

# 实例讲解 SSH 技术架构

## 前言

Struts + Spring + Hibernate 是近两年在 Web Application 中使用的比较多的技术，虽然新的技术不断涌现，SSH 可能终将成为昨日黄花，但是在其框架间的思想却是不会过时的。故而希望借此机会与师姐妹们共同学习☺

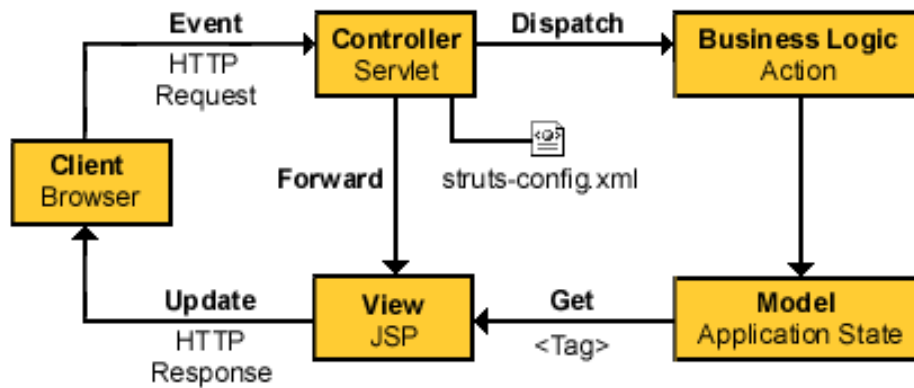
注：部分资料和图片来自 Internet 在此向原作者表示诚挚的感谢！

## 简介

我是个懒人，没有仔细地查 Struts, Spring 和 Hibernate 的准确定义；这里和大家分享下我的认识：

### Struts

Struts 是一个建立在 Servlet 和 JSP 技术基础上的 Web Application 表示层的技术框架，它实现了 MVC2 的模式，从而使得表示层中的视图和表示逻辑较好的分离（换言之，就是说你不用写大量的 java 代码在你的 JSP 中了）

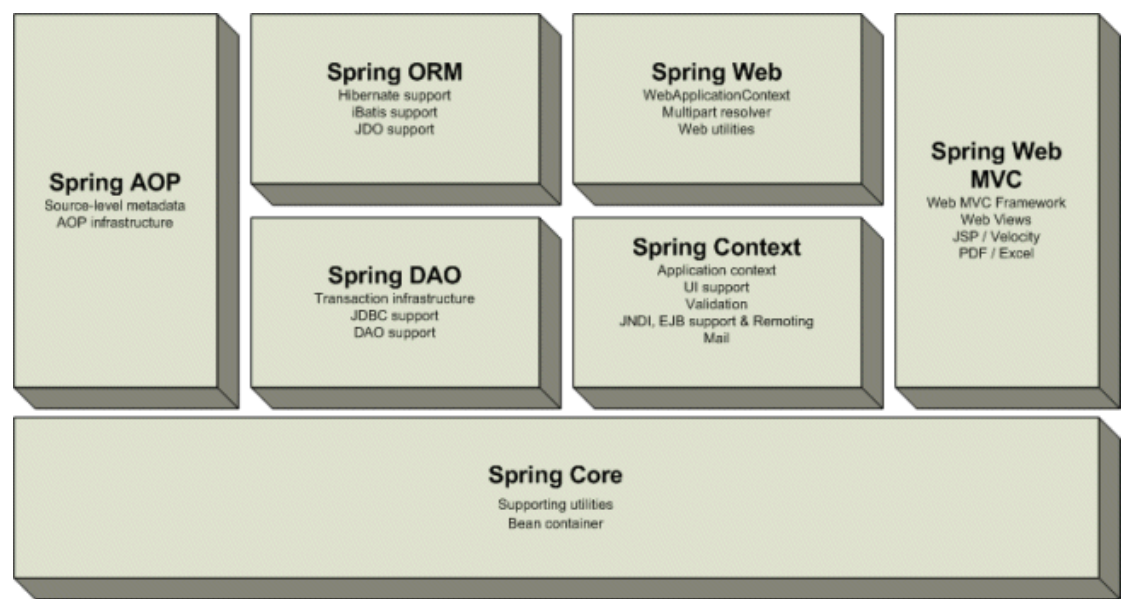


图表 1 Struts MVC2

### Spring

Spring 是一个轻量级的模块化框架，我们常用它来取代业务逻辑层的应用服务器，由它来提供原来由应用服务器提供的诸如对象创建，事务管理，日志等等公共基础设施。它实现了 Inverse of Control (Dependency Injection) 和 Aspect Oriented Programming 的编程模型。简单说就是：IoC 使得对象的初始化完全由 Spring 来负责，因此对象实例之间的关系成为可声明的（在 Spring 的配置文件中声明）；AOP 使得我们不再为跨越系统各个模块的公共问题伤脑

