

基于 SpringBoot 的小型日常交流论坛的设计与实现

李忠毅

(中山大学新华学院信息科学学院, 东莞 523133)

摘要:

随着互联网的发展和 5G 的到来,网络变得越来越重要,人们除了日常交流外,使用博客、论坛等平台的需求也日益增多。为了方便地通过网络记录生活和信息交流,降低搭建网站的门槛,适宜使用 SpringBoot 后端技术和 Bootstrap 前端框架进行小型日常交流论坛搭建。为此介绍小型交流论坛的需求、系统设计、数据库设计和系统测试,并利用 SpringBoot 后端技术实现系统的功能。

关键词:

SpringBoot; Bootstrap; 交流论坛

0 引言

过去的日常交流论或者博客,后端大多使用 JSP、SSH、SSM 技术进行实现,基于这几种技术实现,无一例外都会遇到大问题:大量的.xml 文件使人眼花缭乱,对于建立个人论坛博客有极大的阻碍,如果使用 Spring-Boot 进行后端技术,可以做到零个.xml 文件,极大地降低了项目的复杂性,前端技术使用 Bootstrap 和 Thymeleaf,实现前后端分离,MVC 架构降低了前后端的耦合,是整个系统拥有高内聚,低耦合的特性,Bootstrap 作为响应性布局框架,使个人博客可以跨平台使用,实现一套代码适配多个平台。

1 开发技术与工具

开发环境:JDK 1.8, Tomcat 服务器。

开发工具:后端开发:IDEA 2020.1,前端开发:WebStrom,数据库:MySQL 5.7,数据库可视化软件:Navicate。

开发技术:SpringBoot 后端技术、MyBatis、Bootstrap。

2 系统设计

2.1 模块设计

小型日常交流论坛的模块有三个:登录注册模块、

用户操作模块、管理员模块。

(1)用户登录注册模块

用户可以在网页进行登录注册功能,拥有登录状态的用户才可以浏览帖子详情,发新帖子,对帖子进行回复,未登录用户只能在主页浏览帖子的标题和作者。

(2)用户的操作模块

已经登录的用户可以查看帖子,点赞帖子,搜索帖子,回复帖子,回复别人的回复,点赞回复,修改和删除自己的帖子。此外还可以进行修改密码,修改昵称,修改个人信息,上传头像的操作。

(3)管理员模块

管理员登录后可以直接对所有的帖子进行删除和修改操作,对所有账号进行管理。

2.2 数据库设计

数据库设计在动态网站的设计中占有非常重要的地位,数据库结构设计的好坏会直接影响到应用程序的运行效率。合理的数据库结构设计能够提高数据读取与存储的效率,能够保证数据的一致性和完整性。该系统的数据库使用 MySQL 来设计和实现^[2]。

小型日常交流论坛需要用到 4 个数据表:用户表、帖子表、回复表、二次回复表,数据库 ER 图如图 1 所示。

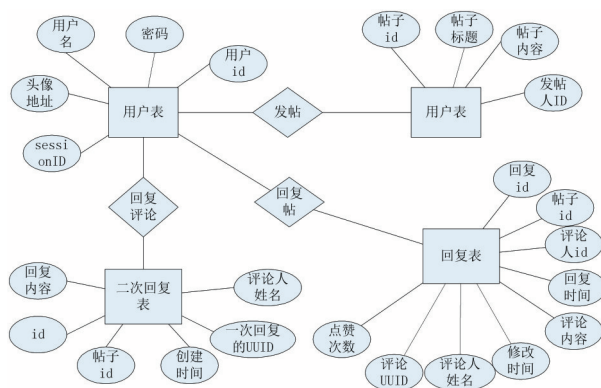


图1 数据库的ER图

(1)用户表用于存放用户 id,用户名,用户密码,头像地址,sessionID,其中头像地址为空时(用户未上传头像)使用默认头像,sessionID 用于保持登录状态。

