个人博客系统设计

摘要：互联网的时代，作为一名开发人员，除了学习新知识以外，还需要总结经验教训，所以有了博客系统的诞生。博客系统可以用来记录自己所学过的知识，然后分享于他人，与志同道合的人相互交换技术。我设计的个人博客系统也是如此，可以将我学到的技术总结成文后再发布到网上，与其他技术人员进行经验交流。我本次设计的个人博客系统采用的是Java语言，前端UI使用的是semantic，后端框架使用的是SpringBoot，jpa，Thymeleaf等框架，数据库使用的是MySQL。

关键词：博客系统；Java；SpringBoot；MySQL；Thymeleaf；Hibernate

目录

[第1章 概述 1](#_Toc42629263)

[1.1 引言 1](#_Toc42629264)

[1.2 设计任务及目标 1](#_Toc42629265)

[1.3 需求分析 1](#_Toc42629266)

[第2章 整体设计 3](#_Toc42629267)

[2.1 环境搭建 3](#_Toc42629268)

[2.2 前端展示 4](#_Toc42629269)

[2.3 后台管理展示页面 8](#_Toc42629270)

[2.4 后台接口及数据库设计 10](#_Toc42629271)

[第3章 详细设计 11](#_Toc42629272)

[3.1 数据库设计 11](#_Toc42629273)

[3.2 Thymeleaf模板配置 12](#_Toc42629274)

[3.3 页面渲染及后台业务逻辑 13](#_Toc42629275)

[第4章 调试与运行 29](#_Toc42629276)

[4.1 后台管理拦截器测试 29](#_Toc42629277)

[4.2 博客新增功能测试 30](#_Toc42629278)

[4.2 测试访客页面渲染情况 31](#_Toc42629279)

[4.3 分类新增和标签新增测试 32](#_Toc42629280)

[4.4 归档渲染测试 33](#_Toc42629281)

[4.5 测试评论功能 34](#_Toc42629282)

[总 结 35](#_Toc42629283)

[参考文献 36](#_Toc42629284)

[附录 37](#_Toc42629285)

[附录1 数据库表E-R图 37](#_Toc42629286)

[附录2 评论树结构 38](#_Toc42629287)

第1章 概述

## 1.1 引言

越来越多的网络用户希望能在网络平台上与别人交流，在传统的WEB1.0时代，无论是论坛、社区还是个人网站，都试图在这些方面进行努力，随着WEB2.0 时代到来，一个新的概念出现了——博客。随着计算机网络的飞速发展，博客已经成为了写网络日志必不可少的一种工具，也是一种简单有效的提供网络用户之间进行在线交流的网络平台，不仅可以交到更多的朋友，还可学习到很多的知识。

本次的软件课程设计研究了一种数据关联规则的个人博客系统，其开发主要包括后台数据库的建立、后台管理及前端展示的Web页面设计。该系统使用的是IDEA开发工具，再结合SpringDataJpa建立数据连接关系，利用其提供的各种组件进行功能开发，并不断改进，直到系统完善。

## 1.2 设计任务及目标

本次设计要求实现一个博客系统网站，与各大主流博客系统网站类似，尽量完善博客系统功能的开发。

Web展示页面简洁大方，满足视觉需求。功能完善， 游客可以自由的搜索、浏览、评论博文，满足人性化需求。

## 1.3 需求分析

1. 游客部分
   1. 能够分页查询所有博文
   2. 能够快速查询博文最多的分类和标签
   3. 能够查询所有博文分类和标签
   4. 能够查询某个分类下的所有博文和某个标签下的所有博文
   5. 能够查询博文的归档信息
   6. 能够查看博主推荐的博文和最新的博文
   7. 能够用关键字进行全局搜索
   8. 能够对单挑博文进行评论留言
   9. 能够对博主的博文进行赞赏
2. 管理员部分
   1. 能够使用账户和密码登录到后台管理
   2. 能够对博文进行分类
   3. 能够给博文添加标签
   4. 能够通过标题和分类查询博客
   5. 能够新增、修改和删除博文
   6. 能够新增、修改和删除博文分类
   7. 能够新增、修改和删除博文标签
   8. 能够设置博文权限：能否浏览、是否推荐、能否评论、能否转载等权限

第2章 整体设计

## 2.1 环境搭建

2.1.1项目搭建环境

* JDK8
* IDEA\_2020.1
* Navicat12.1
* SpringBoot
* Thymeleaf
* Visual Studio Code
* Semantic-UI

