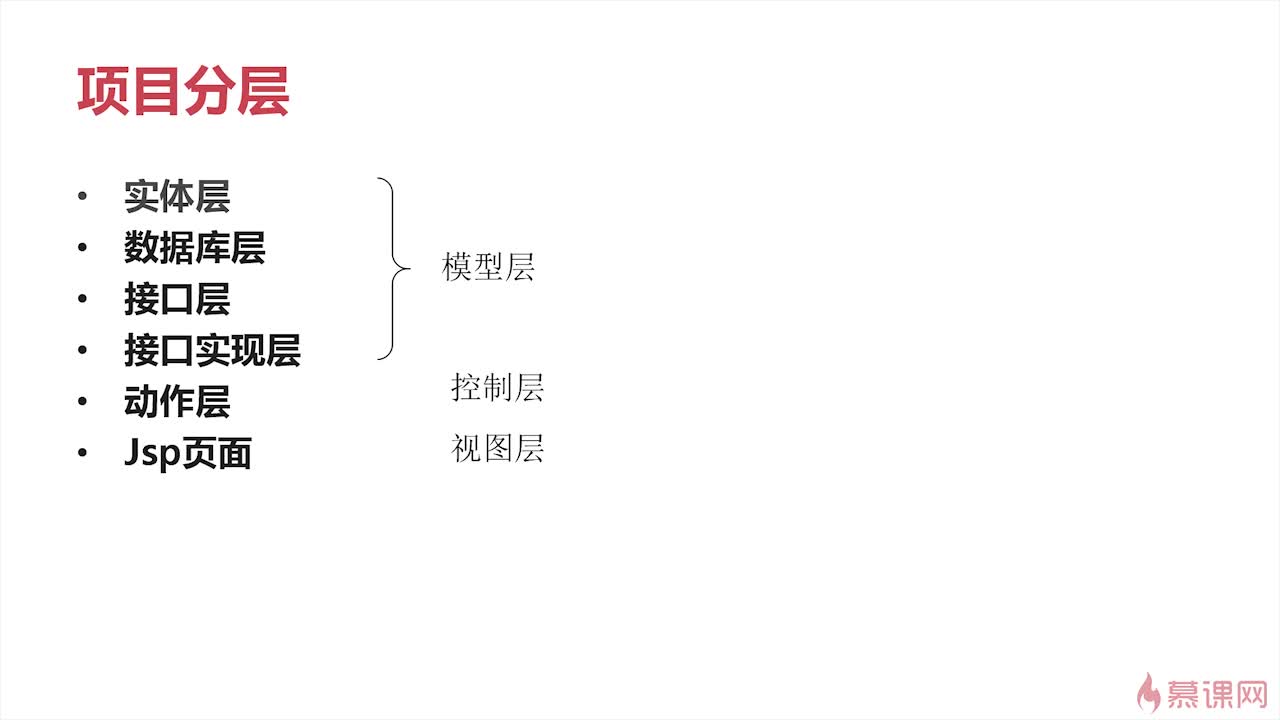
一、前期准备工作

1、Struts2和Hibernate整合



2、项目分层



* entity **实体层** – 持久化类

DB **数据库工具层** – 用来建立驱动连接

Dao 数据访问层 – 用来访问数据库CRUD，一个Dao对应一张表

Service **业务逻辑层** – 处理控制层发来的请求，如果需要访问数据库，直接调用Dao层；如果处理其他一些业务逻辑，比如说转换字符，直接在service层解决。一般分为接口类和实现类。

Action **控制层** – 利用servlet技术去处理视图层的请求，主要功能是接收参数，保存参数到session，转发页面，有关业务逻辑的方法交给service层解决。

JSP页面 **视图层** --- 主要在浏览器页面显示的界面，其他功能交给后端解决。

3、开发中的注意点

1. action类

* action接受表单数据的三种方式：

1.普通属性

2.领域对象

3.模型驱动

1. struts通配符使用

1.首先，注意头部信息，这个应该是用来指定文件中允许使用那些标签。

　　<!DOCTYPE struts PUBLIC"-//Apache Software Foundation//DTD Struts Configuration 2.5//EN" "http://struts.apache.org/dtds/struts-2.5.dtd">

