图书商城

环境搭建

1. 导入原型

* 用户模块
* 分类模块
* 图书模块
* 购物车模块
* 订单模块

2　功能分析

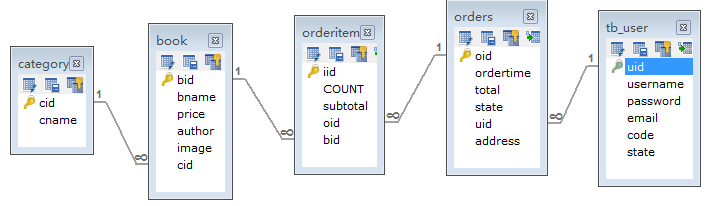
* 前台
* 用户模块：
* 注册
* 激活
* 登录
* 退出
* 分类模块：
* 查看所有分类
* 图书模块：
* 查询所有图书
* 按分类查询图书
* 查询图书详细（按id查）
* 购物车模块：
* 添加购物车条目；
* 清空所有条目；
* 删除指定条目；
* 我的购物车（按用户查询购物车）
* 订单模块：
* 生成订单；
* 我的订单（按用户查询订单）
* 按id查询订单
* 确认收货
* 付款功能（只是跳转到银行页面）
* 付款回调功能（由银行来调用我们这个方法，表示用户已经付款成功）
* 后台：
* 管理员：
* 登录
* 分类管理：
* 添加分类
* 查看所有分类
* 删除分类
* 按id查询
* 修改分类
* 图书管理（我的）
* 查看所有图书
* 按id查询
* 删除图书
* 修改图书
* 添加图书（上传图片）
* 订单模块
* 查询所有订单
* 按状态查询订单
* 发货

3　框架的搭建

* 1. 导包
* 数据库：
* mysql驱动
* c3p0（jar包和配置文件）
* dbutils
* lanou-tools
* commons-beanutils
* commons-logging
* javamail
* mail.jar
* activation.jar
* 上传
* commons-fileupload
* commons-io
* ajax
* json-lib
  1. 创建package
* 根：com.lanou.bookstore
* user
* domain
* dao
* service
* web.servlet
* category
* domain
* dao
* service
* web.servlet
* book
* domain
* dao
* service
* web.servlet
* cart
* domain
* web.servlet
* order
* domain
* dao
* service
* web.servlet

3.3 表

* tb\_user
* category
* book
* orders
* orderitem



用户模块

1　用户模块的相关类创建

* com.lanou.bookstore.user
  + domain：User
  + dao：UserDao
  + service：UserDao
  + web.servlet：UserServlet