2.1.2 项目思维导图

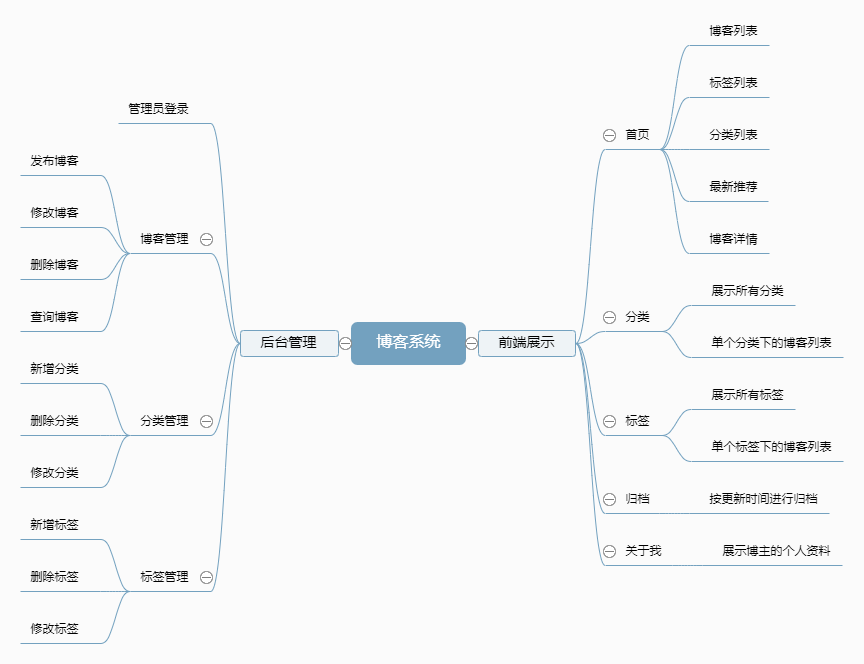


图1项目思维导图

## 2.2 前端展示

前端展示页面是每个来访者都能看到的页面，分为了首页、分类页、标签页、归档页和关于我等5个主要页面，再加上博客详情页面和搜索结果页面一共有7个展示页面。

2.2.1 首页展示

首页是整体展示页面，其中包含有所有博客的列表、分类列表、标签和推荐列表，最终效果如下图所示：

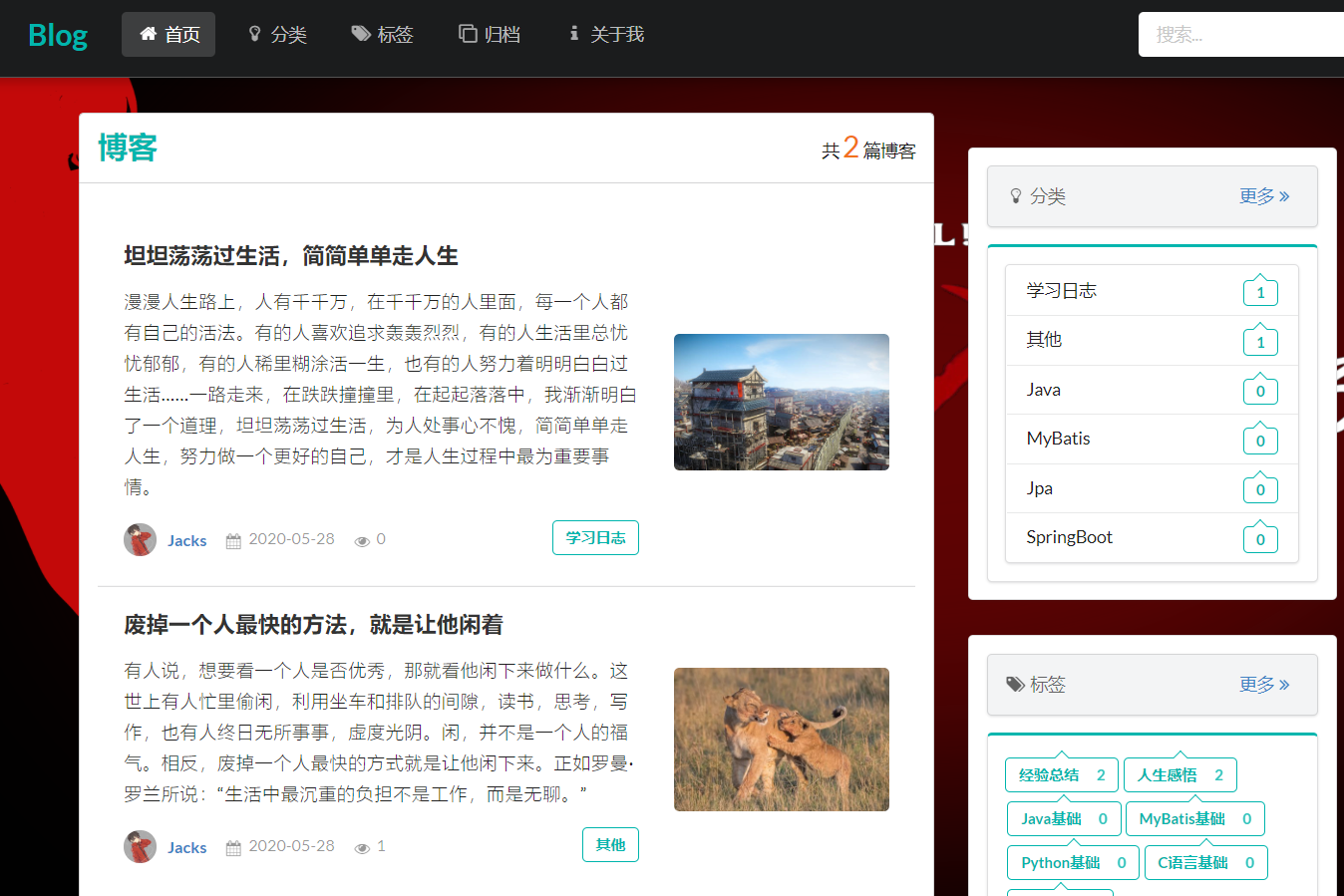


图 2首页展示

2.2.2 分类展示

分类页面共有两部分内容，其一是需要展示出总的分类数目和各分类下的博客总数，其二是需要展示出选中分类下的博客列表，如下图所示：

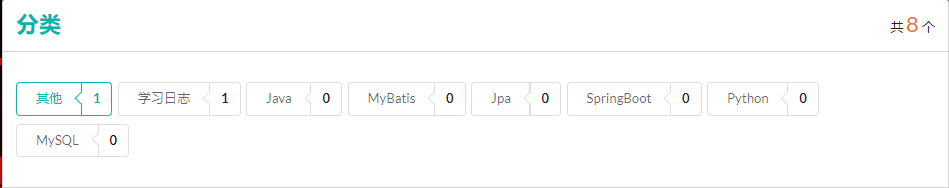


图 3 分类页面第一部分

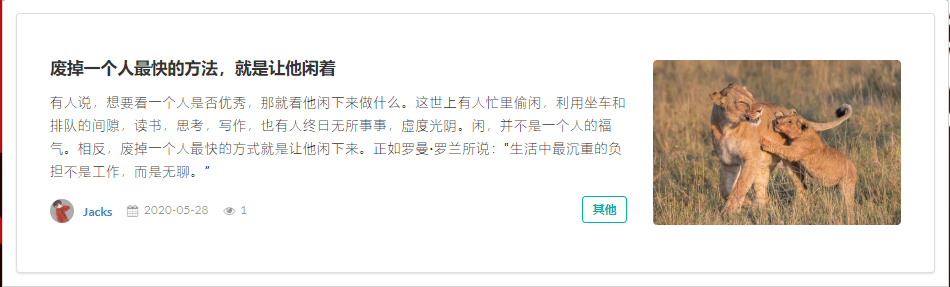


图 4 分类页面第二部分

2.2.3 标签展示

标签展示与分类展示十分类似，也是由两部分组成，如下图所示：

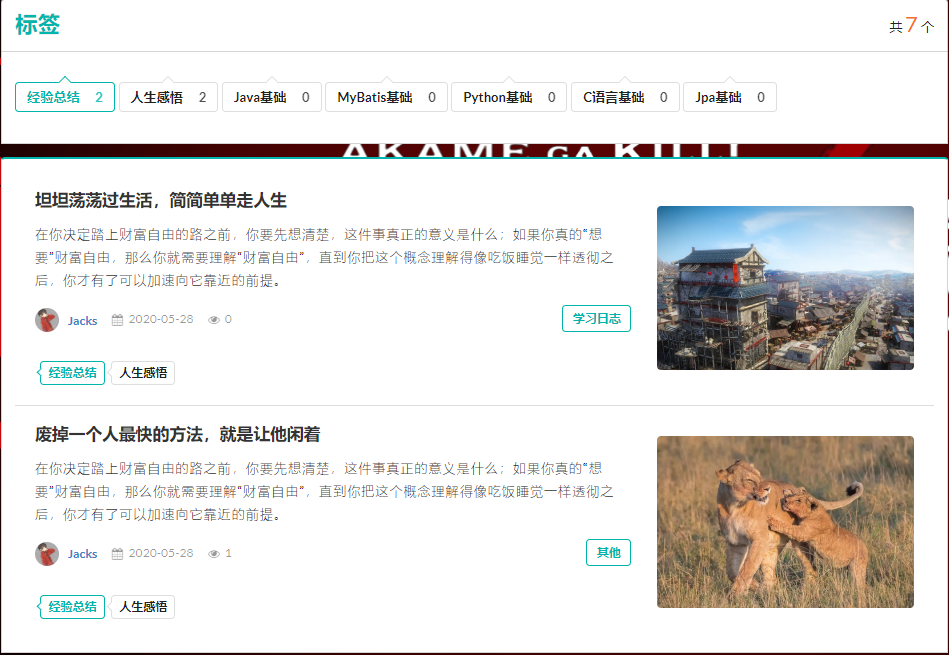


图 5 标签页展示

2.2.4 归档页面展示

归档页面只需要显示相应年份和对应年份下的博客列表就可以了，如下图所示：



图 6 归档页面展示

2.2.5 关于我页面展示

这个展示页面主要用于展示博主相关信息，我使用了图片、标签、简介的形式来展示个人信息，如下图：

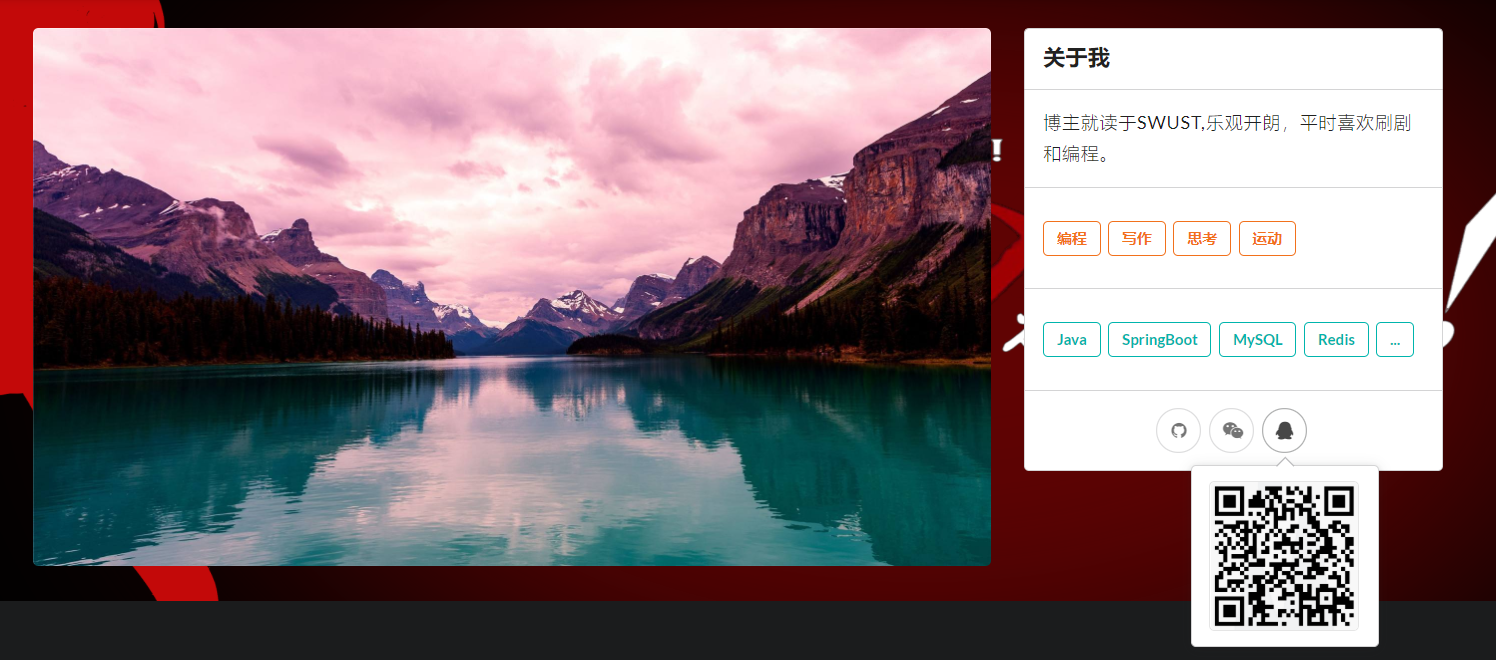


图 7 关于我展示页面

2.2.6 博客详情页面展示

博客详情页需要展示博客的类型（原创，转载或翻译）、作者、发布时间、浏览数目、正文、标签、留言等信息，如下图所示：

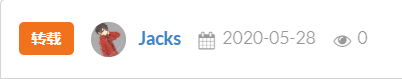


图 8 详情页头部信息

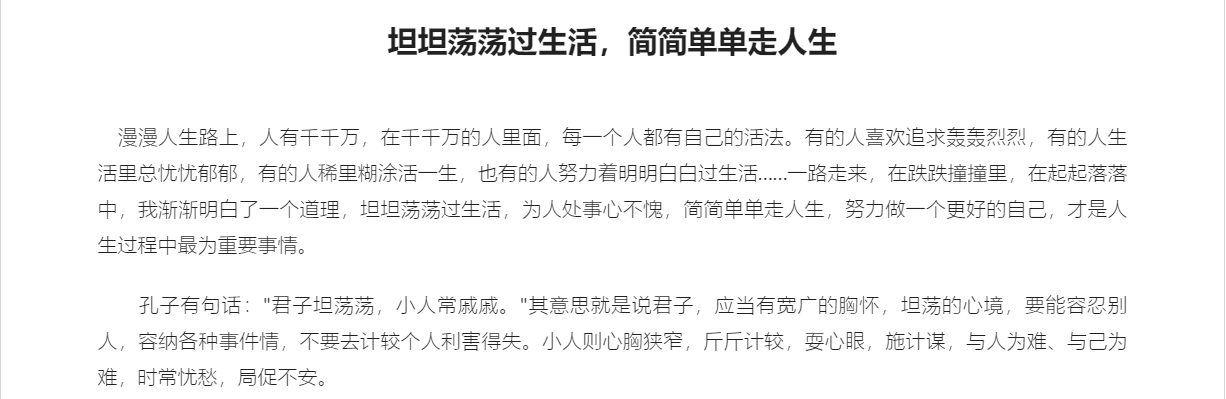


图 9 详情页正文内容



图 10 博客标签、赞赏、转载声明部分



图 11 留言板部分

2.2.6 搜索结果页面

访客可以在顶部的搜索框中输入关键字进行全局搜索，搜索结果会展示在一个新的页面，如图所示：

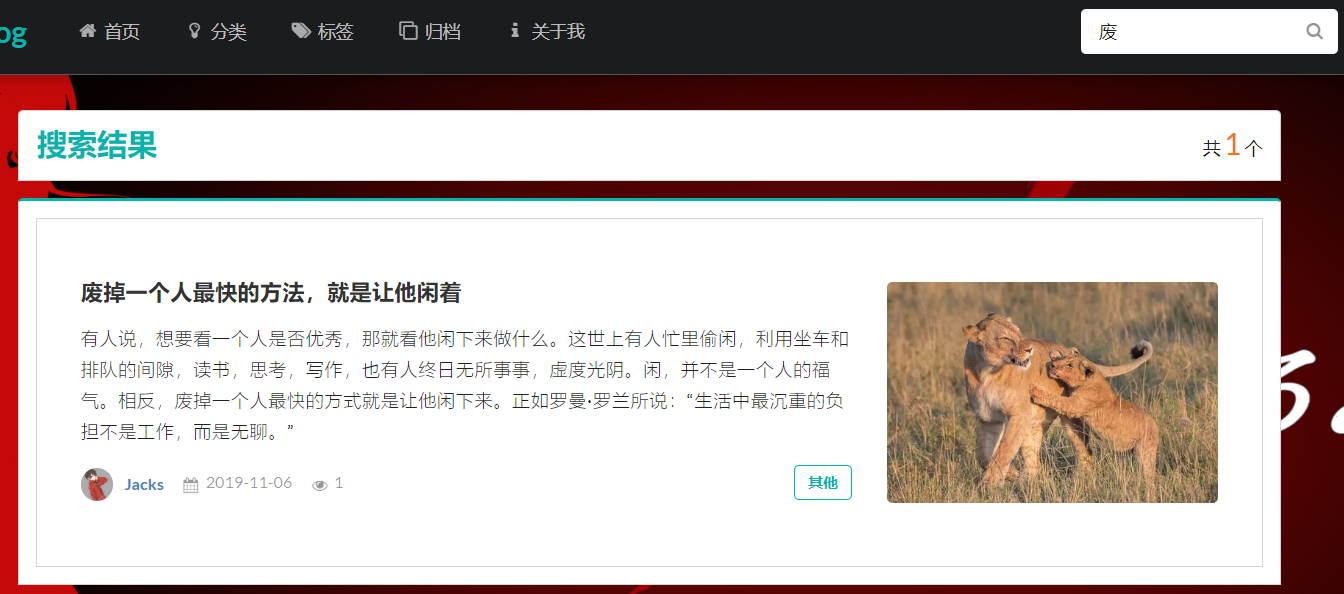


图 12 搜索结果展示图

## 2.3 后台管理展示页面

后台管理页面用于管理员维护博客相关的信息，从登录到管理博客一共有8个页面，分别是登录页面、欢迎页面、博客列表页面、博客编辑页面、分类列表页面、分类编辑页面、标签列表页面、标签编辑页面，下面以图片方式一一展示。

2.3.1 登录和欢迎界面

本次设计是个人博客系统，不考虑注册功能，登录界面也只是一个简洁的登陆窗口，欢迎界面采用的是一行话加上一幅图片组成的一个页面，如下图所示：



图 13 登录界面



图 14 欢迎界面

2.3.2 博客管理页面

博客管理页面包括两个部分，分别是博客列表和博客编辑两个页面。博客列表页面有编辑、删除、新增等操作按钮，点击按钮后会进入博客编辑页面。管理员还能够使用关键字和分类进行查询，如下图所示：



