证管理：

- 1、可以为单独人员添加身份证信息
- 2、修改身份证信息
- 3、删除一个身份证信息
- 4、可以列出全部的身份证号，检索操作

数据库创建脚本

```
-- 删除表
DROP TABLE idcard ;
DROP TABLE person ;
-- 创建表
CREATE TABLE person
(
    pid VARCHAR(32) NOT NULL PRIMARY KEY ,
    name VARCHAR(20) NOT NULL ,
    password VARCHAR(20) NOT NULL ,
```

```

    age INT ,
    sex VARCHAR(2)
) ;

CREATE TABLE idcard
(
    pid VARCHAR(32) ,
    serial VARCHAR(18) ,
    expiry INT ,
    FOREIGN KEY (pid) REFERENCES person(pid) ON DELETE CASCADE
) ;
-- 事务提交
commit ;

```

Person.hbm.xml:

```

<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE hibernate-mapping PUBLIC "-//Hibernate/Hibernate Mapping DTD 3.0//EN"
"http://hibernate.sourceforge.net/hibernate-mapping-3.0.dtd">
<!--
    Mapping file autogenerated by MyEclipse - Hibernate Tools
-->
<hibernate-mapping>
    <class name="org.lxx.vo.Person" table="person">
        <id name="pid" type="java.lang.String">
            <column name="pid" length="32" />
            <generator class="uuid.hex"></generator>
        </id>
        <property name="name" type="java.lang.String">
            <column name="name" length="20" not-null="true" />
        </property>
        <property name="password" type="java.lang.String">
            <column name="password" length="20" not-null="true" />
        </property>
        <property name="age" type="java.lang.Integer">
            <column name="age" />
        </property>
        <property name="sex" type="java.lang.String">
            <column name="sex" length="2" />
        </property>
        <one-to-one name="idCard" class="org.lxx.vo.IDCard"
            cascade="all">
        </one-to-one>
    </class>

```

```
</hibernate-mapping>
```

idcard.hbm.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE hibernate-mapping PUBLIC "-//Hibernate/Hibernate Mapping DTD 3.0//EN"
"http://hibernate.sourceforge.net/hibernate-mapping-3.0.dtd">
<!--
    Mapping file autogenerated by MyEclipse - Hibernate Tools
-->
<hibernate-mapping>
    <class name="org.lxh.vo.IDCard" table="idcard">
        <id name="pid" column="pid" type="java.lang.String">
            <generator class="foreign">
                <param name="property">person</param>
            </generator>
        </id>
        <property name="serial" type="java.lang.String">
            <column name="serial" length="18"></column>
        </property>
        <property name="expiry" type="java.lang.Integer">
            <column name="expiry"></column>
        </property>
        <one-to-one name="person" class="org.lxh.vo.Person"
            constrained="true">
        </one-to-one>
    </class>
</hibernate-mapping>
```

编写 DAO 接口

属性文件内容:

```
# Resources for parameter 'org.lxh.struts.ApplicationResources'
# Project 1VS1SHDemo01
# 不能为空!
errors.required={0}\u4e0d\u80fd\u4e3a\u7a7a\u5417
# 必须是数字!
errors.integer={0}\u5fc5\u987b\u662f\u6570\u5b57\u5417
# 用户编号
err.pid = \u7528\u6237\u7f16\u53f7
# 用户姓名
err.name = \u7528\u6237\u59d3\u540d
# 用户密码
err.password = \u7528\u6237\u5bc6\u7801
```

```
# 用户性别
err.sex = \u7528\u6237\u6027\u522b
# 年龄
err.age = \u5e74\u9f84
# 身份证号
err.serial = \u8eab\u4efd\u8bc1\u53f7
# 使用年限
err.expiry = \u4f7f\u7528\u5e74\u9650
```

struts-config.xml:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE struts-config PUBLIC "-//Apache Software Foundation//DTD Struts
Configuration 1.2//EN" "http://struts.apache.org/dtds/struts-config_1_2.dtd">

<struts-config>
  <data-sources />
  <form-beans>
    <form-bean name="personForm"
      type="org.apache.struts.validator.DynaValidatorForm">
      <form-property name="sex" type="java.lang.String" />
      <form-property name="password" type="java.lang.String" />
      <form-property name="age" type="java.lang.Integer" />
      <form-property name="pid" type="java.lang.String" />
      <form-property name="name" type="java.lang.String" />
    </form-bean>
    <form-bean name="idcardForm"
      type="org.apache.struts.validator.DynaValidatorForm">
      <form-property name="expiry" type="java.lang.Integer" />
      <form-property name="pid" type="java.lang.String" />
      <form-property name="serial" type="java.lang.String" />
    </form-bean>
  </form-beans>
  <global-exceptions />
  <global-forwards />
  <action-mappings>
    <action attribute="personForm" input="/errors.jsp"
      name="personForm" parameter="status" path="/person" scope="request"
      type="org.lxx.struts.action.PersonAction" validate="true">
    </action>
    <action attribute="idcardForm" input="/errors.jsp"
      name="idcardForm" parameter="status" path="/idcard" scope="request"
      type="org.lxx.struts.action.IdcardAction" validate="true">
    </action>
  </action-mappings>

```

```
</action-mappings>

<message-resources parameter="org.lhx.struts.ApplicationResources" />
<plug-in className="org.apache.struts.validator.ValidatorPlugIn">
    <set-property property="pathnames"
        value="/WEB-INF/validator-rules.xml,/WEB-INF/validation.xml" />
</plug-in>
</struts-config>
```

validation.xml:

```
<!DOCTYPE form-validation PUBLIC
    "-//Apache Software Foundation//DTD Commons Validator Rules Configuration
    1.1.3//EN"
    "http://jakarta.apache.org/commons/dtds/validator_1_1_3.dtd">
<form-validation>
    <formset>
        <form name="personForm">
            <field property="pid" depends="required">
                <arg key="err.pid" resource="true" />
            </field>
            <field property="name" depends="required">
                <arg key="err.name" resource="true" />
            </field>
            <field property="password" depends="required">
                <arg key="err.password" resource="true" />
            </field>
            <field property="sex" depends="required">
                <arg key="err.sex" resource="true" />
            </field>
            <field property="age" depends="required,integer">
                <arg key="err.age" resource="true" />
            </field>
        </form>
        <form name="idcardForm">
            <field property="pid" depends="required">
                <arg key="err.pid" resource="true" />
            </field>
            <field property="serial" depends="required">
                <arg key="err.serial" resource="true" />
            </field>
            <field property="expiry" depends="required,integer">
                <arg key="err.expiry" resource="true" />
            </field>
        </form>
    </formset>
</form-validation>
```

```
</form>
</formset>
</form-validation>
```

我们现在在项目之中使用框架：frameset、frame

ClassCastException → 类型转换异常

验证框架是比较好使，但是有些情况下也会造成某些负面的效果。

为一个人员增加身份证、修改一个人员的身份证信息、修改人员信息、删除人员信息

如果页面显示的是白板，则有可能是 EL 写错了

增加身份证，如果要增加身份证，肯定需要所属的 person 属性，person 属性之中要的只是 Person 这个 VO 的 pid 编号

增加身份证时，需要判断，如果身份证已经存在了，则显示用户身份证号，否则显示增加窗口

一对一关系在开发中也是较为常见的

但是我们此段代码使用的是 Struts 的验证框架，在框架的使用上配置较多，如果大家不是很熟悉，可以去看一下 Struts 中关于此段知识的介绍。