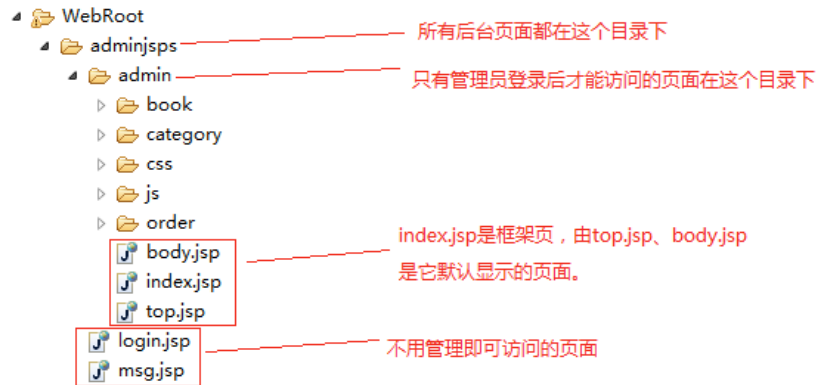


后台功能实现

1 后台管理介绍

1.1 后台页面

后台页面都在 adminjps 目录下：



1.2 后台功能模块介绍

后台功能模块分为：

- 管理员模块：
 - 管理员登录；
 - 管理员退出；
- 分类管理模块：
 - 查看所有分类；
 - 添加一级分类；
 - 添加二级分类；
 - 编辑一级分类；
 - 编辑二级分类；
 - 删除一级分类；
 - 删除二级分类；
- 图书管理模块：
 - 查看指定分类图书；
 - 查看指定条件图书；
 - 添加新图书；

- 编辑图书;
- 删除图书;
- 订单管理模块:
 - 查看所有订单;
 - 查看指定状态订单;
 - 查看订单详细信息;
 - 取消订单;
 - 订单发货。

其中只有管理员模块是前台没有的,其他三个模块前台都有。但是,尽管前台有这三个模块,前后台也不能共享 **Servlet**! 但前后台可以共享 **Service** 和 **DAO** 层! 所以只有管理员模块需要创建所有层的类,而其他三个模块只需要创建 **Servlet** 层类即可。

后台有很多功能与前台是相同的,例如按分类查看图书功能,只是转发的页面是后台页面而已,所以这里很多功能就不在赘述了。

2 后台管理员模块

后台管理员模块功能:

- 管理员登录;
- 管理员退出。

2.1 后台管理员相关类创建

- `cn.itcast.goods.admin.admin.domain.Admin;`
- `cn.itcast.goods.user.admin.admin.dao.AdminDao;`
- `cn.itcast.goods.user.admin.admin.service.AdminService;`
- `cn.itcast.goods.user.admin.admin.web.servlet.AdminServlet。`

`Admin` 类对应 `t_admin` 表,不再废话,直接给出基本代码:

`Admin.java`

```
public class Admin {  
    private String adminId;//ID  
    private String adminname;//管理员名  
    private String adminpwd;//管理员密码  
    ...  
}
```

2.2 管理员登录

`login.jsp` 是后台登录页面,页面本身提供了表单显示功能。提交表单请求 `AdminServlet#login()` 方法:

- 把表单数据封装到 `Admin` 对象中;
- 使用表单数据调用 `AdminService#login()` 方法查询用户;

- 如果查询到用户说明登录成功：把查询到的用户保存到 session 中，转发到 index.jsp；
- 如果没有查询到，说明登录失败：在 request 中保存错误信息，转发到 login.jsp。

登录成功后转发到/adminjsps/admin/index.jsp 页面，该页面 top.jsp 部分需要显示当前用户名，所以需要修改该页面。

2.3 管理员退出

top.jsp 页面中存在“退出”链接，让该链接请求 AdminServlet#quit()方法。quit()方法从 session 中移除当前用户，转发到 login.jsp 页面。

3 分类管理

分类管理功能：

- 查看所有分类；
- 添加一级分类；
- 添加二级分类；
- 编辑一级分类；
- 编辑二级分类；
- 删除一级分类；
- 删除二级分类；

3.1 分类管理相关类创建

Service 和 Dao 与前台共享即可，无需再创建，而 Servlet 需要单独创建。

- cn.itcast.bookstore.admin.category.web.servlet.AdminCategoryServlet；

在 web.xml 中配置该类时，<url-pattern>为/admin/AdminCategoryServlet，方法后面添加登录过滤器。不只是该模块的 Servlet 如此，图书和订单模块都是如此。