图 15 博客列表页

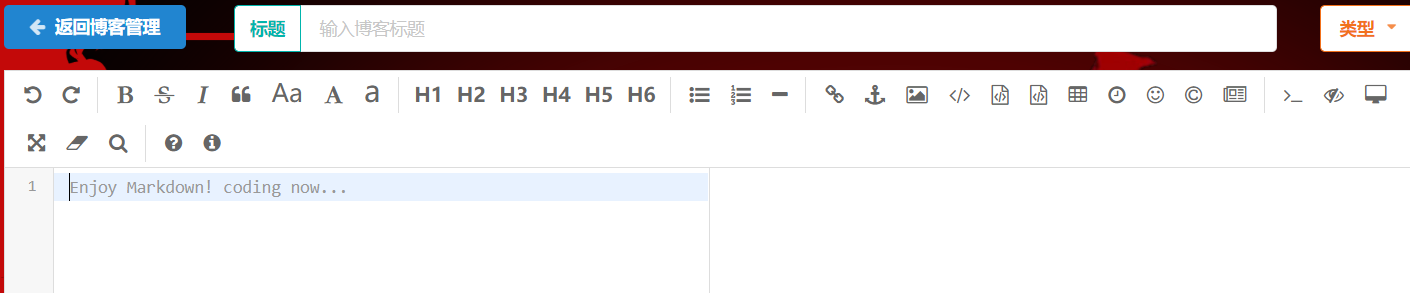


图 16 博客编辑页1

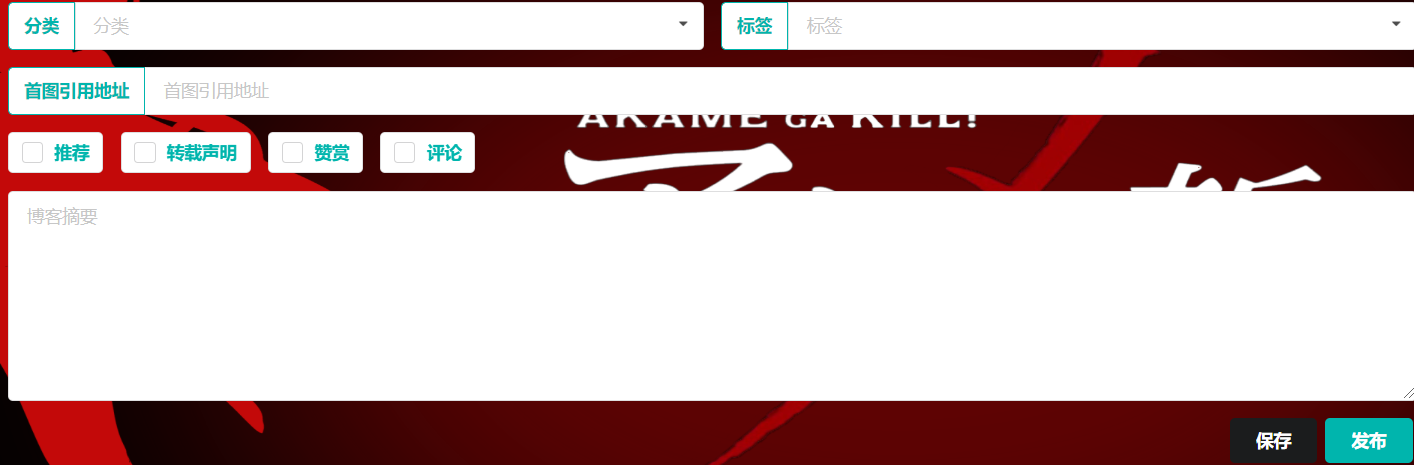


图 17 博客编辑页2

2.3.3 分类管理页面

分类管理页面与博客管理页相似，分类列表页和编辑页，如下图所示：

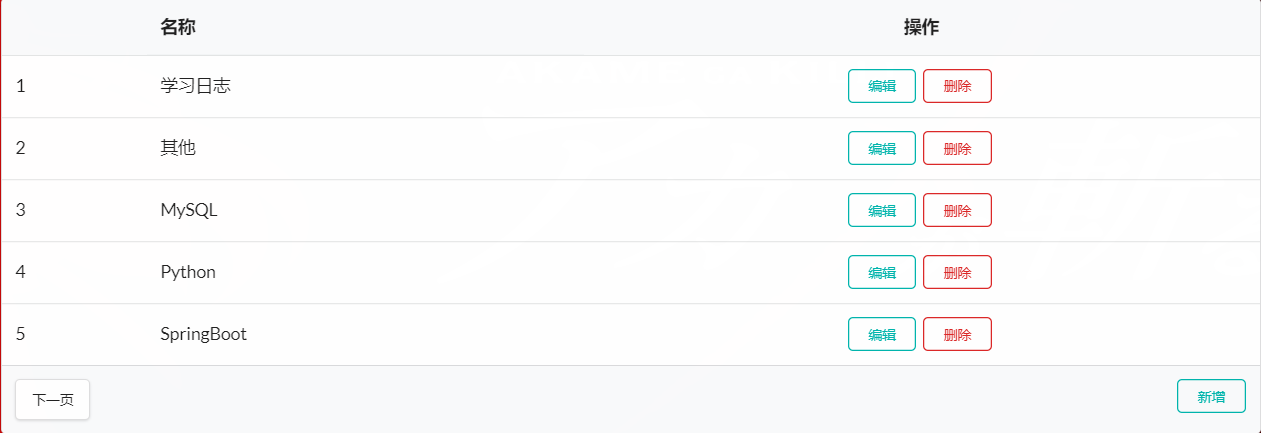


图 18 分类列表页面



图 19 分类编辑页

## 2.4 后台接口及数据库设计

2.4.1 数据库设计

我使用的数据库版本是MySQL8.0，根据需求分析需要建造5个主表，分别是用户表、博客表、分类表、标签表和评论表， E-R图详情见**附录1**。其中博客与标签是多对多的关系，所以需要外键关联表来支持多对多的联系，所以最终需要建6个表，表的详细属性见第三章详细设计。

2.4.2 后台接口设计

接口分为前端展示接口和后台管理接口，前端展示接口需要从数据库中拿到相关数据，所以需要设计较多接口模块，接口详情见第三章详情设计。

第3章 详细设计

## 3.1 数据库设计

3.1.1 用户表

|  |  |  |  |
| --- | --- | --- | --- |
| **名字** | **类型** | **长度** | **描述** |
| id | bigint | 20 | 主键 |
| avatar | varchar | 255 | 头像 |
| create\_time | datetime | 6 | 创建时间 |
| update\_time | datetime | 6 | 更新时间 |
| email | varchar | 255 | 邮箱地址 |
| nick\_name | varchar | 255 | 昵称 |
| type | int | 11 | 账户类型 |
| user\_name | varchar | 255 | 用户名 |
| password | varchar | 255 | 密码 |

3.1.2 博客表

|  |  |  |  |
| --- | --- | --- | --- |
| **名字** | **类型** | **长度** | **描述** |
| id | bigint | 20 | 主键 |
| title | varchar | 255 | 标题 |
| content | longtext | 255 | 文本内容 |
| first\_picture | varchar | 255 | 插图地址 |
| flag | varchar | 255 | 类型 |
| views | int | 11 | 浏览次数 |
| description | varchar | 255 | 描述 |
| appreciation | bit | 1 | 赞赏 |
| share\_statement | bit | 1 | 转发声明 |
| comment\_tabled | bit | 1 | 评论 |
| recommend | bit | 1 | 推荐 |
| published | bit | 1 | 发布 |
| createtime | datetime | 6 | 创建时间 |
| updatetime | datetime | 6 | 更新时间 |
| type\_id | bigint | 20 | 分类 |
| user\_id | bigint | 20 | 博主 |

3.1.3 分类表

|  |  |  |  |
| --- | --- | --- | --- |
| **名字** | **类型** | **长度** | **描述** |
| id | bigint | 20 | 主键 |
| name | varchar | 255 | 名字 |

3.1.4 标签表

|  |  |  |  |
| --- | --- | --- | --- |
| **名字** | **类型** | **长度** | **描述** |
| id | bigint | 20 | 主键 |
| name | varchar | 255 | 名字 |

3.1.5 评论表

|  |  |  |  |
| --- | --- | --- | --- |
| **名字** | **类型** | **长度** | **描述** |
| id | bigint | 20 | 主键 |
| avatar | varchar | 255 | 头像 |
| content | varchar | 255 | 内容 |
| create\_time | datetime | 6 | 创建时间 |
| email | varchar | 255 | 邮箱地址 |
| nick\_name | varchar | 255 | 昵称 |
| blog\_id | bigint | 20 | 对应博客 |
| parent\_comment\_id | bigint | 20 | 自关联 |
| admin\_comment | bit | 1 | 博主评论 |

3.1.6 博客与标签的外键关联表

|  |  |  |  |
| --- | --- | --- | --- |
| **名字** | **类型** | **长度** | **描述** |
| blogs\_id | bigint | 20 | 博客主键 |
| tags\_id | bigint | 20 | 标签主键 |

## 3.2 Thymeleaf模板配置

前端展示页面需要Thymeleaf模板引擎进行渲染，所以取用每个单独页面共同的部分作为模板内容，包含CDN资源链接的引用、导航栏和底端，具体代码实现见附件项目工程，如下图部分：



图 20 模板页面

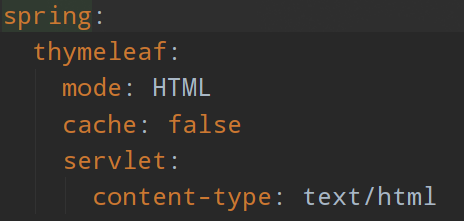


图 21模板配置文件配置

## 3.3 页面渲染及后台业务逻辑

3.3.1 首页

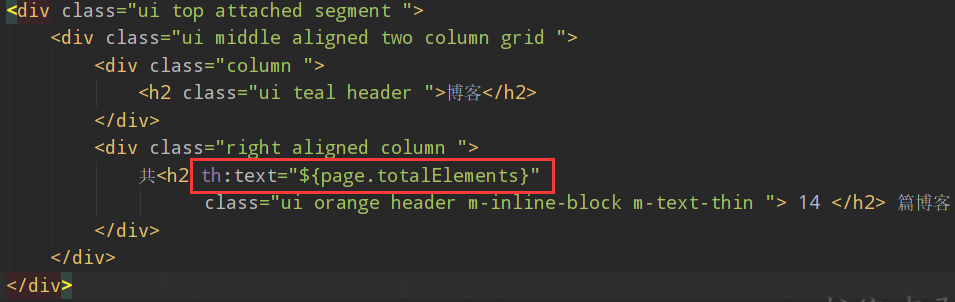
首页需要展示博客列表、分类列表、标签列表和推荐列表，接口定义如下：



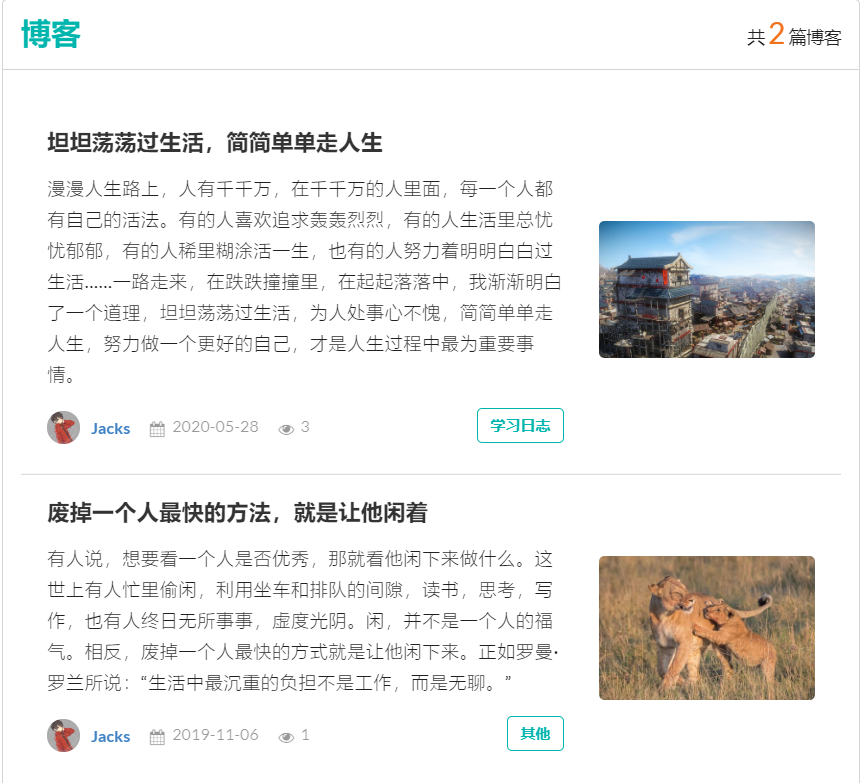
如图所示，用Pageable对象将博客列表分页封装好，在用Model对象将封装好的博客列表以键值对k-v的方式存储，最后在HTML页面中进行取值，以达到渲染页面的效果，分类列表、标签列表和推荐列表类似。