(2)帖子表用于存放帖子 id,帖子标题,帖子内容,发帖人 ID。

(3)回复表用于存放帖子 id,回复 id,评论人 id,回复时间,评论内容,修改时间,评论人姓名,评论的随机 id,点赞次数,其中重要的是平人的随机 id,给每一条评论赋予唯一标识。

(4)二次回复表存放二次回复 id,回复内容,帖子 id,创建时间,一级回复的随机 id,评论人姓名。

3 系统实现

系统采用 MVC 结构进行实现,对视图层、模型层、控制层进行了分离,实现高内聚低耦合的特点。其中,视图层使用 Bootstrap 进行设计和实现,其跨平台的特性能够做到一套代码多个平台使用,极大减轻了开发者的负担。模型层也是数据访问层,进行对数据库的查找、添加、修改和删除,实现数据持久化操作。控制层实现了业务逻辑,对模型层进行控制,实现用户的需求。

3.1 发帖功能模块

发帖功能是一个小型日常交流论坛的核心,用户在登录后即可获得发帖功能。

发帖功能的代码实现如下所示:

```
@RequestParam(value = "content") String content,
@RequestParam(value = "id") Integer id,
@RequestParam(value = "username") String username,
Model model) {
    model.addAttribute("title", title);
    model.addAttribute("content", content);
    /* 内容校验 */
    if (title == null || title == "") {
        model.addAttribute("noTitle", "标题不能为空!");
        return "publish";
    }
    if (content == null || content == "") {
        model.addAttribute("noContent", "内容不能为空!");
        return "publish";
    }
    User publish_user = null;
    //放入时间
    Date date = new Date();
    String nowTime = new SimpleDateFormat("yyyy-MM-dd HH:mm:ss").format(date);
    //放入内容
    Tie tie = new Tie();
    tie.setTitle(title);
    tie.setContent(content);
    tie.setUsername(username);
    tie.setCreatorId(id);
    tie.setDate(nowTime);
    /* 数据持久化操作 */
    tieMapper.addTie(tie);
    return "redirect/";
}
```

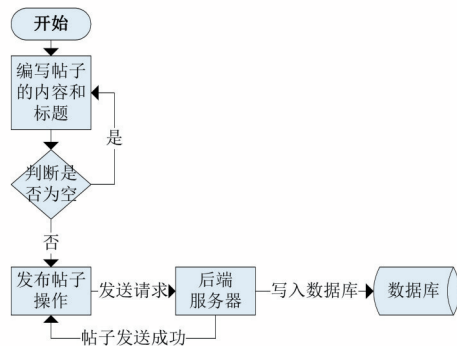


图2 发帖功能流程图

3.2 展示帖子模块

展示帖子也是论坛的核心功能,后端通过 JDBC 操作数据库获取数据并实现分页功能,通过 model 方法给前端返回数据,前端对数据进行循环遍历,展示到主页面上供用户查看,核心代码段如下:

```
// 用来首页分页展示帖子
public PageDTO list(Integer page, Integer size) {
    PageDTO pageDTO = new PageDTO();
    Integer totalCount = tieMapper.count();
    pageDTO.setPageNation(totalCount, page, size);
    if (page < 1) {
        page = 1;
    }
    if (page > pageDTO.getTotalPage()) {
        page = pageDTO.getTotalPage();
    }
    //size*(page-1)
    Integer offset = size * (page - 1);
    list<Tie>
    List<Tie> ties = tieMapper.list(offset,size);
    List<TieDTO> tieDTOList = new ArrayList<>();
    for (Tie tie : ties) {
        User user=userMapper.ByTieIdFinUserId(tie.getCreatorId());
        TieDTO tieDTO = new TieDTO();
        BeanUtils.copyProperties(tie,tieDTO);
        tieDTO.setUser(user);
        tieDTOList.add(tieDTO);
    }
    pageDTO.setTie(tieDTOList);
    return pageDTO;
}
```