3.2 查看所有分类



在 top.jsp 页面点击“分类管理”链接完成查看所有分类。该链接请求 AdminCategoryServlet#findAll()方法，得到所有分类，即 List<Category>，然后保存到 request 中，转发到/adminjsps/admin/category/list.jsp 页面显示。

CategoryService 和 CategoryDao 都已经写好了，所以无需再写！

3.3 添加一级分类

从 `/adminjsps/admin/category/list.jsp` 中 点 击 添 加 一 级 分 类 到 达 `/adminjsps/admin/category/add.jsp` 页面。

在 `add.jsp` 页面提交表单，请求 `AdminCategoryServlet#addOneLevel()`方法，完成添加分类。`addOneLevel()`在添加完成后，调用 `AdminCategoryServlet#findAll()`方法，回到 `list.jsp` 显示所有分类。

3.4 添加二级分类

添加二级分类共两步：一是到达表单页面，二是提交表单完成添加。

1. 到达表单页面

在 `/adminjsps/admin/category/list.jsp` 中，在某个一级分类上点击“添加二级分类”，请求 `AdminCategoryServlet#addTwoLevelPre()`方法，传递一级分类 `cid`。`addTwoLevelPre()`会获取所有一级分类，以及当前一级分类 `cid`，然后转发到 `/adminjsps/admin/category/add2.jsp` 页面。

添加2级分类

分类名称:

一级分类:

程序设计

===选择1级分类===

程序设计

办公室用书

图形 图像 多媒体

操作系统/系统开发

分类描述:

在 `add2.jsp` 页面中显示所有一级分类，因为二级分类有父分类。

2. 完成添加二级分类

在 `add2.jsp` 页面提交表单，完成添加二级分类。提交表单请求 `AdminCategoryServlet` 的 `addTwoLevel()`方法：

- 封装表单数据到 `Category child` 对象中；
- 给 `child` 指定 `cid`；
- 获取表单 `pid`，创建 `Category parent`，把 `pid` 设置给 `parent` 的 `cid`；
- 把 `parent` 设置给 `child`；
- 调用 `categoryService#add(Category)`方法完成添加；
- 调用 `AdminCategoryServlet#findAll()`，回到 `list.jsp`。

3.5 修改一级分类

修改一级分类分为两步：

- 在 `list.jsp` 中点击某个一级分类上的“修改”，到达 `edit.jsp`；

- 在 `edit.jsp` 页面提交表单。

1. 到达 `edit.jsp`

在 `/adminjsps/admin/category/list.jsp` 页面上，点击某个一级分类上的“修改”链接，请求 `AdminCategoryServlet#editOneLevelPre()` 方法，传递 `cid`。

`editOneLevelPre()` 方法通过 `cid` 加载当前分类，保存到 `request` 中，转发到 `edit.jsp`，在表单中显示。

2. 提交表单完成修改一级分类

在 `edit.jsp` 页面提交表单，请求 `AdminCategoryServlet#editOneLevel()` 方法，传递表单数据，其中表单中需要添加隐藏字段 `cid`，即被修改的分类 `id`。

`editOneLevel()` 调用 `service` 完成修改后，返回到 `list.jsp`。

3.6 修改二级分类

修改二级分类也是分为两步：

- 在 `list.jsp` 中点击某个二级分类上的“修改”，到达 `edit2.jsp`；
- 在 `edit2.jsp` 页面提交表单。

1. 到达 `edit2.jsp`

在 `/adminjsps/admin/category/list.jsp` 页面上，点击某个二级分类上的“修改”链接，请求 `AdminCategoryServlet#editTwoLevelPre()` 方法，传递 `cid`。

`editTwoLevelPre()` 方法通过 `cid` 加载当前分类，保存到 `request` 中，再加载所有一级分类保存到 `request` 中。转发到 `/adminjsps/admin/category/edit2.jsp` 页面。

`edit2.jsp` 在表单中显示当前二级 `Category` 对象，而且还要把所有一级分类显示到下拉列表中。然后让下拉列表上显示当前 `Category` 的父分类。

2. 在 `edit2.jsp` 提交表单，完成修改

在 `/adminjsps/admin/edit2.jsp` 页面上提交表单，请求 `AdminCategoryServlet#editTwoLevel()` 方法，传递表单数据，还有添加隐藏字段 `cid`，即要修改的分类 `cid`。

`editTwoLevel()` 方法获取表单数据，封装到 `Category child` 对象中，再获取 `pid`，创建 `Category parent`，把 `pid` 设置给 `parent`，再把 `parent` 设置给 `child`，然后调用 `categoryService` 的 `edit(Category)` 方法完成修改。最后回到 `list.jsp` 显示。

