

WEB 开发技术-BookStore-迭代三开发文档

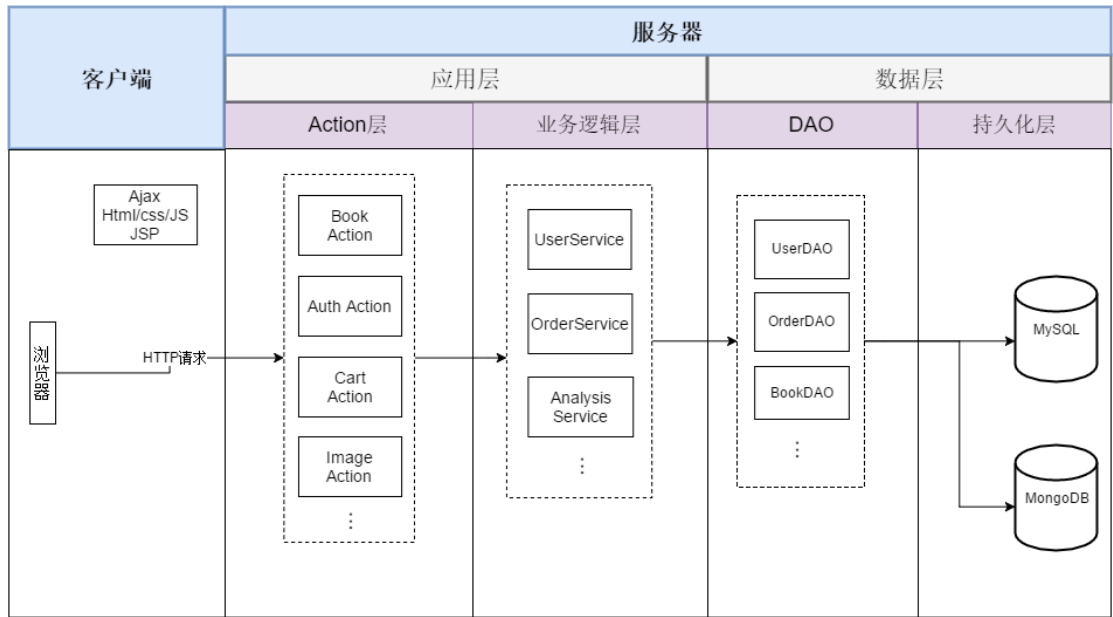
F1403703 5140379066 肖煜伟

一、概述

开发主要使用了 ssh、html\css\bootstrap\less、js\jquery\easyui\angular\d3js 等语言和框架，数据库使用了 MySQL 和 MongoDB，实现了书店后台管理员对 book\user 的增删改查，对 order 的删除，对销售额分类统计（年、月、日）、对按书类别进行销售量统计、对销售额按给定日期统计以及对特定用户的消费行为进行分析。书店支持权限控制（登陆登出注册、用户角色）、浏览商品、查找商品、购物车、支付订单、用户信息查看及修改等功能。网站后端使用了 MVC 框架、ORM 映射、IOC 注入、AOP，且做到一定程度的前后端分离，具有较好可维护性。

二、设计说明

1、逻辑架构



1.1 Struts

- 精细化控制 struts-json 序列化提高性能
- 使用拦截器实现权限控制
- 利用 Package\namespace 实现结构清晰的转发逻辑

1.2 Hibernate

- 使用 Lazy 保证性能
- 使用 OpenSessionInViewFilter 解决 lazy 导致的 view 内无法加载元素

1.3 spring

a) IOC

- 为 action 对象注入 service
- 为 service 对象注入 DAO
- 为 DAO 对象注入 sessionFactory
- 注入 MongoDB 配置

b) AOP

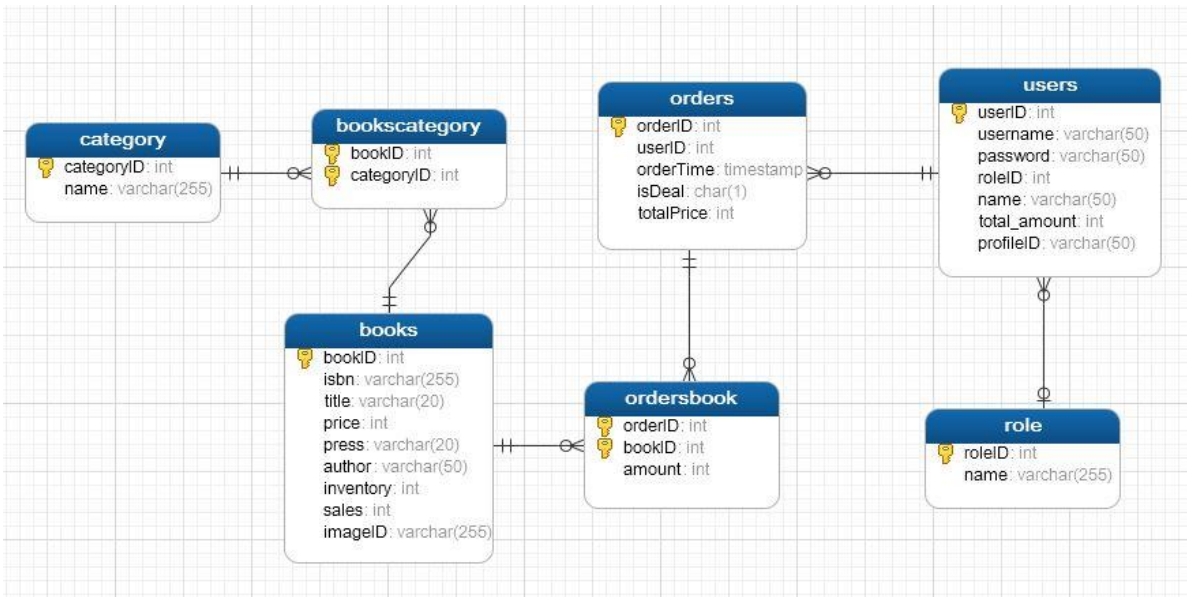
- service 层管理数据库操作的事务逻辑，设置读写权限、隔离级别保证安全性和性能。