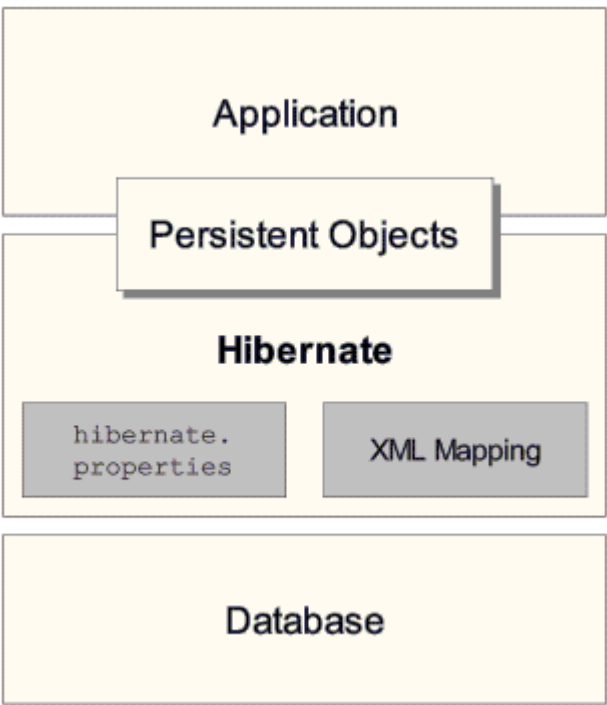
筋（事务、安全等等），Spring 为我们提供了接近完全透明的解决方案。



图表 2 Spring Architecture

## Hibernate

Hibernate 是一个数据库访问框架，着力于提供可声明的（declarative）持久化服务。



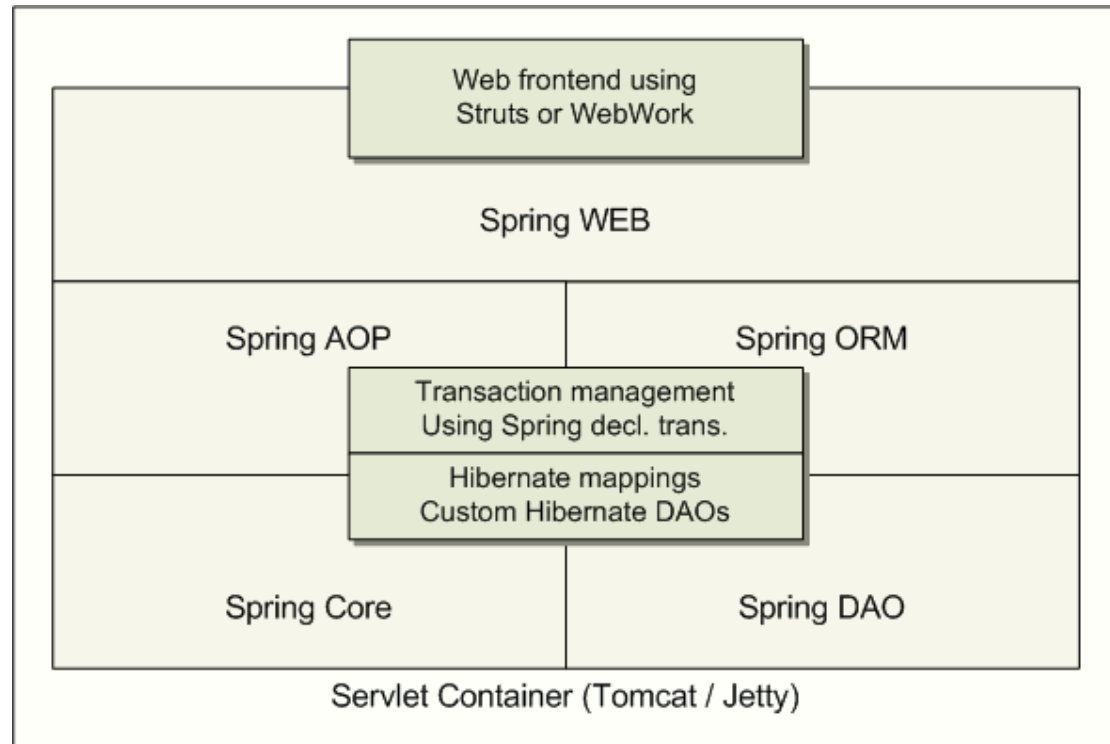
图表 3 Hibernate

# 准备实例

这里我们有一个具体的例程来和大家一同学习 SSH:

[SelectiveCourseManagementSystem.rar](#)

在这个例子里具体的来说我们就是要使用 Spring 为我们业务逻辑层提供事务服务，并以它为胶水来把 Struts 和 Hibernate 整合起来。



图表 4 Spring in Middle Tier

## 例程环境

MyEclipse 5.5 GA (eclipse 3.1 以上也可以, MyEclipse 好用些而已)

MySQL 4.0 以上

## 开始前的准备

- 1) 将压缩文件直接解压后导入 Eclipse
- 2) 新建一个 create.sql 脚本，将 sql.txt 文件中 sql 脚本 copy 过来
- 3) 在 Eclipse 中连接数据库，运行 create.sql 脚本
- 4) 在新数据库中加入数据请参照下述 Domain Model，注意数据之间的关系（请添加数据，否则无法测试）
- 5) 将 application-service.xml 中的数据库连接的属性修改为当前数据库服务器对应的值
- 6) 修改项目的 build path 中已经无效的 mysql connector jar，将其替换为恰当的 connector
- 7) 部署项目
- 8) start Tomcat 在浏览器中测试<your-tomcate-stratpoint>/StudentManger