3.7 删除一级分类

在 `/adminjsps/admin/category/list.jsp` 页面上，点击某个一级分类上的“删除”链接完成删除。点击链接请求 `AdminCategoryServlet#deleteOneLevel()` 方法，传递当前一级分类的 `cid`。

`deleteOneLevel()` 方法首先获取 `cid`，然后调用 `CategoryService` 的 `deleteOneLevel()` 方法完成删除，转发到 `list.jsp` 显示。注意，`CategoryService#deleteOneLevel()` 方法可能会抛出 `CategoryService` 异常，如果抛出了异常，那么 `Servlet` 会把异常信息保存到 `request` 中，转发到 `/adminjsps/msg.jsp` 页面显示。

`CategoryService#deleteOneLevel()` 方法会先判断该 1 级分类下是否存在子分类，如果存在那么它会抛出 `CategoryService` 异常；如果不存在，那么删除该分类。

3.8 删除二级分类

在/adminjsps/admin/category/list.jsp 页面上，点击某个二级分类上的“删除”链接完成删除。点击链接会请求 ActionCategoryServlet#deleteTwoLevel()方法，传递当前二级分类的 cid。

deleteTwoLevel()方法首先获取 cid，然后调用 CategoryService 的 deleteTwoLevel()方法完成删除，转发到 list.jsp 显示。注意，CategoryService#deleteTwoLevel()方法可能会抛出 CategoryException 异常，如果抛出了异常，那么 Servlet 会把异常信息保存到 request 中，转发到/adminjsps/msg.jsp 页面显示。

CategoryService#deleteTwoLevel()方法会先判断该 2 级分类下是否存在图书，如果存在那么抛出 CategoryException 异常；如果不存在，那么删除该分类。

4 图书管理

在 top.jsp 上点击“图书管理”进入图书管理主页：/adminjsps/admin/book/main.jsp，main.jsp 是框架页，分为左部和中部，其中左部为所有分类（与前台主页左部相同），中部默认显示 body.jsp，body.jsp 有两个链接：添加图书和高级搜索。



图书管理功能如下：

- 按分类查询图书；
- 按作者查询图书；
- 按出版社查询图书；
- 高级搜索（多条件组合查询）；
- 查询图书详细；
- 添加图书；
- 编辑图书；
- 删除图书。

后台图书管理中有很多功能与前台一样，可参照前台完成。

4.1 与前台相同功能实现

1. 左部显示所有分类

- 修改 main.jsp 中 left 框架，让其请求 AdminCategoryServlet 的 findAllForBook() 方法；
- 为 AdminCategoryServlet 类添加 findAllForBook() 方法，让其获取所有分类，然后转发到 /adminjsps/admin/book/left.jsp 页面；
- 在 /adminjsps/admin/book/left.jsp 页面中使用 Javascript 小工具（Q6Menu）显示所有分类。

2. 按分类查询图书（分页显示）

- 点击 Q6Menu 上的 2 级分类，请求 AdminBookServlet#findByCategory() 方法；
- 该方法实现参照 BookServlet#findByCategory() 方法，但最终转发到 /adminjsps/admin/book/list.jsp 页面；
- 修改 list.jsp，显示图书列表。

3. 按作者、出版社查询

- 修改 list.jsp 上的链接，让其请求 AdminBookServlet 对应的方法；
- 参数 BookServlet 对应方法完成，最后转发到 /adminjsps/admin/book/list.jsp 页面；

4. 高级搜索

- 修改 gj.jsp 上的表单，让其请求 AdminBookServlet 的 findByCombination() 方法；
- 该方法参照 BookServlet 实现，最后转发到 /adminjsps/admin/book/list.jsp 页面；

4.2 加载图书

在 list.jsp 上点击图书的图片、图书名，请求 AdminBookServlet#load() 方法，load() 方法完成加载图书，转发到 desc.jsp 显示。

desc.jsp 页面不仅可以显示图书详细信息，而且还可以作为编辑图书的表单。该页面上有一个“编辑或删除”复选框，如果勾选了它，那么 desc.jsp 页面上就显示表单，用来作为编辑图书而用；如果没有勾选 desc.jsp 只是显示图书详细信息。

☒ 编辑或删除

	<div>商品编号: 5C68141786B84A4CB8929A2415040739</div> <div>书名: JavaScript高级程序设计(第3版)JavaScript技术名著, 国内JavaScript第一书, 销量超10万册</div> <div>当前价: 68.3</div> <div>定价: 99.0 折扣: 6.9 折</div> <hr/> <div>作者: Nicholas C. Zakas</div> <div>出版社: 人民邮电出版社</div> <div>出版时间: 2012-3-1</div> <div>版次: 1</div> <div>页数: 730</div> <div>字数: 100000</div> <div>印刷时间: 2012-3-1</div> <div>开本: 16</div> <div>纸张: 胶</div> <div>一级分类: 程序设计</div> <div>二级分类: Java Javascript</div>
---	--