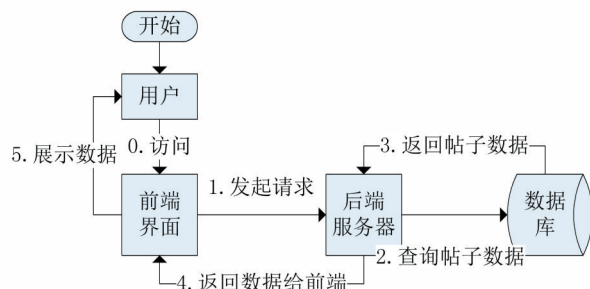


图3 展示帖子流程图

3.3 维持登录状态模块

小型日常交流论坛经常会有用户登录,在 Spring-Boot 中使用拦截器,可以便捷地通过 cookie 保持用户的登录态,减少用户登录次数,方便用户操作,提高了用户体验,下面为核心代码段:

```
@Override
public boolean preHandle (HttpServletRequest request,
    HttpServletResponse response, Object handler) throws Exception {
    Cookie[] cookies = request.getCookies();
    if (cookies != null && cookies.length != 0)
    for (Cookie cookie : cookies) {
        if (cookie.getName().equals("token")) {
            String token = cookie.getValue();
            User user = userMapper.findByToken(token);
            if (user != null) {
                request.getSession().setAttribute("user", user);
            }
            break;
        }
    }
    return true;
}
```

4 系统运行与测试分析

系统功能实现完成之后,需要进行各个模块的测试。不仅是为了保障模块能够正常运行,还需要保障系统符合设计需求。

系统测试:

表 1 测试环境搭建

测试环境	Jdk 1.8, tomcat 服务器
测试设备参数	CPU:酷睿 i5 7 代处理器; 运存: 8G; 内存: 1TB
操作系统	Windows 10
网络环境	2G/3G/4G/Wi-Fi

在表 1 的环境下,进行如下测试,得到表 2。

表 2 系统功能测试

操作名称	具体操作	预期结果	实际结果	判定
登录注册	创建用户名, 密码, 点击注册, 并进行登录操作	跳转主页并拥有登录状态	与预期一致	通过
发帖	登录条件下写入帖子标题和内容, 点击发送	返回发帖成功, 并返回主页, 看到帖子	与预期一致	通过
对帖子进行回复	选择一个帖子, 进行回复, 点击回复按钮	刷新主页后看到自己的回复	与预期一致	通过
上传头像	点击上传头像, 选择照片, 上传操作	返回上传成功并刷新主页, 看见自己的头像	与预期一致	通过
响应式布局	拉动页面到达手机屏幕大小	主页会自动变化	与预期一致	通过

5 结语

本文使用了十分热门的 SpringBoot 后端技术和 Bootstrap 前端技术搭建了小型日常交流论坛。基于这两项技术搭建的论坛界面灵活、代码量少、响应速度

快,极大地提高了程序员的开发效率。通过介绍需求分析、系统设计、数据库设计、核心功能实现,对基于 SpringBoot 技术的平台开发具有一定的参考意义。

参考文献:

- [1]彭勇,陈俞强. 基于 SSM 的实验室管理系统的研究与设计[J]. 东莞职业技术学院计算机工程系,2020:81-83.
[2]康梅娟. 基于 SSM 的摄影爱好者论坛的设计[J]. 临沂大学信息科学与工程学院,2019:6-7.

作者简介:

李忠毅(1998-),男,广东肇庆人,本科,研究方向为 Java 后端开发、响应式前端开发

收稿时间:2020-05-08

修稿时间:2020-06-12

Design and Implementation of Small Daily Communication Forum Based on SpringBoot

LI Zhong-yi

(Information and Science Department, Xinhua College, Sun Yat-Sen University, Dongguan 523133)

Abstract:

With the development of the Internet and the arrival of 5g, the network has become more and more important. Besides daily communication, people have an increasing need to use blogs, forums and other platforms. In order to conveniently record life and information exchange through the network and lower the threshold of website construction, it is appropriate to use SpringBoot back-end technology and bootstrap front-end framework to build small daily communication forums. Introduces the requirements of small exchange forum, system design, data-base design and system test, and uses the technology of SpringBoot backend to realize the function of the system.

Keywords:

SpringBoot; Bootstrap; Communication BBS