2、 功能特点

- a) 注册、登陆、登出
- b) 对书的搜索、列表查看、详情查看
- c) 实现了购物车逻辑，使用 session 保存购物车，并且支持对购物内的物品进行修改（增加减少数量、删除物品）
- d) 实现了订单逻辑，包含支付流程。订单支持多种状态，如未支付、已支付、已关闭。
- e) 用户对个人信息的修改
- f) 管理员后台对 book\user 的增删改查，对 order 的删除
- g) 实现了对销售额的统计，分别可以按年月日查看，并且使用了 D3JS 库进行可视化呈现。
- h) 实现了指定日期区间对销售额进行统计，并实现了可视化。
- i) 实现了按书的类别对销售量进行统计，将统计结果进行了可视化。
- j) 实现了对指定用户的消费行为分析，并将数据进行了可视化。
- k) 实现了对每一本书本销量的统计呈现。
- l) 实现了精密的权限控制。普通用户无法进入后台以及各类敏感 API 的访问控制。

三、数据库设计

1. 数据库 schema



数据库结构主要分为三个模块：用户、书籍、订单。

- 用户（users）由一张表维护，同时维持一个角色（role）表，二者为多对一关系。同时用户表有属性 profileID，为 mongodb 中存储用户详细信息的文档。
- 书籍（books）由一张表维护，同时和类目（category）维持多对多关系，即一本书可能属于多个目录，比如《WEB 开发》这本书既归类于教育也归类于计算机技术。在数据库中，将多对多关系拆分为一对多、多对一，维护一个中间表（bookcategory）。书的图像由 MongoDB 保存，MySQL 表中维护图片在 MongoDB 中的索引 imageID。
- 订单（orders）由一张表维护，与用户（users）表有多对一关系，和书籍（books）类有多对多关系，在数据库中拆分为中间表（ordersbook），中间表记录购买书籍的本数。

2. 数据库存储过程

所有数据统计从应用层抽离写成数据库内的函数(存储过程)，提高查询性能。同时，考虑到统计操作设计数据规模较大，故在应用层的事务管理时，设置隔离级别为 READ_UNCOMMITTED，减少对其他数据库操作的影响。具体存储过程如下所示：

- `getSalesWithLimitsBy(format varchar(12), beginTime varchar(12), endTime varchar(12))`
在指定时间范围内以指定的时间格式获取销售量

- `getSoldAmountByCategory()`
获得各个分类的销售量
- `getTotalPriceBy(format varchar(12))`
按指定的时间格式获取销售量
- `getUserBehaviorOfBookCategory(user_id int)`
获得指定用户所购买书类目统计
- `getUserRecentSales(user_id int)`
获得指定用户最近一星期的消费额

3.Trigger

- `updateBook`
在订单创建的时候触发，自动更新 `books` 表内对应书籍的销售量和库存
- `deleteOrdersBook`
未支付订单被删除的时候，自动更新 `books` 表内对应书籍的销售量和库存

4.完整性设计

- `books`
库存 `inventory`、销售量 `sales`、价格 `price` 的取值非负；
关键属性非空（`isbn`、`price`、`inventory`、`title`）
- `Orders`
关键属性非空（`orderTime`、`isDeal`）
- `Users`
关键属性非空（`username`、`password`、`roleID`）

4.索引设计

除主键及外键自动添加的索引外，数据库设计中根据需求额外设计的若干索引，以增加性能：

- `category` 表中以 `name` 建立索引
- `users` 表中以 `username` 建立索引
- `role` 表中以 `name` 建立索引

5.MongoDB 使用

- 存储用户详细信息

- 使用 gridFS 存储用户头像、书籍封面图片

四、 本地部署配置

方法一：

- 1、把 bookstore.sql 导入数据库
- 2、用 IDE 打开工程，使用 pom.xml 下载依赖
- 3、配置 hibernate.cfg.xml 内数据库连接属性（username\password\url）
- 4、使用 Tomcat 部署
- 5、通过 Localhost: (port)/bookstoreiter3 访问

注：

默认普通账户： zhangsan 密码： 123456

默认管理员账户： admin 密码： 123456

五、 项目约束条件

- 1、数据库使用 Mysql5.6.24
- 2、Java version 1.8.0_25
- 3、Tomcat v8.0