如上图，这里的博客的篇幅是个变量，根据数据库中实际保存的博客条数显示数字，页面渲染代码如下：



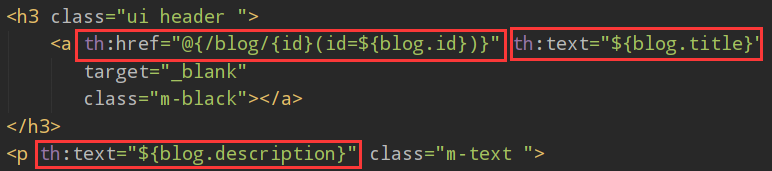
博客列表也是通过page键来进行渲染的，效果如下：



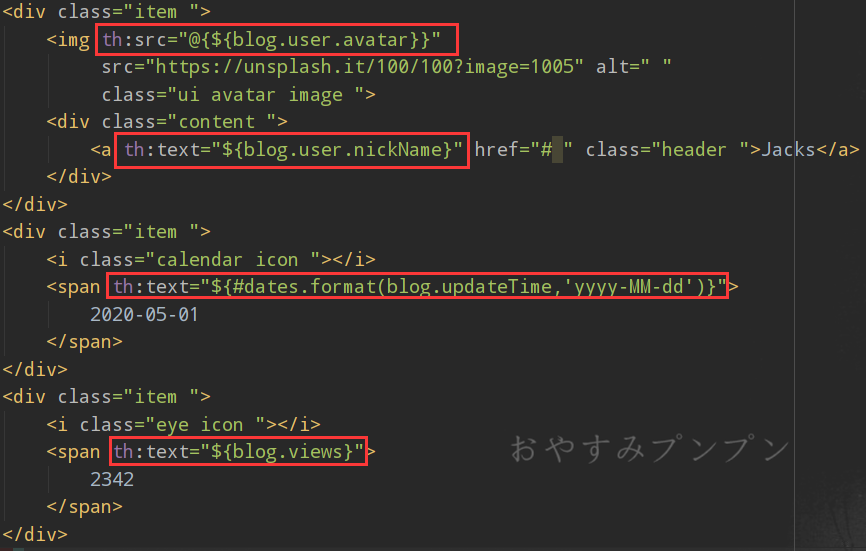
页面渲染代码如下：



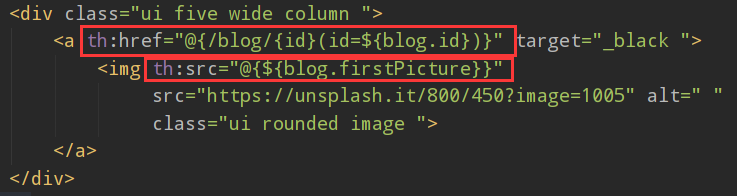
首先通过Thymeleaf模板引擎提供的的th:each标签对page键的content值进行循环，遍历出每一个blog对象。其后用th:if标签判断blog对象的发布状态，已发布的博客才进行取值渲染，未发布的博客则不会被渲染到页面上。



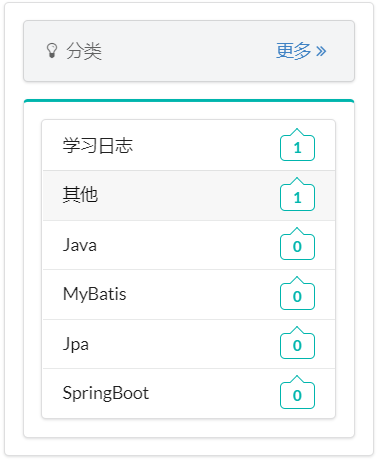
从已发布的blog对象中取出标题和摘要部分渲染到页面上，并且把标题放到a标签中，当用户点击标题时就可以跳转到博客详情页面进行阅读。

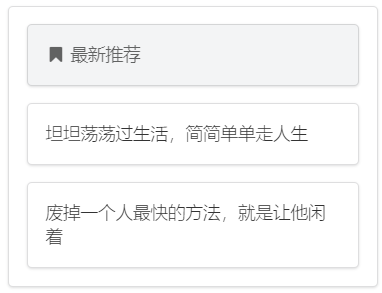


同样的处理手段，取出博客的作者信息（头像、昵称）、博客更新时间（格式化后的时间）、浏览数目等数据渲染到页面上。



右边的图片链接也取出来，放到每个博客的右边进行展示，点击图片时也会跳转到相应的博客详情页。右边的top列表（分类、标签、推荐）也用同样的方法渲染到页面，如下图所示

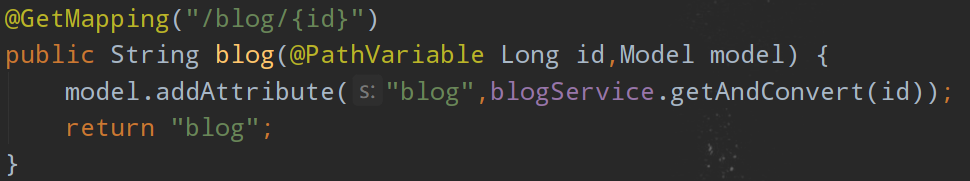




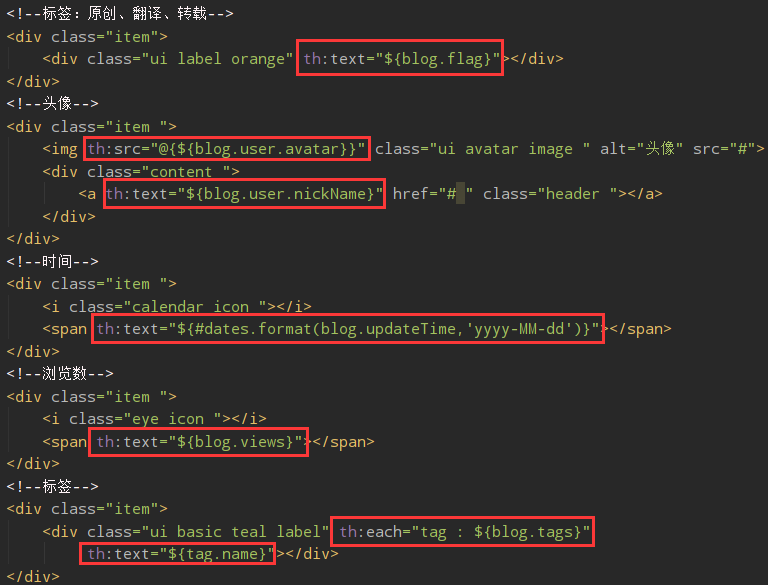
3.3.2 博客详情页



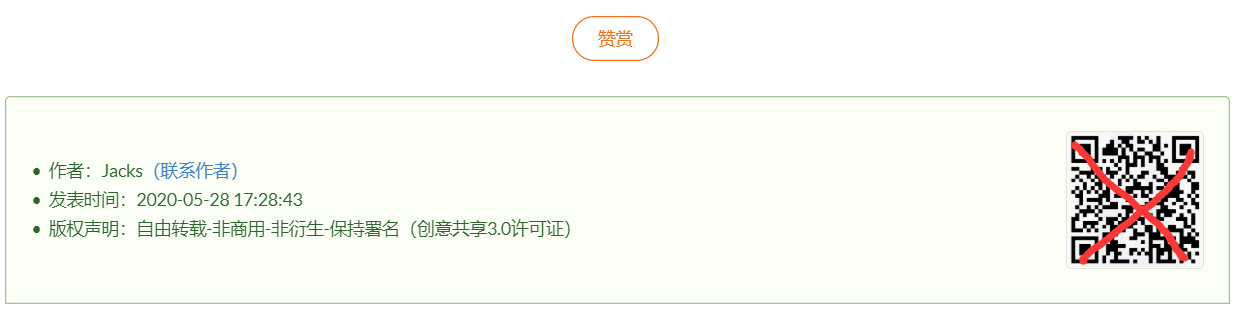
首先是头部和正文部分，头部有博客类型、作者头像和昵称、更新时间、标签等属性，正文部分就是作者编写的博客内容。页面接口如下：



由于正文部分的内容是以markdown格式存入数据库的，要渲染到页面上需要将markdown格式转换成HTML格式，这里的getAndConvert方法是整合了网上提供的工具类MarkdownUtils的自定义方法，具体实现见附件项目工程里面com.jk.blog.utils包下的MarkdownUtils类。页面渲染主要代码如下：



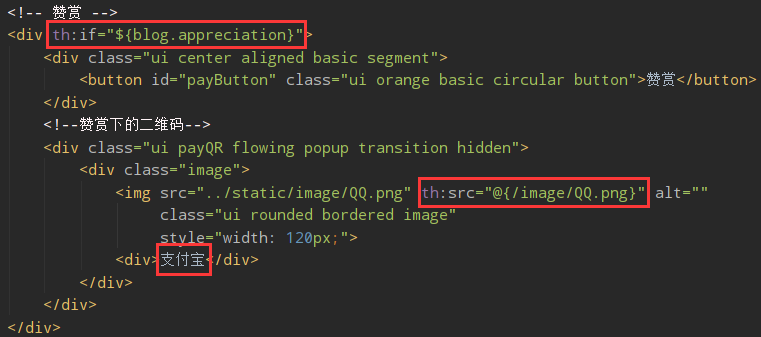
一篇博客允许拥有多个标签，是多对多的关系，所以需要使用th:each循环后进行逐个渲染。赞赏和转载声明展示如下：



当访客点击赞赏按钮后会出现一个二维码提供给读者进行打赏。如下图所示：



页面渲染主要代码如下：



首先判断是否开启赞赏功能，如果博主没有开启赞赏功能，页面就不显示赞赏按钮。（注：项目中所有需要提供二维码的地方都用的QQ二维码代替，所以路径都是/image/QQ.png）

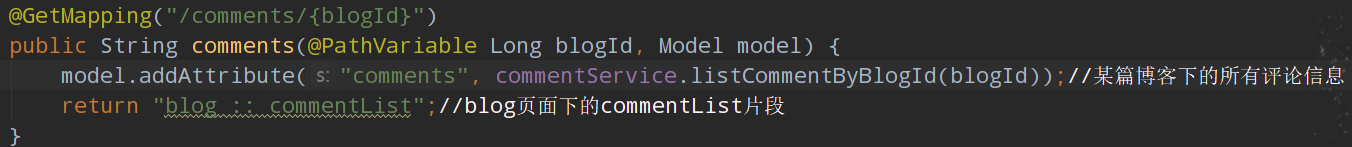
转载声明渲染代码如下：



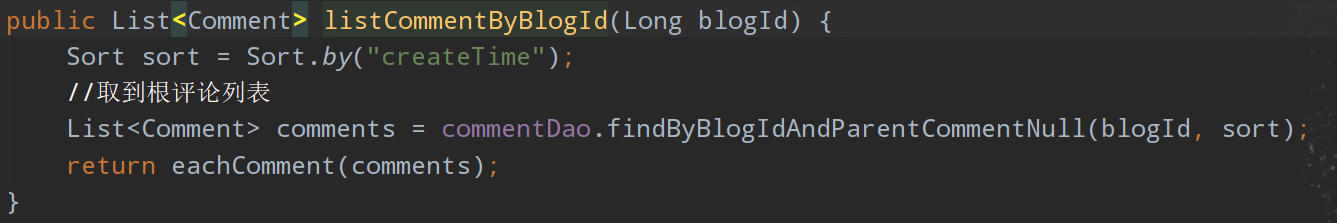
留言板部分页面展示如下：



留言板部分是整个项目中复杂板块之一，接口定义如下：



正如注释中说明那样，首先是根据博客的id查找该博客下的全部评论信息，然后用k-v方式存储，用于更新blog.html页面下的commentList板块。业务逻辑代码实现如下：



首先创建了一个Sort对象，根据评论创建的时间进行排序（默认升序），然后取出所有的根评论（这里主要区分于回复评论），然后使用自定义函数eachComment()递归遍历出树结构中所有子评论。评论树结构见**附录2**，业务逻辑代码如下：