Ok, the SSH application is running now! 让我们开始分析☺

## 简单的需求分析

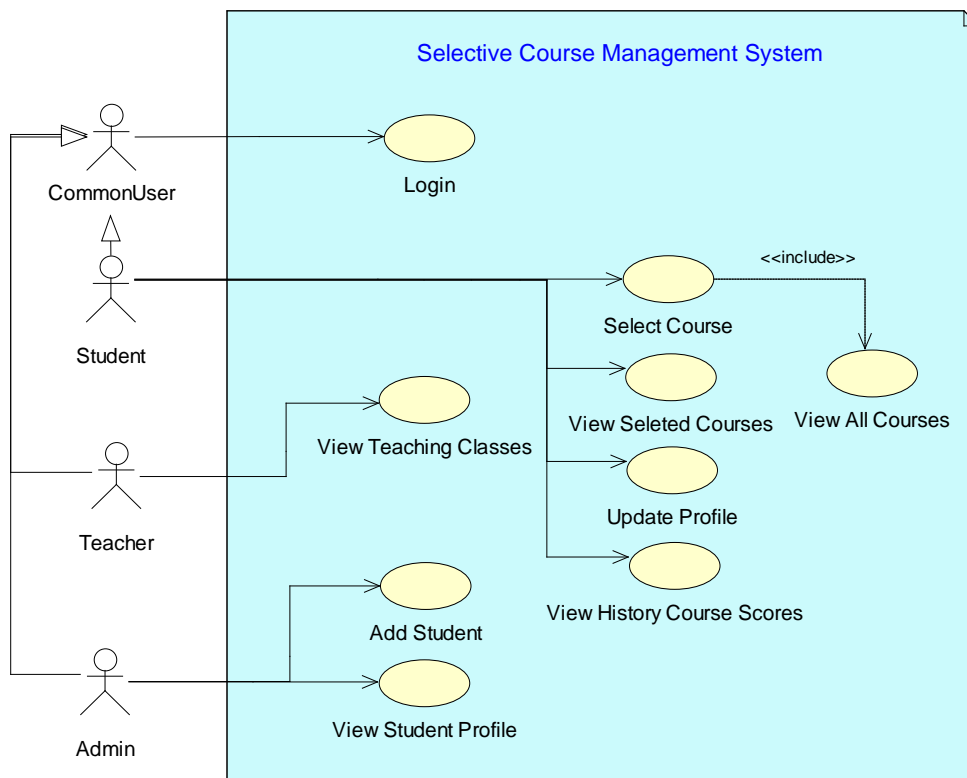
注：简便起见，这里我们没有完整的需求分析，大家注意在实践中还是要完整的需求过程的。

这个应用是个简单的选修课管理系统：

# Use Case 分析

## Selective Course Management System Use Case Diagram

Version 1.0  
Author 王青  
Date 08-04-2007



图表 5 Use Case Diagram

## 描述

- 1) 该系统可以方便处理学生在课程选报，学分查询，成绩查询，以及 成绩发布等。
- 2) 系统以班级为核心，一门课程可以对应多个班级。
- 3) 一名教师可以带不同的班级。
- 4) 学生可以选报不同课程所对应的班级。
- 5) 班级自身有目前人数，和最大人数，以及上课时间，上课地点的属性。
- 6) 学生在选报班级之后，班级的人数会自动加一，直到等于最大人数时，其他学生将会有人数已满的错误提示。
- 7) 如果学生选择了同一课程的不同班级，也将收到错误提示。
- 8) 学生有密码，系别，学分，地址，电话等属性。
- 9) 教师在系统中主要负责成绩发布。
- 10) 教师可以对其所带的班级的学生的成绩修改。

- 11) 系统会以成绩是否大于等于 60 来判断学生是否通过考试，如果通过会将该课程的学分累加到学生学分
- 12) 如果教师二次修改了成绩，而且小于 60，系统会在学生学分上扣掉该课程的分。
- 13) 课程在系统中具体体现为班级，自身带有学分属性。
- 14) 系有编号，名称的属性，同时可以作为联系教师，课程，学生的桥梁。

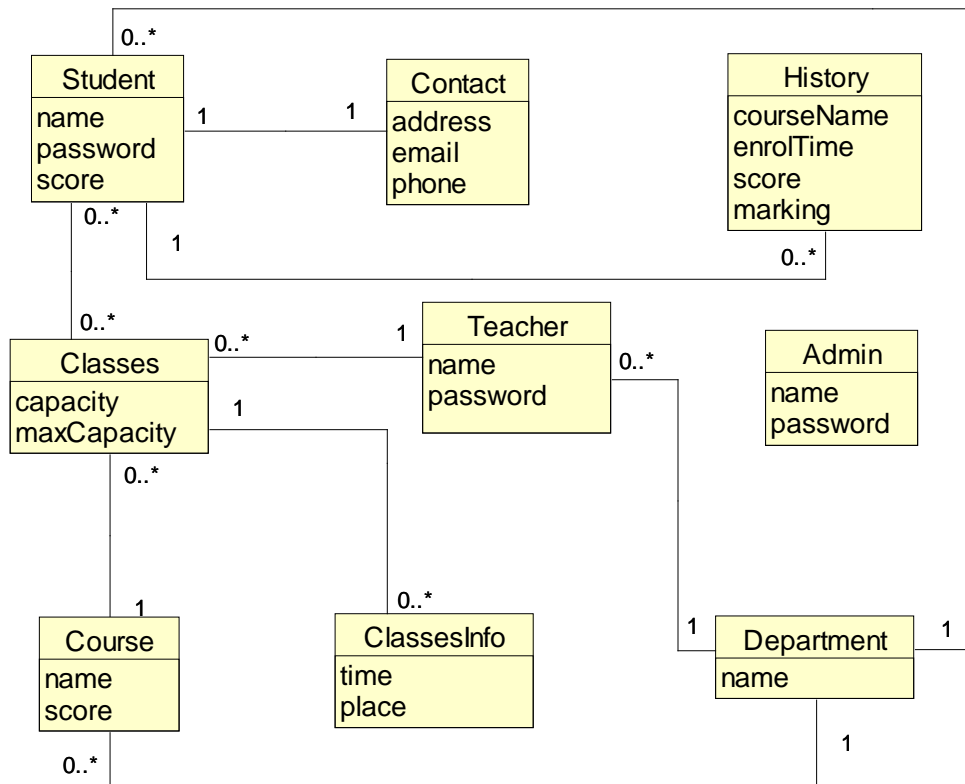
## 功能模块

- 1) 身份验证模块： 根据用户名，密码，用户类别 转发用户到不同的模块。
- 2) 学生模块： 查看课程，查看班级，选报课程，查看已选课程，成绩查询。
- 3) 教师模块： 录入成绩
- 4) 管理员模块： 对学生，教师，课程，班级，系 增，删，查，改。