2　用户注册

流程：/jsps/user/regist.jsp 🡪 UserServlet#regist() 🡪 msg.jsp

页面：

* regist.jsp
* 表面页面，请求UserServlet#regist()方法
* 参数：整个表单数据
* msg.jsp

Servlet：

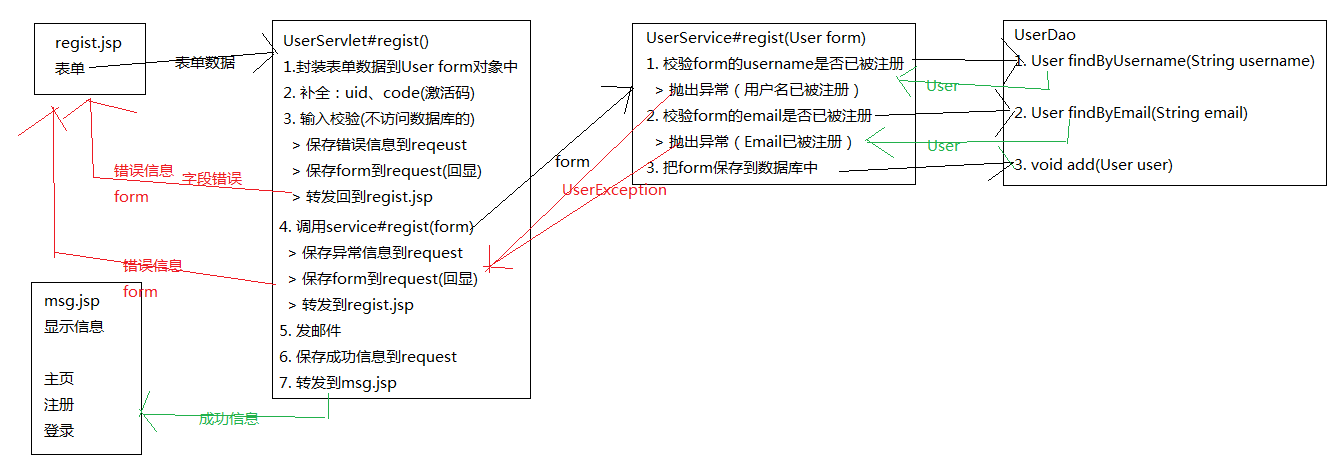
* UserServlet#regist()
* 一键封装表单数据到User form对象
* 补全：uuid、激活码
* 输入校验：
* 保存错误信息到request
* 保存当前表单数据（form）到request（回显）；
* 转发回到regist.jsp
* 调用service的regist()方法，传递form过去；
* 如果抛出异常：保存错误信息（异常信息）、保存表单数据（回显）、转发到regist.jsp
* 如果没有抛出异常：
* 发邮件（发件人、收件人、标题、内容），内容中包含超链接，超链接指向可完成激活的Servlet地址！链接中要有激活码参数！
* 保存成功信息到request中
* 转发到msg.jsp

Service：

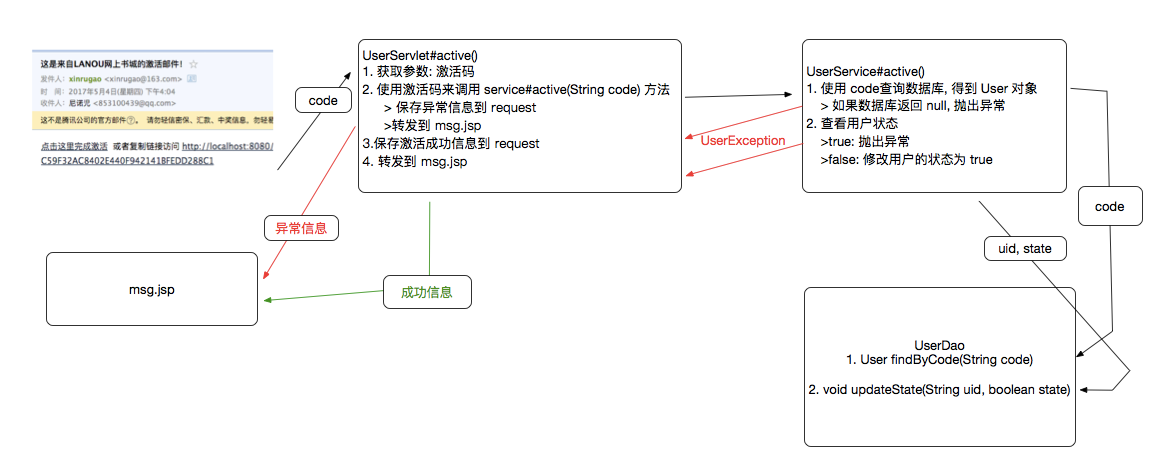
* UserService#regist(User form)
* 校验用户是否被注册，如果注册，抛出UserException；
* 校验邮箱是否被注册，如果注册，抛出UserException；
* 把user插入到数据库中

Dao：

* User findByUsername(String username)：按用户名查询用户
* User findByEmail(String email)：按emal查询用户
* void add(User form)：插入用户到数据库中