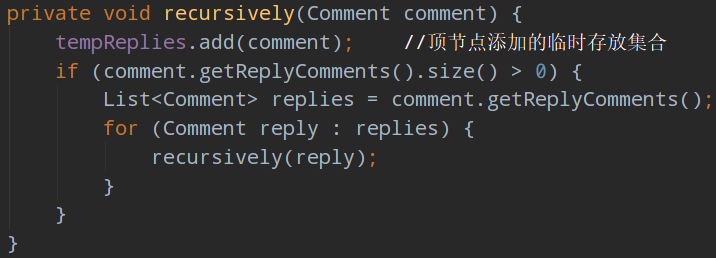
这里首先将用foreach循环遍历出每一个根节点，后将单个根节点复制到Comment对象c中，再把一个个根节点添加到中转父节点列表commentView中，最后将中转后的评论列表传给combineChildren()方法中进一步处理，这样做的好处是避免影响到数据库中的数据。



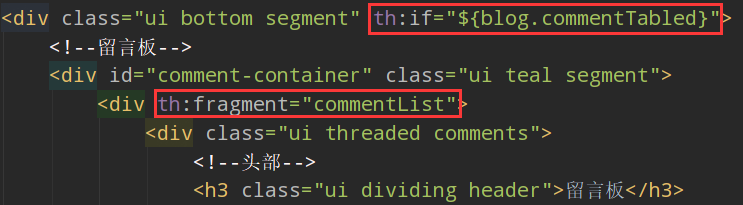
在combineChildren()方法中，首先遍历每一个根评论节点，然后将每个根评论节点的所有子回复节点放入新建列表replies集合中，再遍历出每一个子回复节点，然后将子回复节点作为起点，使用自定义递归查找函数recursively()找出分支下的所有子回复节点，最后将查询出的所有级的子回复节点设置为根评论的二级节点，这样就可以在页面渲染的时候分层显示根评论和回复评论。临时列表tempReplies的定义如下：



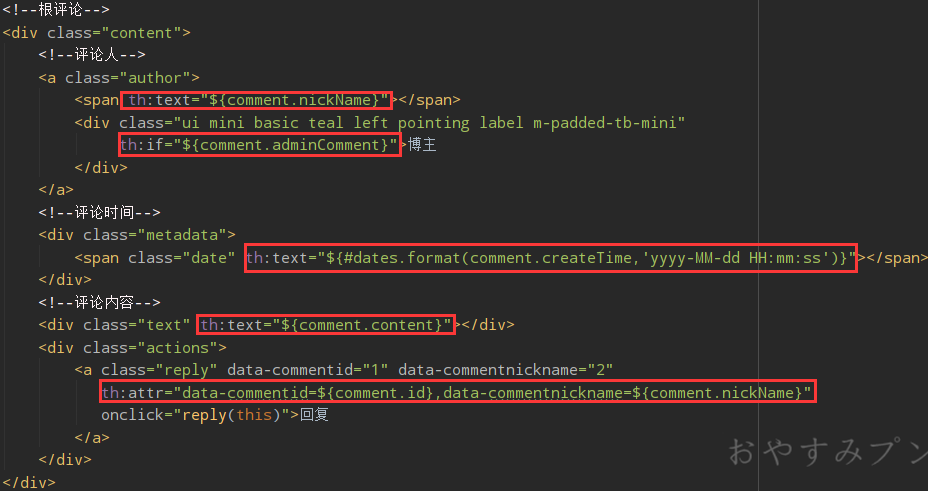
递归查询函数recursively()定义如下：



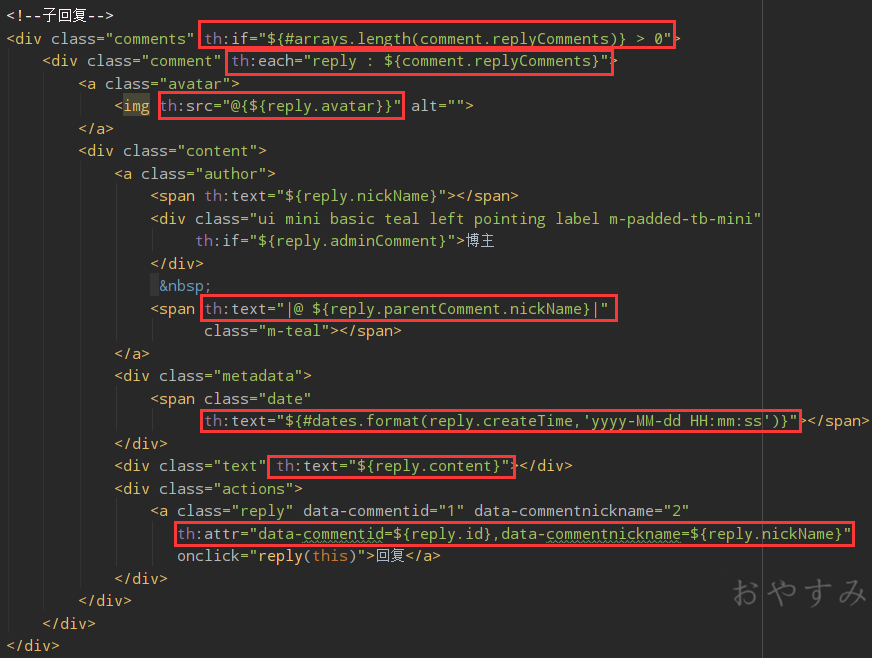
首先将作为起点的评论节点添加到零时列表tempReplies中，判断起点节点是否还有子评论，若有则继续像前面那样处理，进行递归查询，最终所有的子评论节点都会添加到临时列表tempReplies中，最终将tempReplies作为二级回复节点设置到根节点中，页面渲染时可以直接到二级回复节点中获取到所有的子回复评论了。页面渲染主要代码如下：



首先判断博主是否开启评论，只有博主开启评论功能后，游客才会看到评论模块和使用评论功能。然后将留言区域设置为commentList段，配合按钮接入留言板接口就可以实现动态更新了，即时评论即时查看。评论内容渲染主要代码如下：

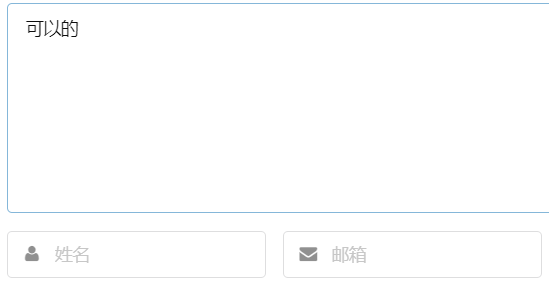


渲染与前面一致，若是博主进行评论或回复则会在昵称后加入博主的标签，回复列表渲染主要代码如下：

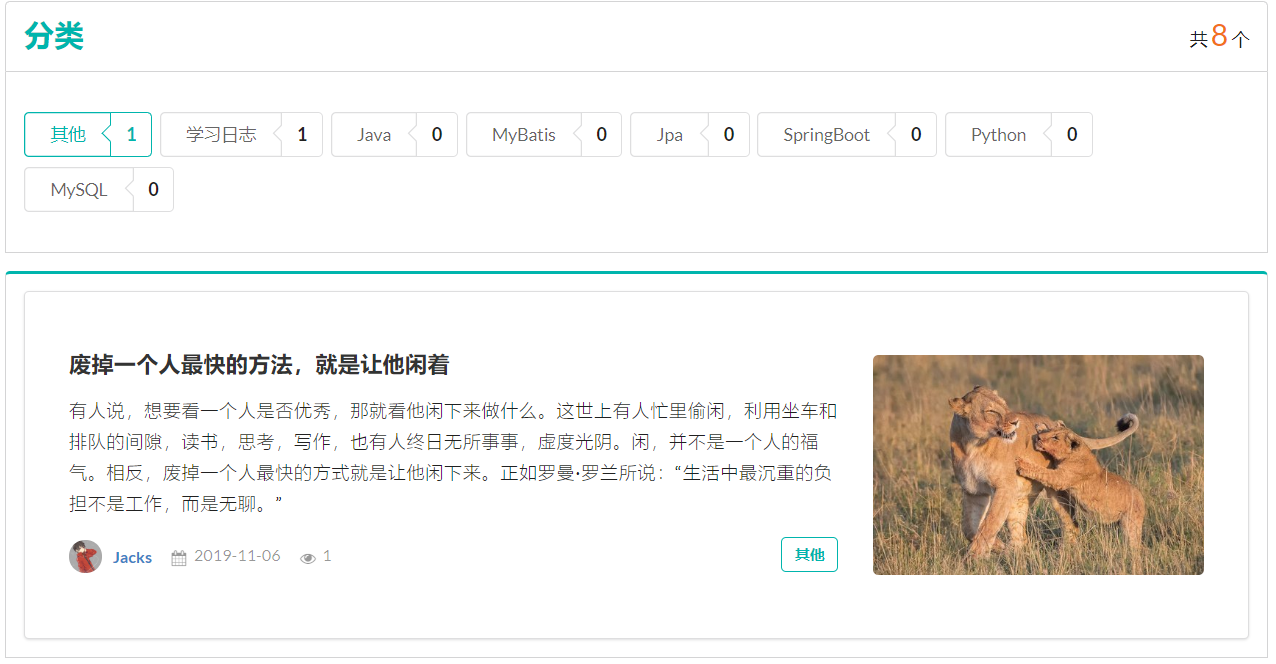


渲染回复列表的时候，首先判断根评论是否存在子回复，只有存在子回复节点的根节点才会进行二级渲染，渲染子回复的时候会以“小明@小红”的形式进行渲染，标识谁回复的谁。当用户点击回复的时候，回复输入框中首先会显示“@xxx”的字样表示当前是回复留言，当输入回复内容后，字样会自动消失，如下图所示：



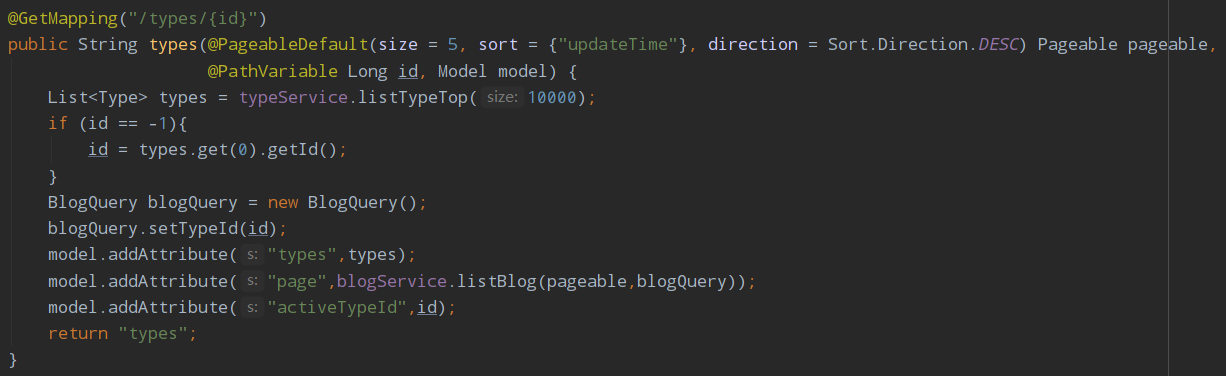


3.3.3 分类展示页面

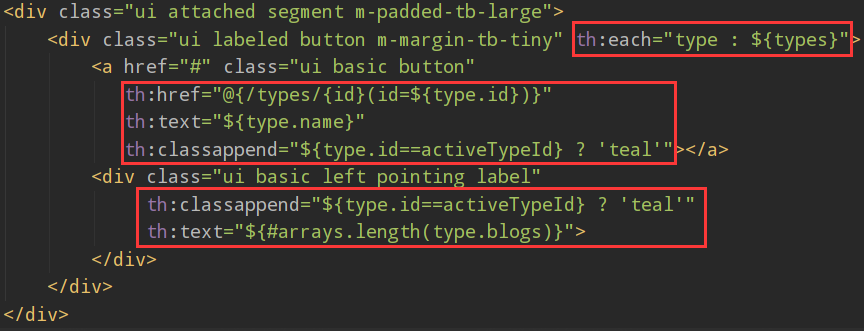


3.3.4 标签展示页面

分类展示页面需要展示分类的数目，分类列表，每个分类下的博客数目以及选中分类下的博客列表。分类接口如下图：



这个接口与博客详情页的接口类似，以k-v的方式存储分类、某个分类下的博客列表，activeTypeId用于标识分类的选中状态，页面渲染主要代码如下：



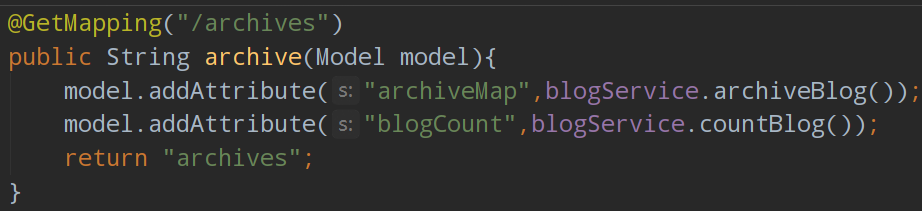
首先循环分类列表，将当前选中分类的id传递到接口，接口根据分类的id渲染博客列表，并为选中分类的标签添加“teal”样式，用于标识选中状态，博客列表渲染与博客详情页一致，这里不作代码展示。