# Domain Model

## Selective Course Management System Domain Model

Version 1.0  
Author 王青  
Date 08-04-2007



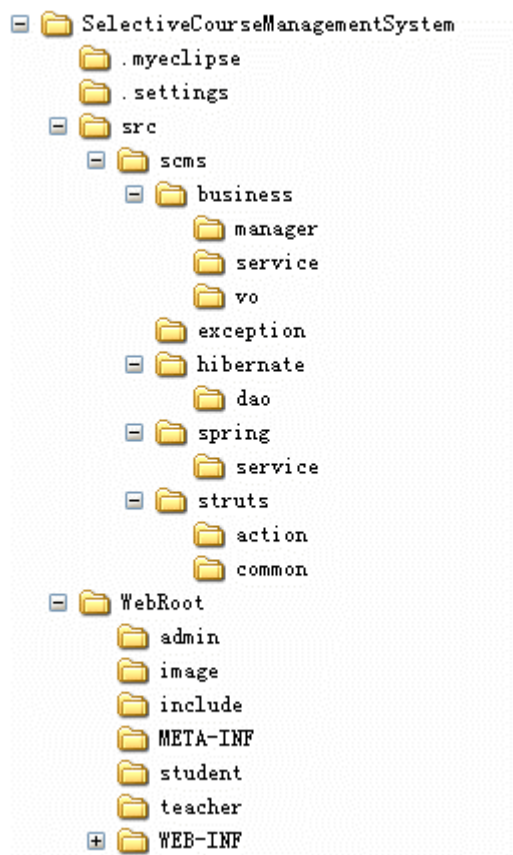
图表 6 Domain Model

Ok, 这里我们对待建的系统已经有了较为清楚的认识。在接下来的部分，我们将略过设计和 coding 直接来看我们最后的成果，从中学习 SSH

## 实例解析

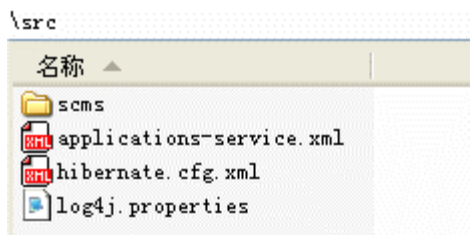
### 包结构剖析

首先我们推荐大家在这次的实训当中采用类似的包结构：



## 全局

在顶层有项目最重要的配置文件



- 1) **application-service.xml** 是 Spring 的配置文件，它定义了数据库连接，和 Spring 管理的对象，事务以及胶合 Hibernate 的配置。
- 2) **hibernate.cfg.xml** 是 Struts 的配置文件，它定义了 ORM 配置文件的路径(注意:Hibernate 中的 Session 和 Servlet 中的 session 各有所指，前者是一次数据库连接，后者是一次应用会话)
- 3) **log4j.properties** Spring 利用 log4j 来作为 logging 的工具，这是 log4j 的配置文件

## 业务逻辑

业务逻辑层分成了实体/值类包 (**vo**)，操作类包 (**manager**) 和服务类包 (**service**)。

注意：这是一种以对象类型来组装业务逻辑层对象的横向分包方式，当项目很小，业务实体



数量很少的情况下，我们经常采用这样的分包。如果待建系统中的业务实体数量较多，我们就会采用以业务逻辑来纵向分包的方法，将业务上相关的实体和其 `manager` 放在同一个包内。

## 实体类



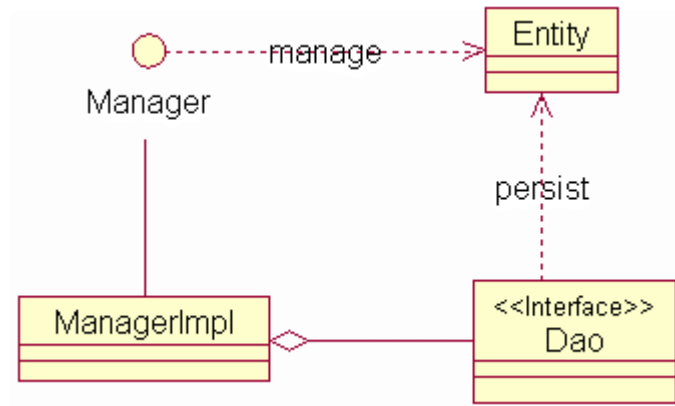
业务实体类，注意 MyEclipse 可以从数据库连接中得到 ORM 文件（.hbm.xml），然后可以在此基础上自动的为你生成这些实体类。