因为 desc.jsp 不仅用来显示图书详细, 而且还用来作为编辑图书的表单页面, 默认 desc.jsp 页面使用非表单形式显示图书。

当 desc.jsp 要显示表单时, 还要允许用户修改图书的分类, 所以 load()方法还要传递所有 1 级分类, 以及当前图书所有 1 级分类下的所有 2 级分类。并且, 用户在修改了 1 级分类后, 页面还要向服务器发出异步请求, 请求当前 1 级分类下的所有 2 级分类, 然后显示在下拉列表中。

desc.jsp 页面:

- 两种显示 book 方式: 一是非表单显示, 二是表单显示;
- javascript:
 - 给“编辑或删除”复选框添加事件, 未勾选时使用非表单显示; 勾选时使用表单显示;
 - 给 1 级分类下拉列表添加 onchange 事件, 异步请求当前 1 级分类下所有 2 级分类, 然后显示在 2 级分类下拉列表中。
 - 因为 desc.jsp 页面的表单有两个提交按钮 (编辑和删除), 所以需要为这两个提交按钮添加事件, 为表单添加隐藏字段 method 的值, 然后提交表单。

AdminBookServlet#load()方法:

- 获取 bid 加载图书, 保存到 request 中;
- 获取所有 1 级分类, 保存到 request 中;
- 获取当前 book 所属 1 级分类下的所有 2 级分类, 保存到 request 中;
- 转发到 desc.jsp。

load()

```
public String load(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {  
    /*  
     * 1. 获取bid,通过bid加载book, 保存到request中  
     */  
    String bid = request.getParameter("bid");  
    Book book = bookService.load(bid);  
    request.setAttribute("book", book);  
    /*  
     * 2. 获取所有1级分类, 保存到request中  
     */  
}
```



```

        */
        request.setAttribute("parents",
categoryService.findParents());
    /*
     * 3. 获取当前图书所属1级分类下的所有2级分类
     */
    // 获取当前图书所属1级分类cid
    String pid = book.getCategory().getParent().getCid();
    request.setAttribute("children",
categoryService.findChildren(pid));
    return "/adminjsps/admin/book/desc.jsp";
}

```

批注 [q1]: 查询所有 1 级分类

批注 [q2]: 查询指定 1 级分类下所有 2 级分类

desc.jsp 需要显示 1 级、2 级分类的下拉列表。

```

<c:forEach items="${parents }" var="parent">
    <option value="${parent.cid }" <c:if
test="${book.category.parent.cid eq
parent.cid }">selected="selected"</c:if>>${parent.cname }</option>
</c:forEach>
<c:forEach items="${children }" var="child">
    <option value="${child.cid }" <c:if test="${book.category.cid eq
child.cid }">selected="selected"</c:if>>${child.cname }</option>
</c:forEach>

```

desc.jsp 需要添加异步请求，在 1 级分类修改后，请求新的 1 级分类下所有 2 级分类，并显示在 2 级分类的下拉列表中。

```

/*
给1级分类下拉列表添加改变事件
在重新选择了1级分类后，加载该1级分类的所有2级分类并显示在2级分类下拉列表中
*/
$("#pid").change(function() {
    var pid = $(this).val();//获取新的1级分类id
    $("#cid").empty();//清空2级分类下拉列表
    // 添加头选项
    $("#cid").append($("#<option value=''>==请选择2级分类
==</option>"));
    if(pid) {//如果新的1级分类不为空!
        $.ajax({
            url:"<c:url value='/admin/AdminCategoryServlet'/>",
            data:{method:"childrenForAjax", pid:pid},
            dataType:"json",
            type:"POST",
            async:"false",
            cache:"false",

```

```

        success:function(children) {
            // 循环遍历所有2级分类
            for(var i = 0; i < children.length; i++) {
                // 创建选项同
                var option = $("<option>").attr("value",
children[i].cid).text(children[i].cname);
                // 向2级分类下拉列表中添加选项同
                $("#cid").append(option);
            }
        }
    });
}

```

4.3 编辑图书

在 desc.jsp 页面点击“编辑图书”按钮完成编辑图书，该链接请求 AdminBookServlet 的 edit()方法。

edit()方法封装表单数据，调用 BookService#edit(Book)方法完成编辑图书。然后保存“编辑成功”到 request 中，转发到/adminjsps/msg.jsp 显示。