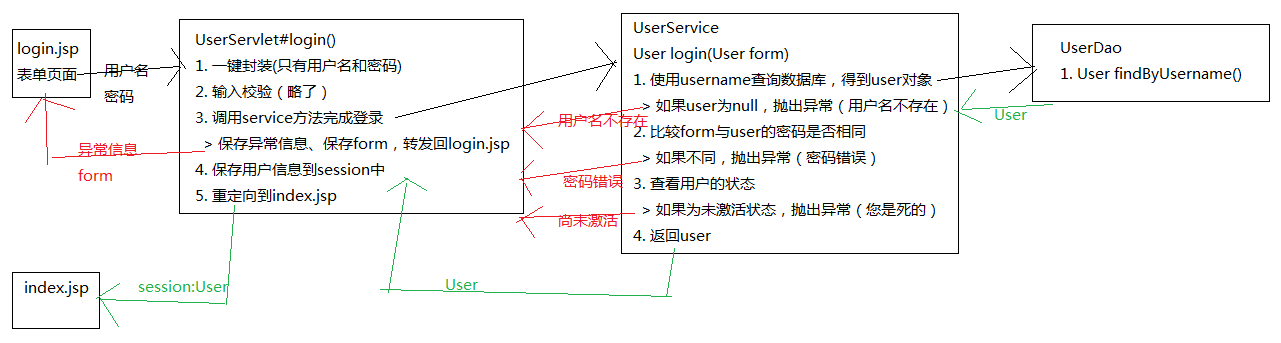


3　用户激活

流程：用户的邮件中 🡪 UserServlet#active() 🡪 msg.jsp

4　用户登录

流程：/jsps/user/login.jsp 🡪 UserServlet#login() 🡪 index.jsp



5　用户退出

流程：top.jsp 🡪 UserServlet#quit() 🡪 login.jsp

quit()：把session销毁！

分类模块

1　分类模块的相关类创建

* com.lanou.bookstore.category
* domain：Category
* dao：CategoryDao
* service：CategoryService
* web.servlet：CategoryServlet

2　查询所有分类

　　流程：main.jsp（<iframe>） 🡪 CategoryService#findAll() 🡪 left.jsp

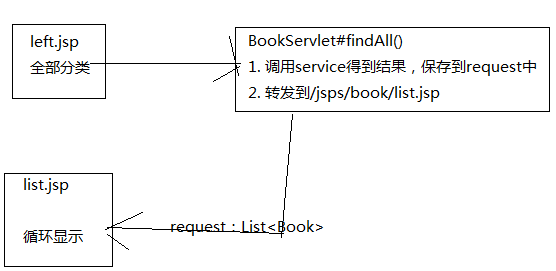
图书模块

1　创建相关类

* com.lanou.bookstore.book
* domain：Book
* dao：BookDao
* service :BookService
* web.servle：BookServlet

2　查询所有图书

流程：left.jsp（全部分类） 🡪 BookServlet#findAll() 🡪 /jsps/book/list.jsp



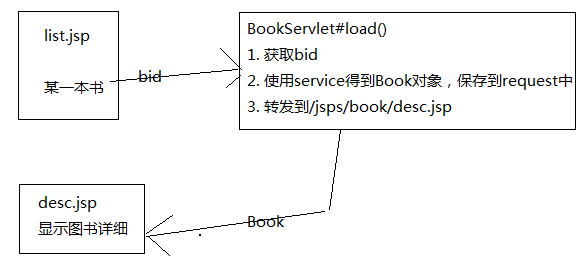
3　按分类查询图书

流程：left.jsp 🡪 BookServlet#findByCategory() 🡪 list.jsp



4　查询详细信息（加载）

　　流程：list.jsp（点击某一本书） 🡪 BookServlet#load() 🡪 desc.jsp



购物车模块

购物车存储：

* **保存在session中**；
* 保存在cookie中；
* 保存在数据库中。

1　创建相关类

* 购物车的结构：
* CartItem：包含图书和数量
* Cart：包含一个Map<String,CartItem>
* dao：没有
* service：没有
* web.servlet：提供！CartServlet