## 操纵类

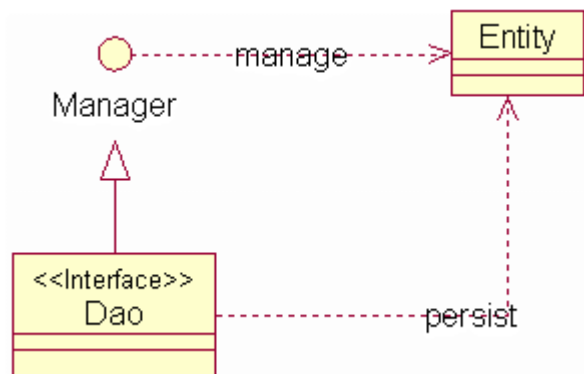


用以操作实体的 `Manager`，它们提供了最基本的对实体 `CRUD` 的能力，和只涉及它所管理对象的简单业务方法。

注意：在这里出于简化将 `Manager` 与 `Dao` 的关系：

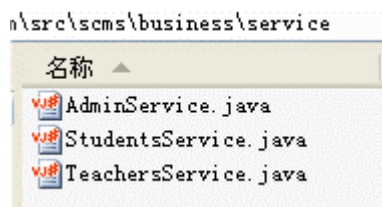


重构为：



并略去了 ManagerImpl

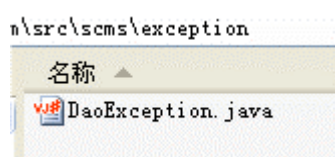
## 业务服务



这里包含了各个功能模块中要用到的涉及多个业务实体的业务方法。

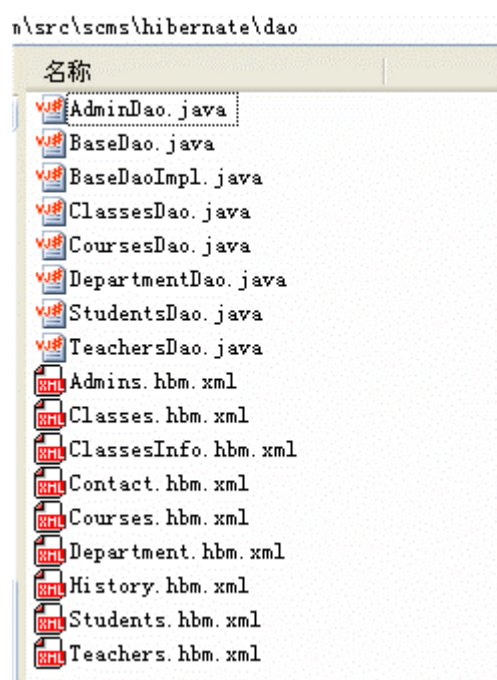
**注意：**这里的 manager 和 service 都是接口，具体的实现将分别由与 hibernate 和 spring 耦合的类来提供

## 自定义异常



这里的异常还很不完善，师姐妹们有时间可以重构整个实例中的异常设计

## 数据访问（Data Access Object）



与 hibernate 耦合的类。如前述,这些 Dao 实现了 Manager 接口。而.hbm.xml 文件就是 Hibernate 的 ORM 定义文件。这里要注意两点:

### 多一映射和多多映射

固然我们可以利用 MyEclipse 生成 ORM 定义,但是这些定义对于利用关联表表示的关联关系通常不能够正确的定义,我们需要手工修改。

### 分页需求

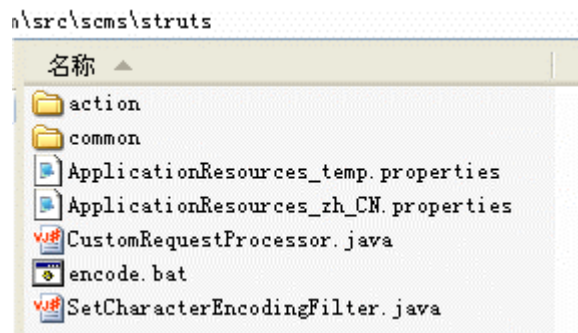
为了满足 UI 层在查询结果较多的时候要分页的需求,这里我们引入了 BaseDao 接口,全面的管理对 Session 的操作。注意我们分页的需求是通过 JSP 的 custom tag, Struts 和这里的 BaseDao 接口共同解决的。

### Web 界面

我们的 Web 层采用 Struts 框架,因为此系统属于较简单 WEB 应用,采用了 OpenSessionInView 模式处理 LazyLoad 问题,这样我们可以在用户视图中使用 get, set 方法来方便地获取关联对象。为了处理庞大的 Action 和 ActionForm 问题,在此我们使用 DynaActionForm (DynaValidatorForm)和 DispatchAction 以及 动态验证框架来解决。同时我们还使用 Tile 标签来解决页面布局问题,并使用自定义标签处理分页和身份验证问题。

## OpenSessionInView

由于 Hibernate 引入了 Lazy Load 特性，使得脱离 Hibernate 的 Session 周期的对象如果再通过 getter 方法取到其关联对象的值，Hibernate 会抛出一个 LazyLoad 的 Exception。所以为了解决这个问题，Spring 引入了 OpenSessionInViewFilter，其主要意思是在发起一个页面请求时 打开 Hibernate 的 Session，一直保持这个 Session，直到这个请求结束，通过这个 Filter 使得 Hibernate 的 Session 的生命周期 变长。

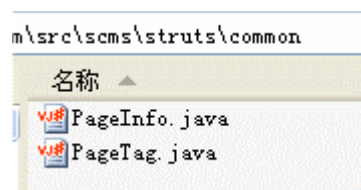


ApplicationResources。。。是 Struts 框架中的国际化（i18n）资源

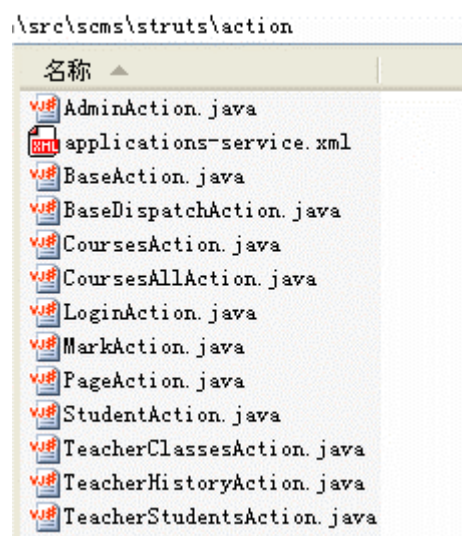
CustomRequestProcess.java 提供了 Struts 控制器 actionServlet 中的 process，用以认证用户