2.加上下面这句。

　　<global-allowed-methods>regex:.\*</global-allowed-methods>

　 或者（不加上面这句），在action中加上指定允许调用的方法的语句：

　　<allowed-methods>login,logout</allowed-methods>

1. Struts2项目：在JSP页面中引用s标签的问题

* 引用标签：<%@ taglib prefix="s" uri="/struts-tags" %>
* 在web.xml中修改过滤器

<filter-mapping>

<filter-name>struts2</filter-name>

<url-pattern>\*.jsp</url-pattern>

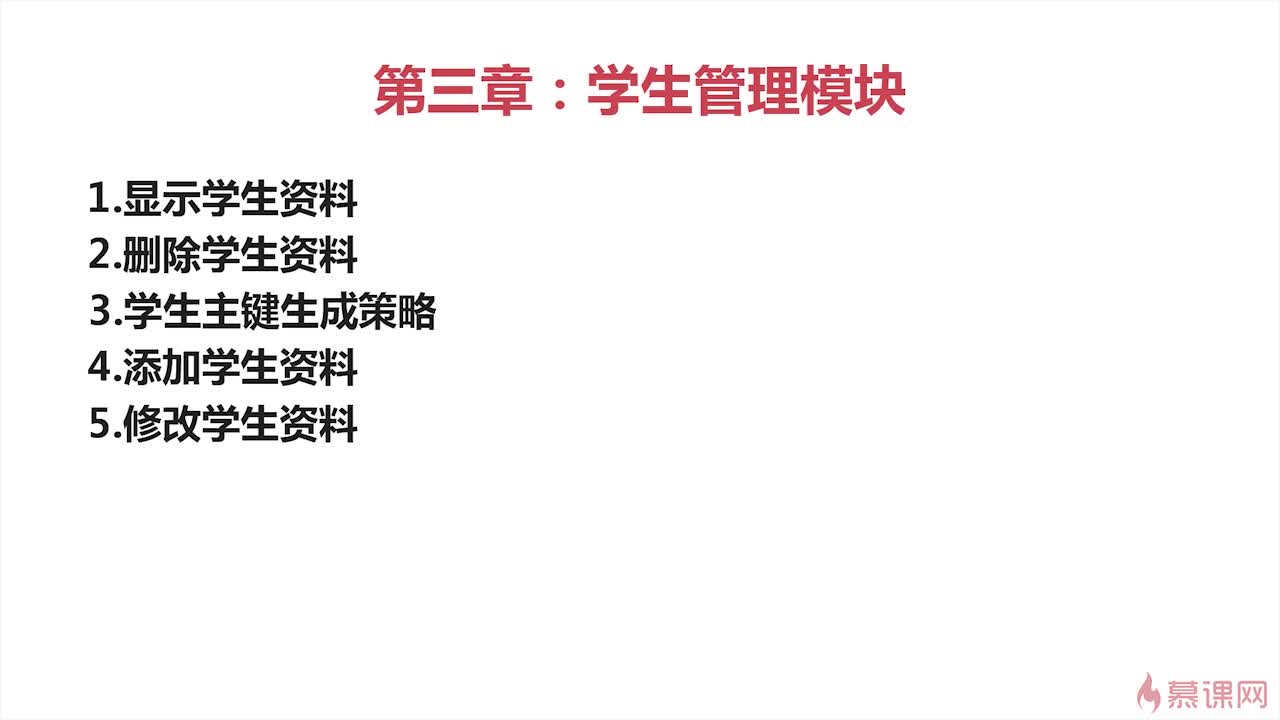
</filter-mapping>

* 登录不成功的错误提示信息，struts2 2.5中要添加一个<result name=”input”>,

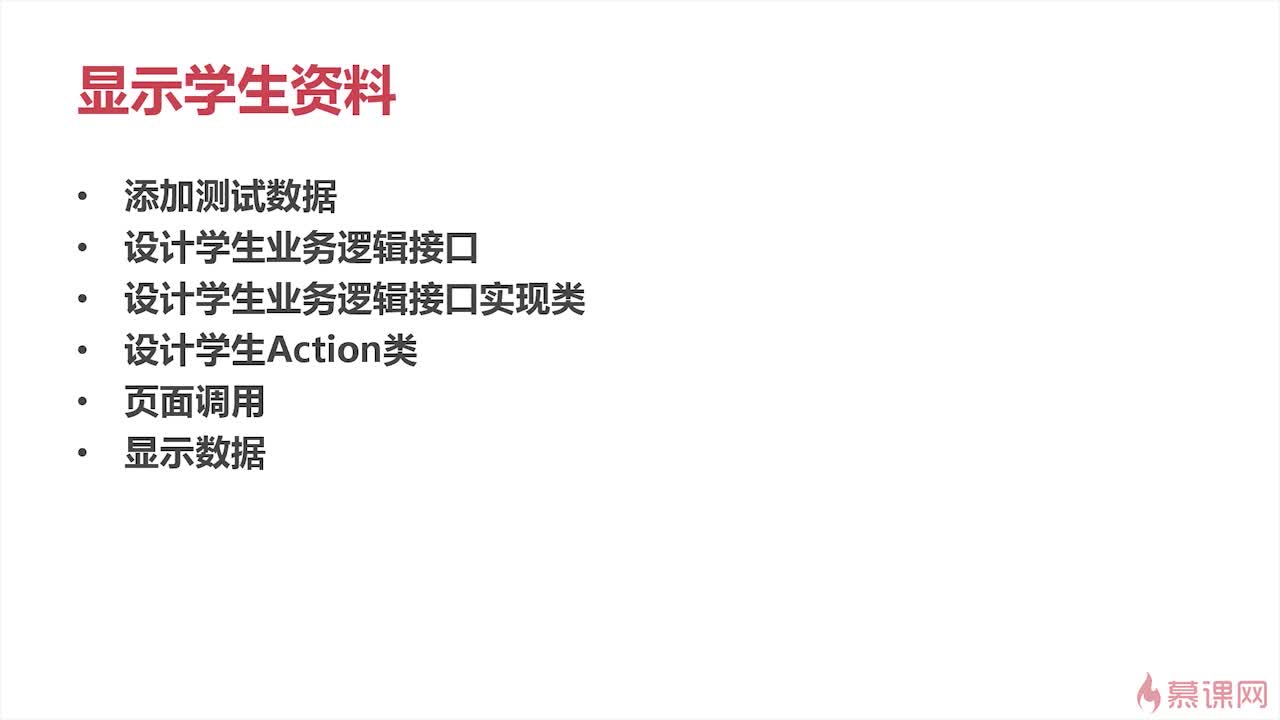
4、开发步骤

（1）用户模块：从Brower获取用户，与数据库进行比对。--完成验证功能

（2）学生模块：学生信息的CURD操作



* 显示学生资料



* 删除学生资料

1. Dao层：重写delete方法，根据传入的sid变量，获取相对应的students对象，调用session.delete方法删除，返回true。--负责业务逻辑以及传入的数据处理