　　修改登录方法，在用户登录成功后，马上在session中添加一辆车！！！

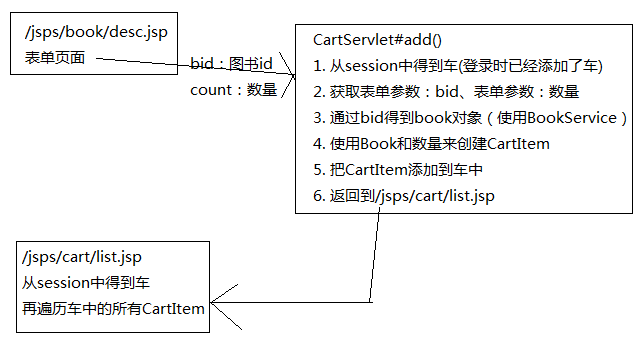
页面：/jsps/cart/list.jsp

它只有一个任务，就是遍历车！

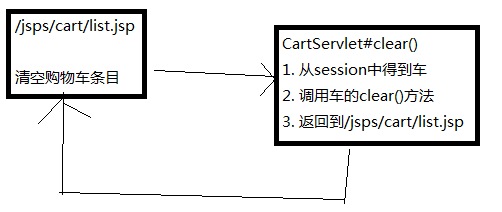
车在session中，通过车可以得到所有的CartItem

${sessionScope.cart.cartItems}

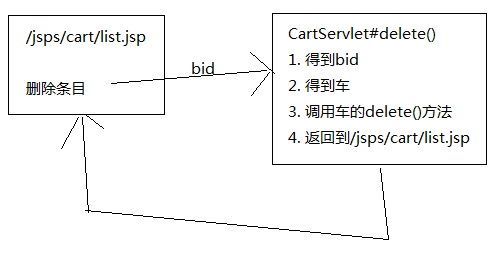
2　添加购物车条目



3　清空条目



4　删除购物车条目



5　我的购物车

top.jsp中存在一个链接：我的购物车

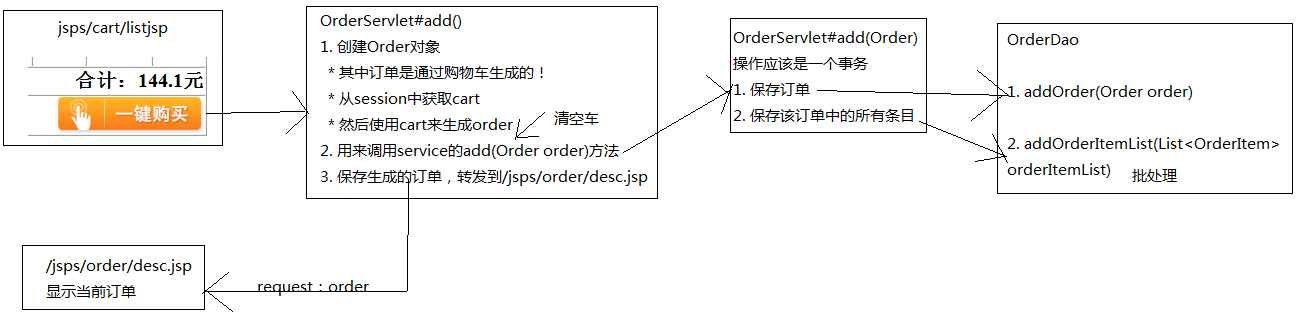
我的购物车直接访问/jsps/cart/list.jsp，它会显示session中车的所有条目