SetCharacterEncodingFilter.java 是一个 Servlet 的 Filter 用以处理字符编码问题

encode.bat 是一个项目中用的小工具，用以将 ApplicationResource\_temp 中的 GBK 编码转换成 UTF-8 编码



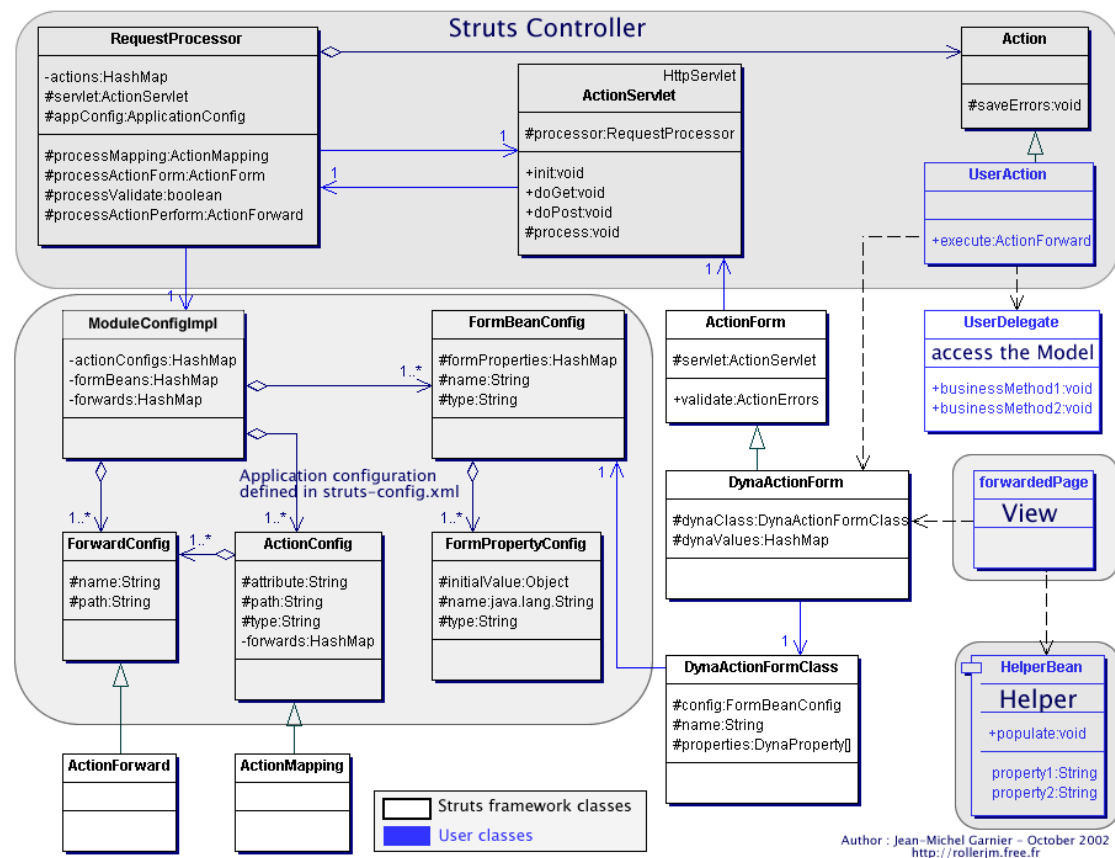
用以解决分页问题的帮助类



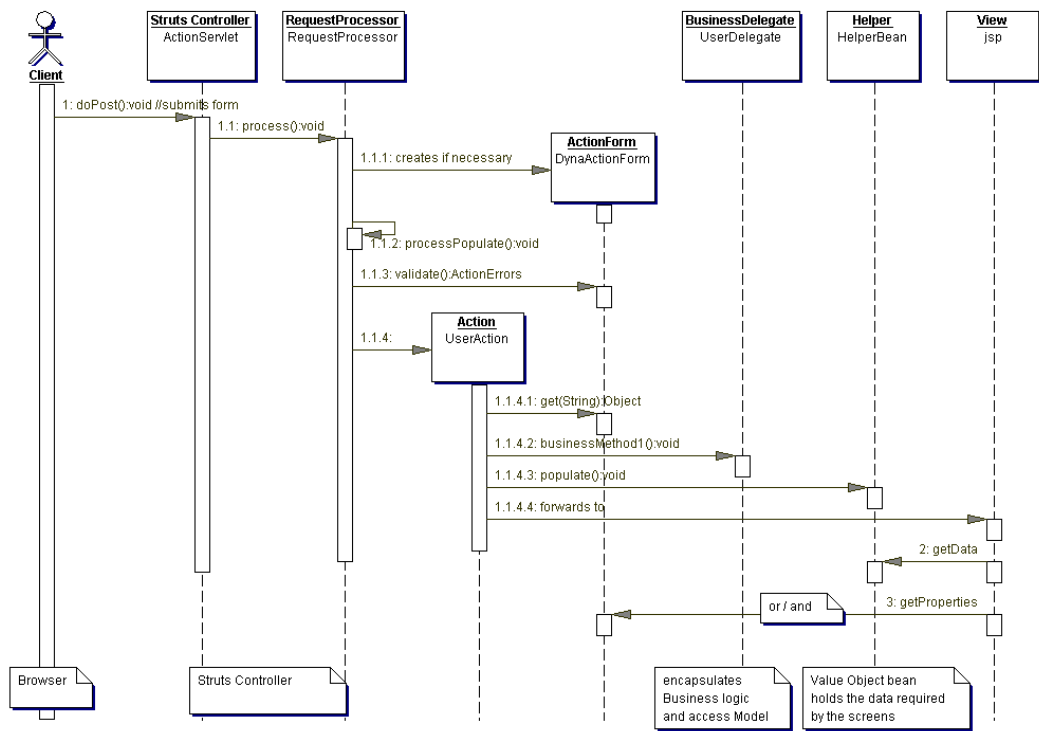
Struts 的 action 类，是 Web UI 层和业务逻辑层打交道的地方。这里我们简化了在大型应用中常用的 Business Delegate + Service Locator 的模式，使用 BaseAction 和 BaseDispatchAction 来持有业务逻辑层 service 的引用。而后其余的 action 均扩展自这两个 action