2. Action层：从页面获取sid的值，调用Dao层的delete方法，返回到delete\_success界面。--负责页面接收，转发。
3. delete\_success通过chain内部跳转到查询页面Students\_query\_success界面，完成页面刷新。---负责显示

* 添加学生资料

1. Dao层：重写add代码，页面传入一个Students对象，为对象生成主键，并将对象添加到数据库中。----添加方法和生成主键
2. Action层：从页面获取关于新添加用户的信息，保存到Students对象当中，调用add方法，转发到添加成功页面。---接收参数、保存session、转发页面

* 将String内容转换成Date类型

SimpleDateFormat sdf = new SimpleDateFormat(“yyyy-MM-dd”)；

s.setBirthday(sdf.parse(request.getParameter(“birthday”))；

* 修改学生资料

业务逻辑代码：查询学生资料并显示到页面上

1. dao层：通过传入的sid，利用students类的反射来获取students对象。
2. action层：从页面获取sid，调用dao层的方法来获取一个students对象，将对象保存到session中，转发到修改页面

业务逻辑代码：将修改后的资料保存到数据库当中

1. dao层：通过传入的students对象，更新数据库中相对应的对象
2. action层：从页面session获取students对象，并将更新后的students对象信息重新获取一遍，调用dao层的update方法更新数据库中的students对象