3.3.5 归档展示页面

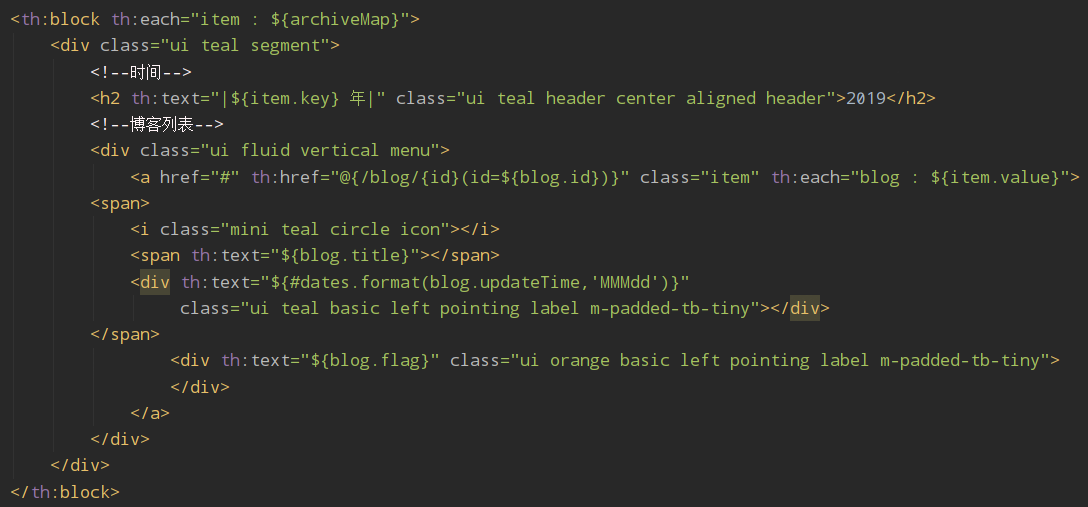


3.3.6 关于我展示页面

归档展示页面根据博客更新年份对博客进行归档，如上图所示，展示某个年份下的博客列表，在博客标题后面表示具体日期和博客类型。后台接口如下：



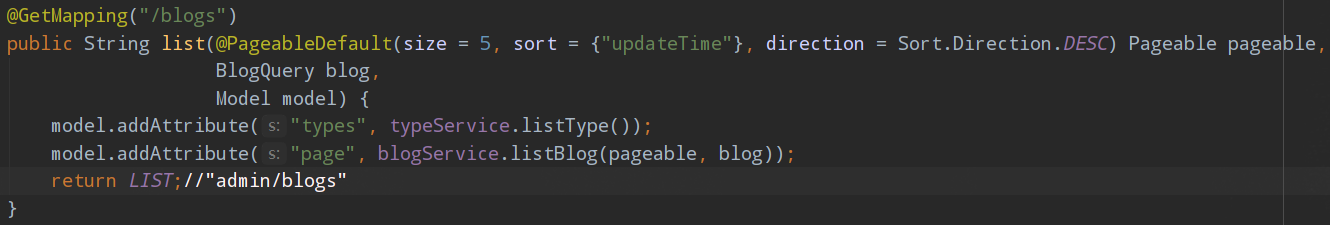
将归档后的博客以k-v的方式保存在archiveMap中，将博客总数保存到blogCount当中，页面渲染代码如下：



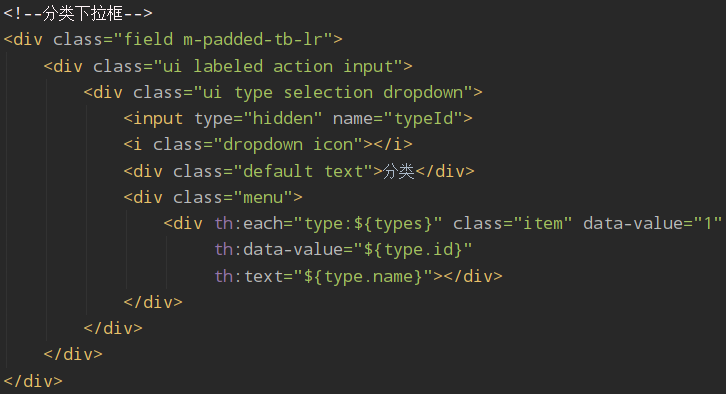
3.3.7 博客管理展示页面



这个页面有搜索框、分类搜索、博客列表及一些功能按钮，后台接口定义如下：

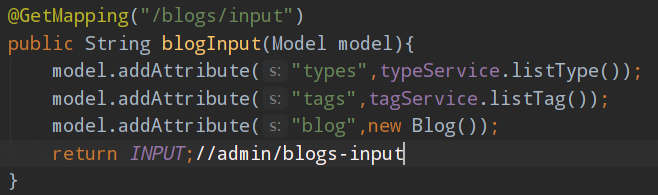


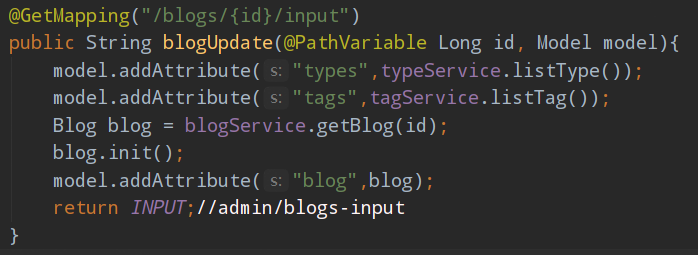
分类列表以k-v的方式存储到types内，然后做成下拉框，用于提供搜索条件。页面渲染主要代码如下：