订单模块

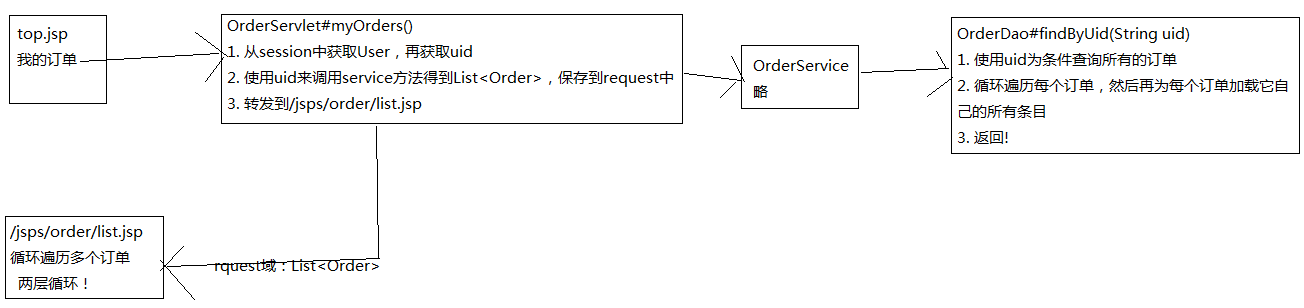
1　创建相关类

* com.lanou.bookstore.order
* domain：
* Order
* OrderItem
* dao：OrderDao
* service：OrderService
* web.servlete：OrderServlet

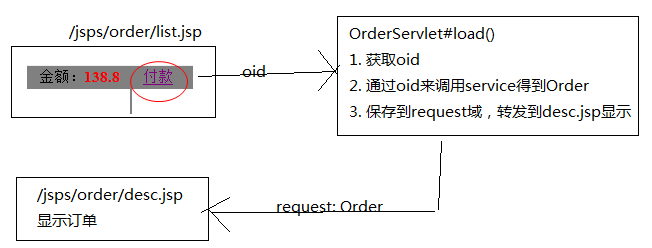
2　生成订单



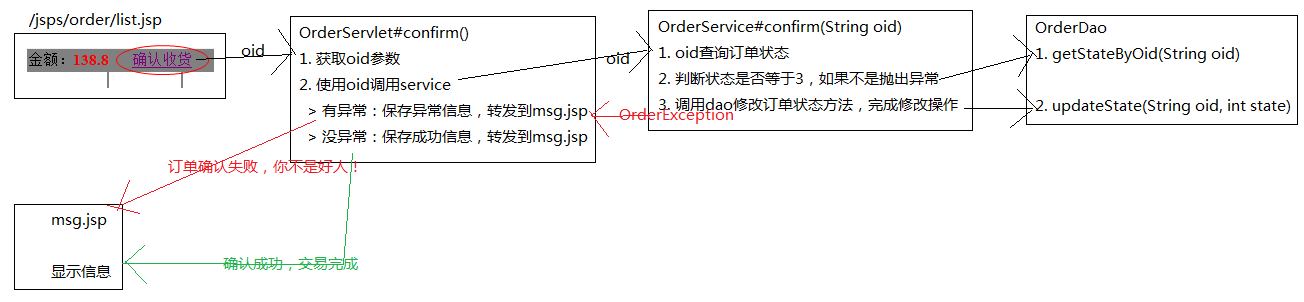
3　我的订单（按用户查）



4　加载订单（按id查）



5　确认收货



支付

在线支付的两种形式：

1. 电商与银行直连！

* 安全
* 不收手续费
* 不与小电商合作！

1. 第三方支付平台

* 支付宝
* 易宝
* 财富通

特点：

* 不安全
* 收手续费（1%）
* 小电商可以与其合作！

需要在第三方注册账户

* 需要认证！

后台

后台的内容，必须要设置权限！

用户可以访问一个网站的哪些内容？

* dao：不行
* service：不行
* servlet：能！
* jsp：能！

　　用户可以访问的只有WEB层！

1　分类管理

* 分类管理：
* 添加分类
* 查看所有分类
* 删除分类
* 按id查询
* 修改分类

1.1　相关类创建

* com.lanou.bookstore.category
  + domain：Category
  + dao：CategoryDao
  + service：CategoryService
  + admin.web.servlet：AdminCategoryServlet（为管理员提供单独的Servlet，然后给这个Servlet添加过滤器！）

