**需要的包** [**http://pan.baidu.com/s/1jIbQ9dc**](http://pan.baidu.com/s/1jIbQ9dc) **创建好项目之后拖到lib里就好**

Model :

**package** com.hdu.model;

@Entity （必须）

**public** **class** Manager {

（属性根据自己需要写）

**public** String username;

**public** String password;

**public** String name;

**public** **int** age;

@Id（主键属性写在get方法顶上） --hibernate 会根据标签属性自动建表

**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 getName() {

**return** name;

}

**public** **void** setName(String name) {

**this**.name = name;

}

**public** **int** getAge() {

**return** age;

}

**public** **void** setAge(**int** age) {

**this**.age = age;

}

}

Dao(以上面建的model为例)

**package** com.hdu.dao.impl;

（包必须引入）

**import** javax.annotation.Resource;

**import** org.springframework.orm.hibernate3.HibernateTemplate;

**import** org.springframework.stereotype.Component;

**import** com.hdu.dao.ManagerDao;

**import** com.hdu.model.Manager;

（标签必须）

@Component("managerDaoImpl（如ManagerDaoImpl 这里一般就写managerDao（不过经测试写什么都行））")

**public** **class** ManagerDaoImpl **implements** ManagerDao {

**private** HibernateTemplate hibernateTemplate;

@Override

（以add为例）

**public** **void** add(Manager manager) {

**this**.hibernateTemplate.save(manager); 这里以add为例（网上搜索hibernateTemplate可知语法）

}

**public** HibernateTemplate getHibernateTemplate() {

**return** hibernateTemplate;

}

@Resource

**public** **void** setHibernateTemplate(HibernateTemplate hibernateTemplate) {

**this**.hibernateTemplate = hibernateTemplate;

}

}

Service:

**package** com.hdu.service.impl;

(以下包必须要)

**import** javax.annotation.Resource;

**import** org.springframework.stereotype.Component;

**import** com.hdu.dao.ManagerDao;

**import** com.hdu.model.Manager;

**import** com.hdu.service.ManagerService;

（标签必须）

@Component("managerService")（命名方法和上面类似）

**public** **class** ManagerServiceImpl **implements** ManagerService {

**private** ManagerDao managerDao; （private 一个dao对象（不是daoimpl））

@Resource（必须）

**public** **void** setManagerDao(ManagerDao managerDao) {

**this**.managerDao = managerDao;

}

**public** ManagerDao getManagerDao() {

**return** managerDao;

}

**public** **boolean** registe(Manager manager) {

managerDao.add(manager);

**return** **true**;

}

**public** **void** delete(String username) {

}

**public** **void** update(String username, String password) {

}

}

Action:

**package** com.hdu.action;

**import** com.hdu.model.Manager;

**import** com.hdu.service.ManagerService;

**import** com.opensymphony.xwork2.ActionSupport;

**public** **class** RegisteAction **extends** ActionSupport(一定要继承) {

**private** ManagerService managerService; （和前面类似private 一个Service类）

（属性和前端属性名对应）

**public** String username;

**public** String password;

**public** String name;

**public** **int** age;

（

<s:form action=*"registe"*>

（和这里面的name对应）具体标签的使用网上可以找（这里只有textfield和password不过也够用了）

<s:textfield name=*"username"* label=*"your username"*></s:textfield>

<s:password name=*"password"* label=*"your password"*></s:password>

<s:textfield name=*"name"* label=*"your name"*></s:textfield>

<s:textfield name=*"age"* label=*"your age"*></s:textfield>

<s:submit></s:submit>

）

（get 方法和set 方法都要写）

**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 getName() {

**return** name;

}

**public** **void** setName(String name) {

**this**.name = name;

}

**public** **int** getAge() {

**return** age;

}

**public** **void** setAge(**int** age) {

**this**.age = age;

}

**public** String execute() **throws** Exception {

Manager manager = **new** Manager();

manager.setUsername(getUsername());

manager.setPassword(getPassword());

manager.setName(getName());

manager.setAge(getAge());

**boolean** flag = managerService.registe(manager);

**if**(flag) {

（这里return 的字符串和strusts.xml里的配置对应）

（

<action name=*"registe"* class=*"com.hdu.action.RegisteAction"*>

<result name=*"input(如果 return input就跳到register.jsp页面)"*>/register.jsp</result>

<result name=*"success（如果 return success就跳到success.jsp页面）"*>/success.jsp</result>

</action>

）

**return** "success";

} **else** {

**return** "input";

}

}

**public** ManagerService getManagerService() {

**return** managerService;

}

**public** **void** setManagerService(ManagerService managerService) {

**this**.managerService = managerService;

}

}

Jdbc.properties:

jdbc.driverClassName=com.mysql.jdbc.Driver

jdbc.url=jdbc\:mysql\://localhost\:3307(端口一般是3306我的比较特殊)/mydb（数据库名）

jdbc.username=root（账号）

jdbc.password=1111111（密码）

（下面不用改）

jdbc.maxActive=100

jdbc.maxIdle=30

jdbc.maxWait=500

jdbc.defaultAutoCommit=true

stucts.xml

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

<!DOCTYPE struts PUBLIC

"-//Apache Software Foundation//DTD Struts Configuration 2.1//EN"

"http://struts.apache.org/dtds/struts-2.1.dtd">

<struts>

<constant name=*"struts.custom.i18n.resources"* value=*"globalMessages"*></constant>

(前面不用改)

<action name=*"registe（和前端form头的action对应）"* class=*"com.hdu.action.RegisteAction（Action路径）"*>

<result name=*"input"*>/register.jsp</result>

<result name=*"success"*>/success.jsp</result>

</action>

</package>

</struts>

RegisteAction(和action类名对应这里以RegisteAction为例)-validation.xml

（用于检验输入）

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

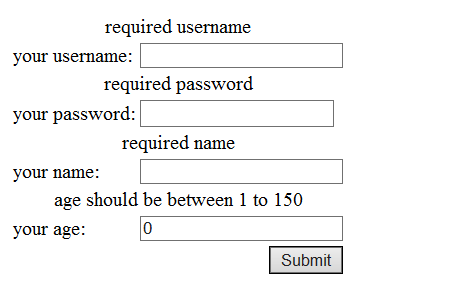
<!DOCTYPE validators PUBLIC "-//Apache Struts//XWork Validator 1.0.2//EN" "http://struts.apache.org/dtds/xwork-validator-1.0.2.dtd">

<validators>

<field name=*"username"*>(就是action里面的属性名)

<field-validator type=*"requiredstring"*>

<message>required username</message>（输入错误时的报错信息可自己随意填，输错就图片这个反应）



</field-validator>

</field>

<field name=*"password"*>

<field-validator type=*"requiredstring"*>

<message>required password</message>

</field-validator>

</field>

<field name=*"name"*>

<field-validator type=*"requiredstring"*>

<message>required name</message>

</field-validator>

</field>

<field name=*"age"*>

<field-validator type=*"required"*>

<message>required age</message>

</field-validator>

<field-validator type=*"int"*>

<param name=*"min"*>1</param>

<param name=*"max"*>150</param>

<message>age should be between ${min} to ${max}</message>

</field-validator>

</field>

</validators>

Web.xml不用改

## 目录结构