注意，编辑图书功能不支付编辑图书图片，所以在 BookDao 中的 update 语句不能修改 image_b 和 image_w 两列。

4.4 删除图书

在 desc.jsp 页面点击“删除图书”按钮完成删除图书，该链接请求 AdminBookServlet 的 delete()方法。

delete()调用 BookService#delete(bid)方法，完成删除图书。然后保存“删除成功”到 request 中，转发到/adminjsps/msg.jsp 显示。

4.5 添加图书

添加图书分为两步：

- 点击 body.jsp 上的“添加图书”链接，到达 add.jsp;
- 在 add.jsp 提交表单。

1. 点击“添加图书”链接，到达 add.jsp

点击“添加图书”链接，请求 AdminBookServlet#addPre()方法，addPre()获取所有 1 级分类，然后转发到 add.jsp 页面。

add.jsp 显示所有 1 级分类到下拉列表中，当用户显示 1 级后，需要发送异步请求，获取该 1 级分类下所有 2 级分类，然后显示到 2 级分类下拉列表中。（与编辑图书一样）

2. 在 add.jsp 提交表单，完成添加图书。

添加图书需要上传图片，这一功能不能放到 AdminBookServlet，而应该为其单独提供一

个 Servlet: AdminAddBookServlet。

在 AdminAddBookServlet 中需要使用 commons-fileupload 组件解析表单。获取所有表单项,封装到 Book 对象中,然后调用 BookService#add(Book)方法完成添加,最终转发到 msg.jsp 显示成功信息。

注意,要对上传的图片扩展名进行校验!然后保存图片到 book_img 目录下!

5 订单管理

在 top.jsp 上点击“订单管理”查看所有订单。

ITCAST网络图书商城后台管理			
管理员: 退出 订单管理 图书管理 订单管理			
未付款 已付款 已发货 交易成功 已取消			
商品信息	金额	订单状态	操作
订单号: E3A1EB6D0543489F9729B2B5BC5DB365 下单时间: 2013-06-01 19:30:22			
	¥203.5	等待付款	查看 取消 发货
订单号: E3A1EB6D0543489F9729B2B5BC5DB365 下单时间: 2013-06-01 19:30:22			
	¥203.5	等待付款	查看 取消 发货
订单号: E3A1EB6D0543489F9729B2B5BC5DB365 下单时间: 2013-06-01 19:30:22			

订单管理功能如下:

- 查看所有订单;
- 查看未付款订单;
- 查看未发货订单;
- 查看未确认收货订单;
- 查看交易成功订单;
- 查看已取消订单;
- 查看订单详细信息;
- 取消未付款订单;
- 订单发货。

后台订单功能与前台都是相同的,可参照前台完成。

5.1 与前台相同功能实现

1. 查看所有订单(分页显示)

这个功能前台没有,但与前台的“我的订单”很相似。

- 修改 top.jsp 中“订单管理”链接,请求 AdminOrderServlet#findAll()方法;
- AdminOrderServlet#findAll()→findAll()OrderService#findAll()→OrderDao#findAll()方法都参照前台的“我的订单来完成”来完成。最后转发到/adminjsps/admin/order/list.jsp 页面。
- 修改 list.jsp 页面,不同状态显示不同链接。

2. 按状态查询订单(分页显示)

这个功能前台没有，但与前台的“我的订单”很相似。

- 修改 `/adminjsps/admin/order/list.jsp` 上 5 个按状态查询的链接，请求 `AdminOrderServlet#findByStatus()` 方法，参数订单状态。最后转发到 `/adminjsps/admin/order/list.jsp` 页面。
3. 查看订单详细信息
- 在 `list.jsp` 页面点击订单编号链接，或点击查看链接，请求 `AdminOrderServlet#load()` 方法，传递订单 `oid`，以及 `oper=desc`。
 - `load()` 方法加载订单，保存到 `request` 中，把 `oper` 也保存到 `request` 中，转发到 `desc.jsp` 页面。
 - `desc.jsp` 页面显示订单信息，并根据 `oper` 显示按钮。

5.2 取消订单和订单发货

取消订单和订单发货都是修改订单状态的功能！

- 修改 `desc.jsp` 页面中“取消订单”和“发货”两个链接，请求 `AdminOrderServlet` 的 `cancel()` 和 `deliver()` 方法；
- 调用 `orderService.updateStatus(String oid, int status)` 方法完成对订单状态的修改；
- 保存成功信息到 `request` 中，转发到 `msg.jsp` 页面显示！

6 登录过滤器

添加后台管理员登录过滤器！

- 页面： `/adminjsps/admin/*`；
- Servlet： `/admin/*`；