## Struts 框架复习

一起来复习下 Struts 框架：



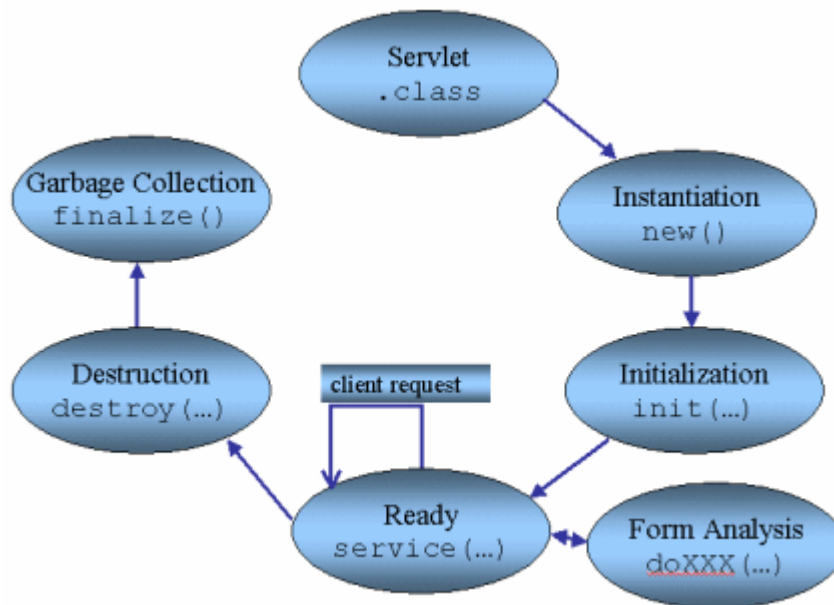
图表 7 Struts Architecture



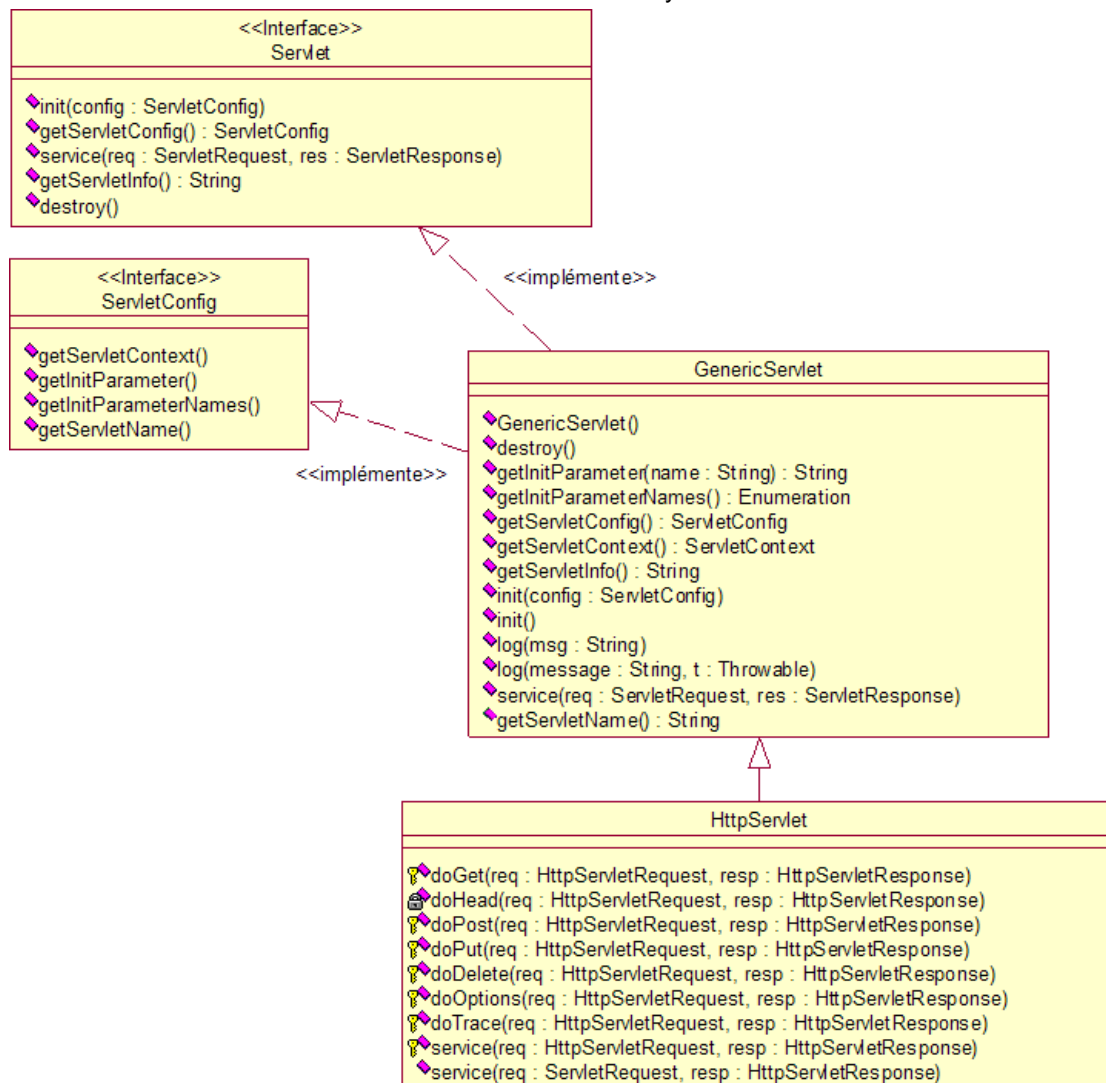
图表 8 Struts Sequence Diagram



图表 9 Struts Action