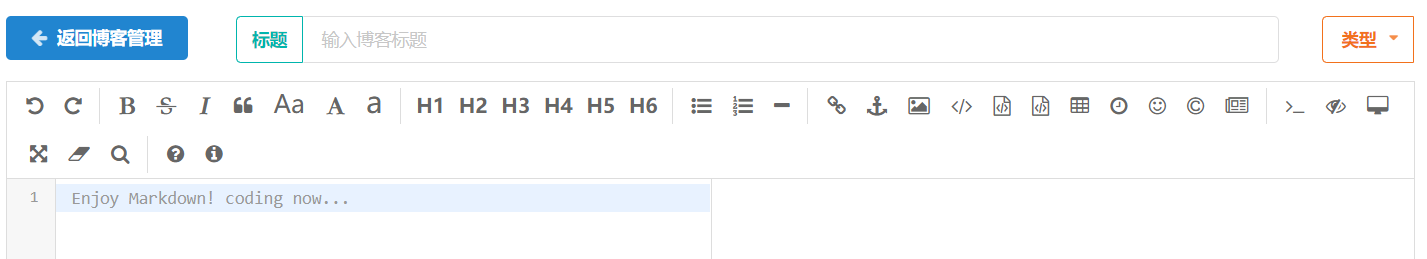
3.3.8 分类管理展示页面

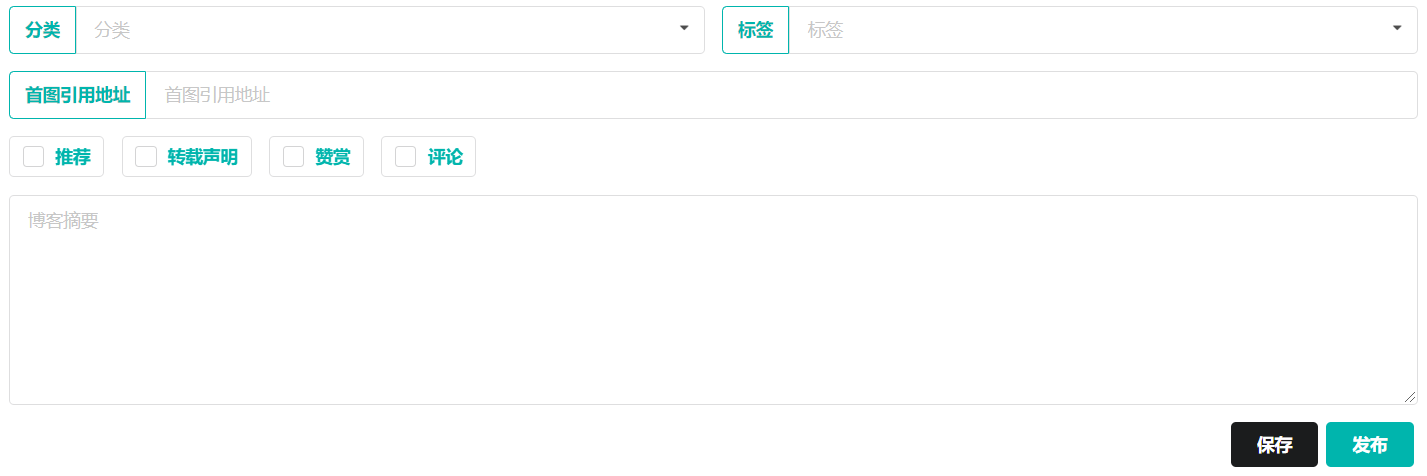
博客编辑和修改接口定义如下：



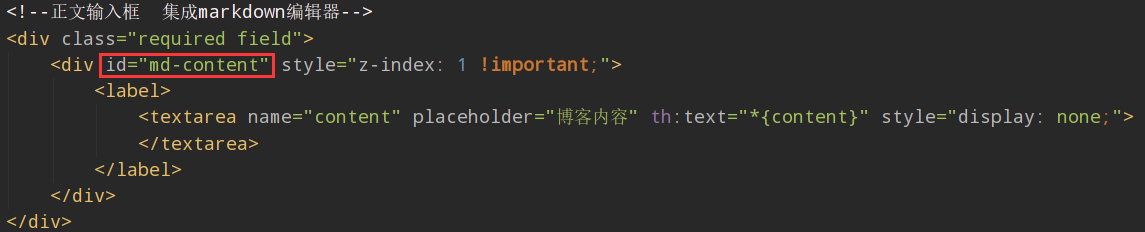


修改和编辑页面如下：



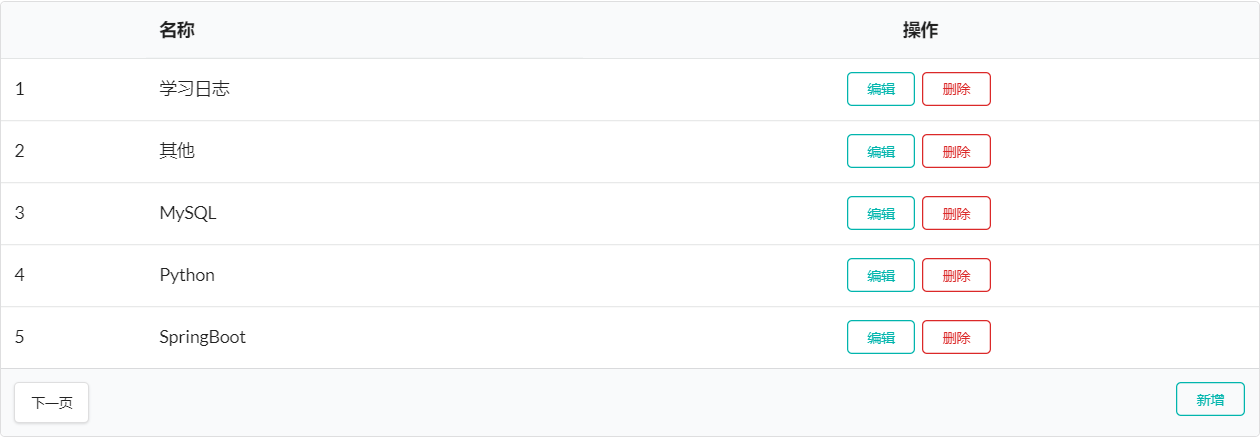


页面渲染和其他页面一致，这里集成了一个markdown编辑器插件，主要代码如下：

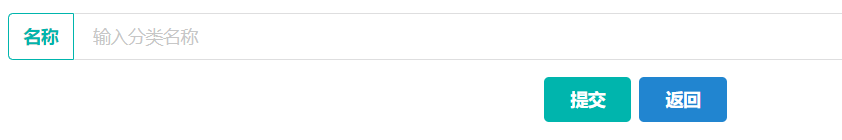




3.3.9 分类管理展示页面



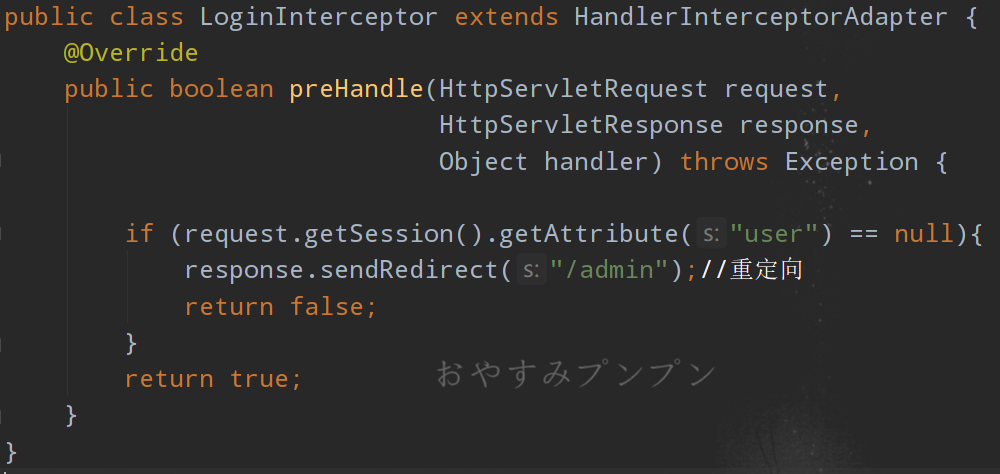
分类管理页面仅展示分类列表和功能按钮，编辑和修改页面如下：

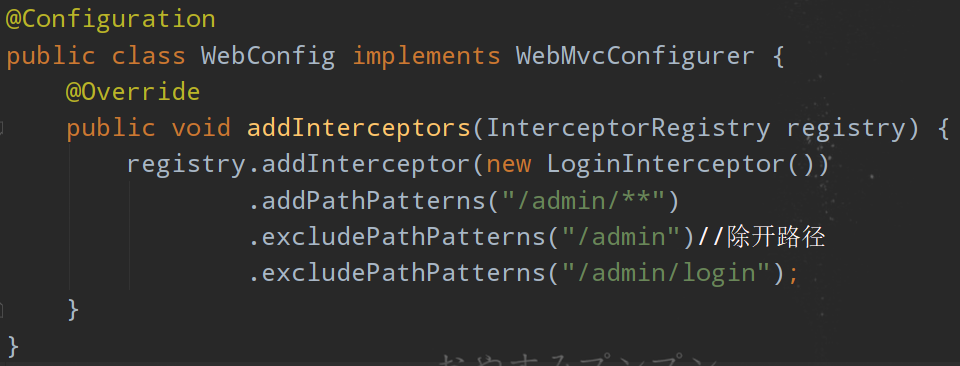


标签展示与分类展示完全一致，详情见附件。

3.3.10 拦截器配置

为了防止访客进入管理员界面进行操作，所以需要拦截器对后台管理页面进行拦截，必须登录才能够进入后台管理页面，拦截器配置如下：

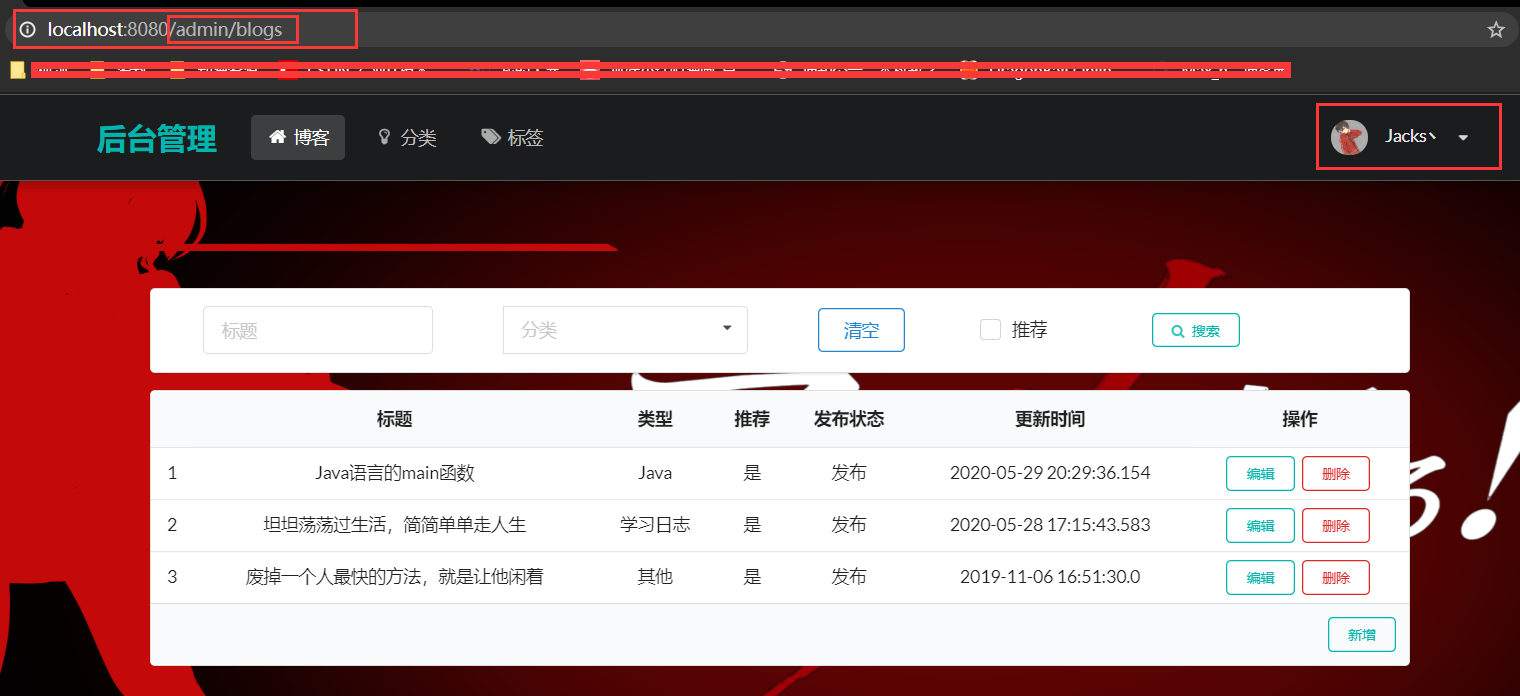




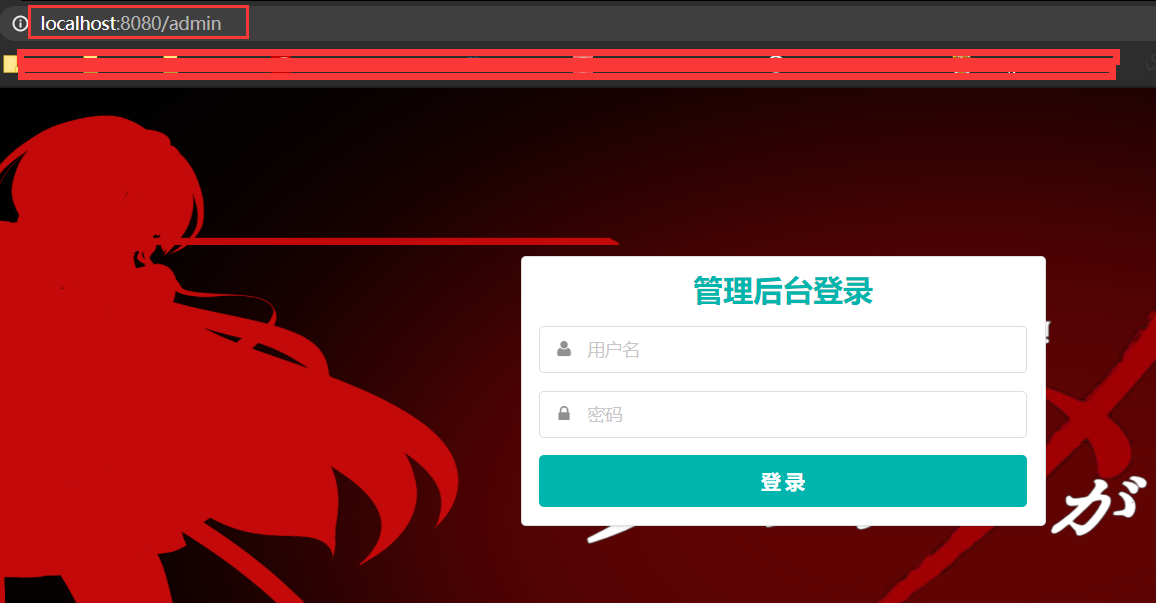
拦截器会拦截admin路径下除开admin和/admin/login路径外的所有接口，只有检查到用户登录状态才允许进入后台管理界面。