1.2　查看所有分类

* left.jsp上的菜单修改一下指向：AdminCategoryServlet#findAll()
* findAll()：
* 调用service得到所有的分类List<Category>
* 保存到request域，转发到/adminjsps/admin/category/list.jsp
* list.jsp：修改页面，显示所有分类！

1.3　添加分类

* add.jsp（表单页面）
* AdminCatetgoryServlet#add()：
* 封装表单数据；
* 补全：cid
* 调用service方法完成添加工作
* 调用findAll()方法
* service#add(Category c)：略
* dao#add(Catetgory c)：略

1.4　删除分类

* list.jsp（删除链接）
* AdminCategoryServlet#delete()
* 获取参数：cid
* 调用service方法完成删除！
* 如果出现异常，保存异常信息，转发到msg.jsp显示
* 调用findAll()
* service#delete(String cid)：
* 通过cid查看该分类下的图书本数，如果大于0，抛出异常；
* 如果等于0，删除该分类；

1.5　修改分类

修改分为两步：1、加载分类，　2、修改分类

第一步：加载分类

* list.jsp（修改链接） 🡪
* AdminCategoryServlet#editPre()
* 获取cid
* 通过cid来调用service方法，得到Category对象
* 保存到request域中，转发到mod.jsp
* mod.jsp：把当前的Category对象显示到表单中

第二步：修改分类

* mod.jsp（提交表单）
* AdminCategoryServlet#edit()
* 封装表单数据
* 调用service方法完成修改工作
* 调用findAll()

2　图书管理

* 图书管理（我的）
* 查看所有图书
* 按id查询
* 删除图书
* 修改图书
* 添加图书（上传图片）

2.1　相关类创建

* web.servlet.admin：
* AdminBookServlet；
* AdminAddBookServlet（添加图书，包含上传）：上传不能使用BaseServlet，因为BaseServlet中需要使用getParameter()方法，而上传getParameter()方法就不能再使用了。

2.2　查询所有图书

* left.jsp（菜单项（查看图书链接））
* AdminBookServlet#findAll()
* 查询所有图书，保存到request
* 转发到/adminjsps/admin/book/list.jsp
* list.jsp：循环遍历所有图书

2.3　加载图书

* list.jsp（点击图名或图片） 🡪
* AdminBookServlet#load()
* 获取bid，通过bid调用BookService方法得到Book对象
* 保存到request中，转发到/adminjsps/admin/book/desc.jsp
* desc.jsp：把当前book对象显示到表单中

2.4　添加图书

添加图书分两步：

1. 加载所有分类，到add.jsp中显示！

* left.jsp（菜单项：添加图书）
* AdminBookServlet#addPre()：
* 查询所有分类，保存到request域，转发到add.jsp
* 在add.jsp中循环遍历所有分类，显示在<select>中

1. 添加图书

* 上传三步：
* 创建工厂
* 创建解析器
* 解析request得到表单字段！
* 把表单字段封装到Book对象中
* 保存上传文件，把保存的路径设置给Book的image属性。
* 调用service方法保存Book对象到数据库中
* 调用findAll()

2.5　删除图书

book表与orderitem有关联关系！

删除图书不是真的数据库表中删除记录，而是给book表添加一个del字段，它是booleanod类型，表示是否已删除！

没有被删除的图书，该列的值为false，否则为true

处理问题：

* 修改BookDao：所有与查询相关的方法，都需要添加where条件，即del=false
* 修改Book类，添加del属性！

删除图书：其实就是把表的del列修改为true！

* desc.jsp（del按钮）
* AdminBookServlet#del()
* 获取bid
* 调用service方法完成删除
* 返回列表，即调用findAll()

2.6　编辑图书

* dsec.jsp（编辑按钮）
* AdminBookServlet#edit()
* 封装表单数据（必须让页面保证把image传递过来）
* 要求页面必须添加一个隐藏字段，把原来的图片路径传递过来！
* 调用service方法完成删除
* return findAll()