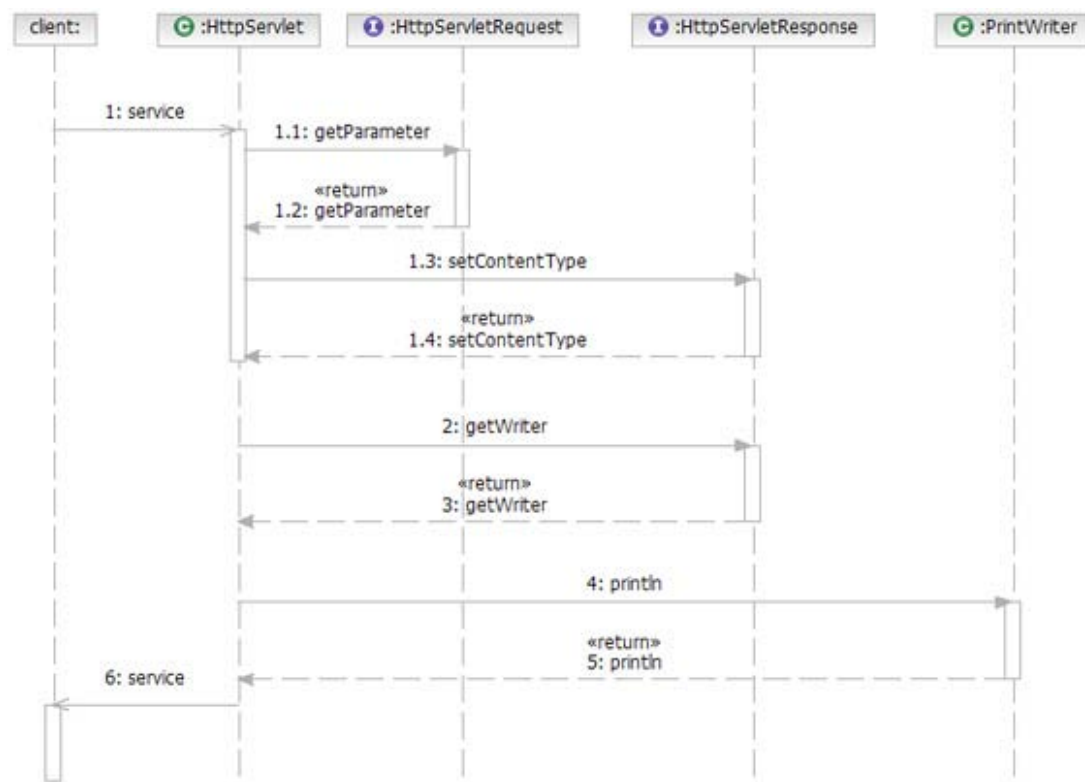


图表 10 Servlet Lifecycle

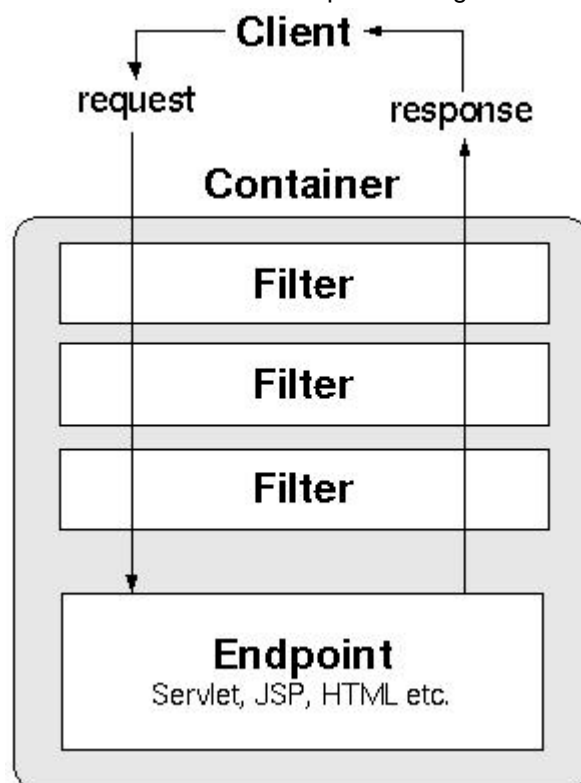


图表 11 Servlet Class Diagram





图表 12 Servlet Sequence Diagram



图表 13 Servlet Filter

## JSP 页面