# 第4章 调试与运行

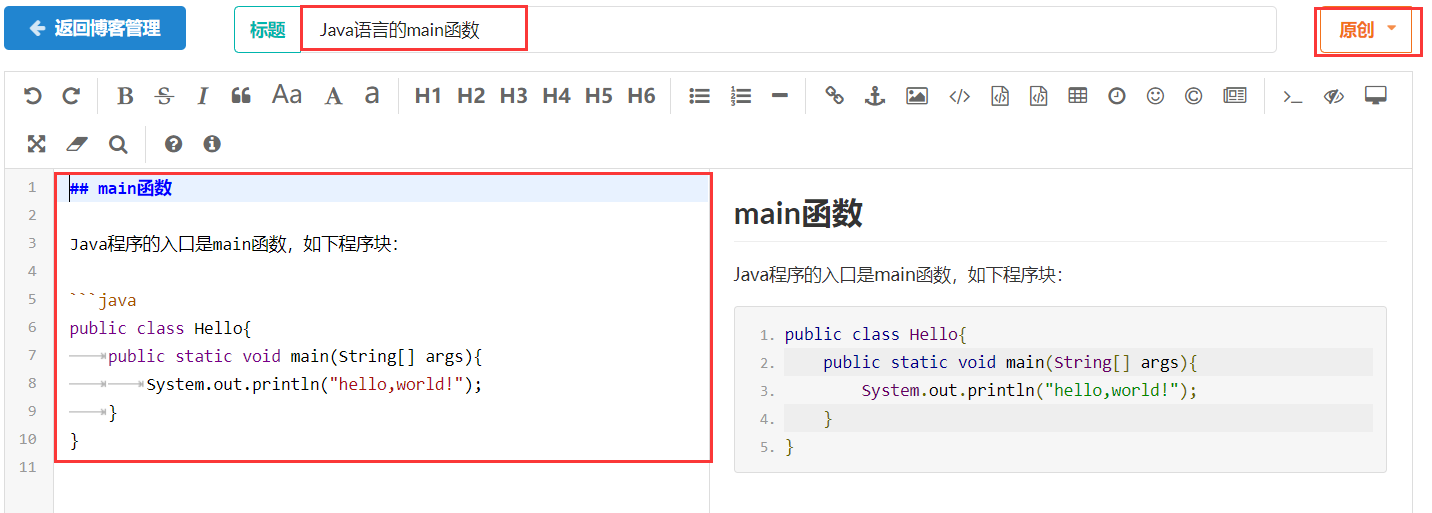
## 4.1 后台管理拦截器测试



现在为登录状态，因为用户信息存储在session里面，所以需要重启项目后在进行测试。重启后刷新界面，如下图所示：



## 4.2 博客新增功能测试



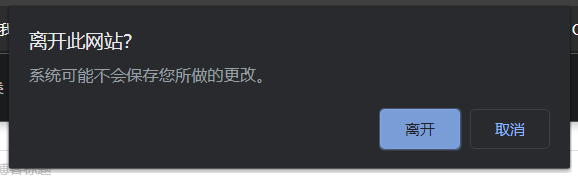




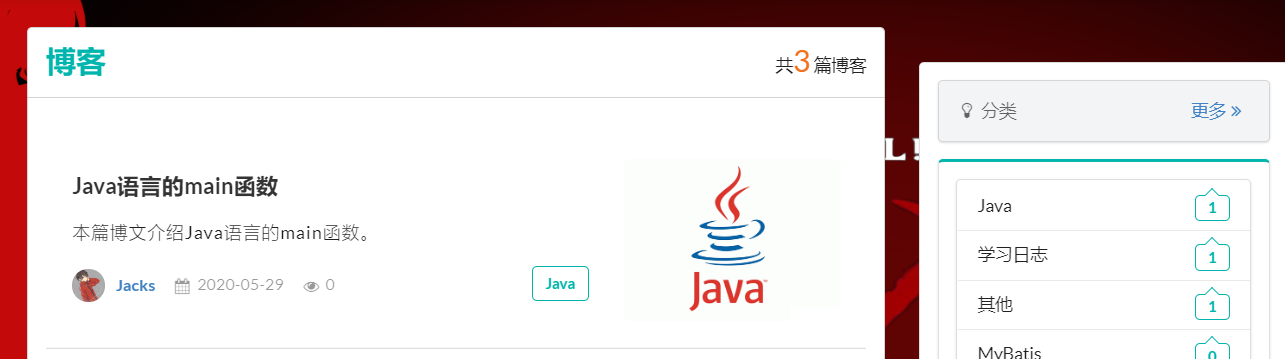
如果什么都没有输入的话，出现如下效果：

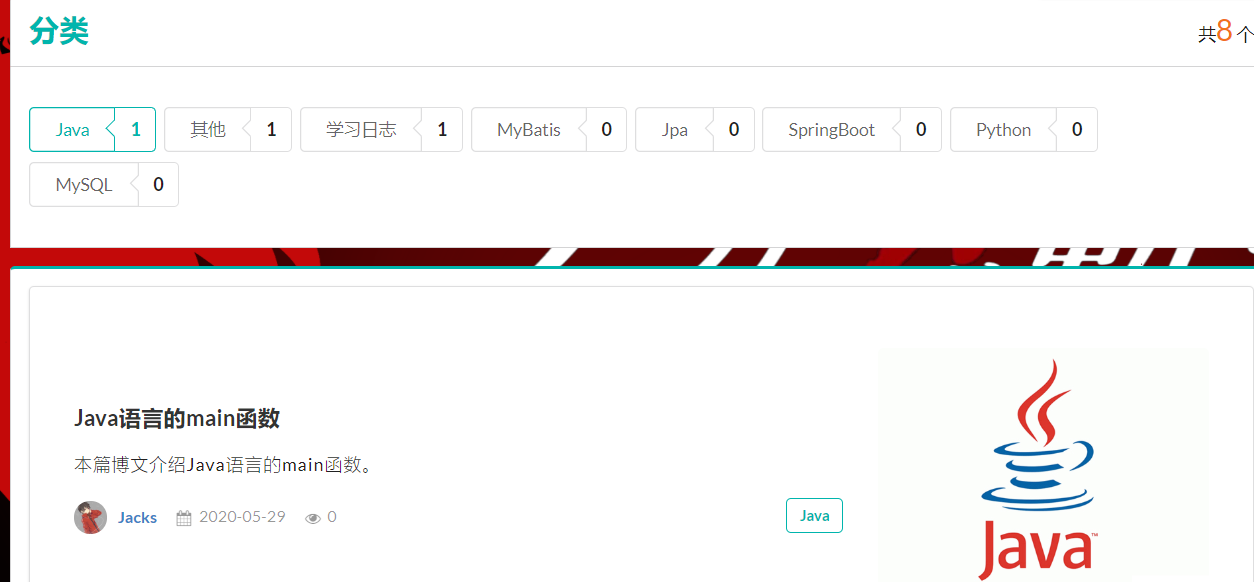


若未点击发布或者保存就离开页面，会出现如下效果：



## 4.2 测试访客页面渲染情况









通过测试，渲染页面成功渲染出预计的内容。

## 4.3 分类新增和标签新增测试

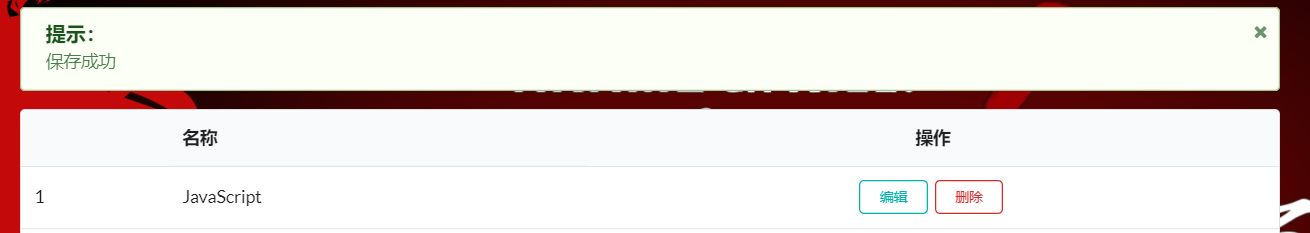
分类新增测试：





标签新增测试：



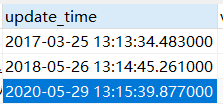


渲染展示：



## 4.4 归档渲染测试

将数据库中的更新时间改成往年年份：

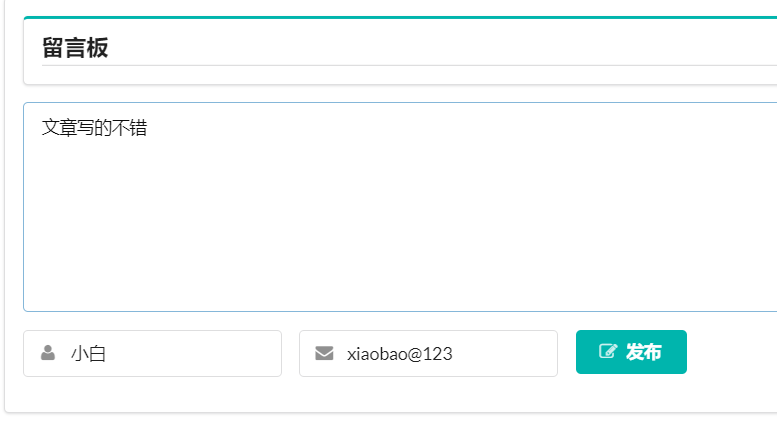


查看归档页面：



## 4.5 测试评论功能

测试访客评论：





测试博主回复：



总 结

本设计针对个人博客系统开发，实现了博客、博客分类、博客标签的新增、修改和删除功能，也成功渲染到了展示页面，设计到的技术指标全部测试成功。

主要完成工作：

（1）前端展示页面设计，主要借助了Thymeleaf模板和semantic-ui组件。

（2）后台接口设计，主要运用的Java语言和SpringBoot框架。

（3）借助Spring Data Jpa 生成了数据库表单。

（4）借助Navicat数据库管理软件进行功能测试。

本次设计尚有几处地方需要完善：

（1）页面动画效果不佳，应多加动画效果，让展示页面看起来更加美观。

（2）博客详情页没有设置目录。

（3）展示页面没有设置返回顶部按钮。

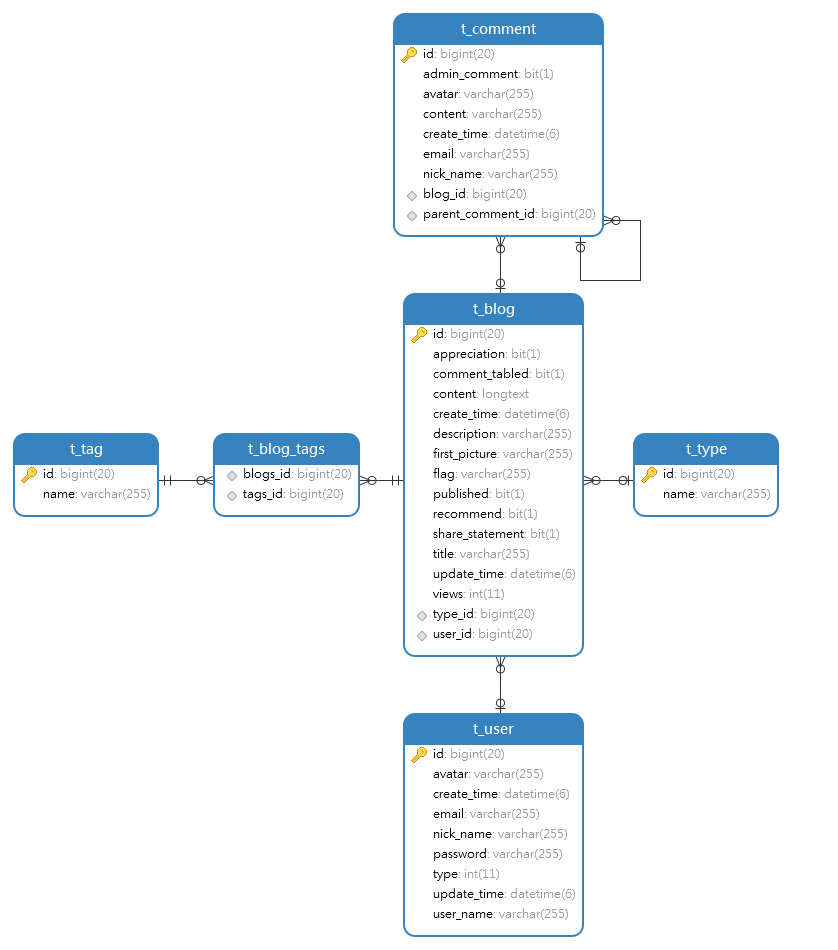
（4）博客正文没有集成代码高亮显示组件。

参考文献

1. 王国辉. Java Web编程宝典:十年典藏版[M].北京:人民邮电出版社，2011.
2. 李刚.轻量级J2EE企业应用实战[M].北京:电子工业出版社，2007.
3. 陈亚探，Struts2 + Spring + Hiberna框架技术与项目实战[M].北京:清华大学出版社，2012.
4. 吴锋.基于SSH框架JavaEE技术研究[D]. 安徽:合肥工业大学，2009.
5. 王向兵. JavaEE多层架构Struts2+Spring3+Hibernate3+Ajax 的整合[D]. 大连:大连海事大学，2009.
6. 赵利庆. Java Web架构中数据库优化模式的研究与实现[D].北京:北京邮电大学，2015.
7. 张琳娜. Java 多态性机制应用研究[J].攀枝花学院学报，2016， 14(02): 25-28.
8. 杨静.基于JAVA WEB中MVC模式的研究与应用[J]. 电脑知识与技术，2014, 9(28):68-71.
9. 李孟燃.网页设计中计算机多媒体技术的应用[J].计算机光盘软件与应用，2014, 6(12):55-57.

附录

## 附录1 数据库表E-R图



## 附录2 评论树结构

