# tzxblog博客系统-概要设计

|  |  |  |  |
| --- | --- | --- | --- |
| 版本：1.0 | | 版本日期：2020-2-1 | 文档状态：草稿 |
| 编制： | 涂宗勋 | | |
| 校对： | 涂宗勋 | | |
| 审核： | 涂宗勋 | | |
| 批准： | 涂宗勋 | | |

**版本变更记录**

|  |  |  |  |
| --- | --- | --- | --- |
| 版本 | 日期 | 修改人 | 变更理由/变更内容 |
| V1.0 | 2020-2-1 | 涂宗勋 | 初稿 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# 目录

[tzxblog博客系统-概要设计 1](#_Toc452)

[目录 3](#_Toc2838)

[一、系统概述 4](#_Toc14866)

[二、整体架构 4](#_Toc16566)

[三、版本控制 4](#_Toc21680)

[代码版本 4](#_Toc14595)

[文档版本 5](#_Toc9080)

[四、数据库设计 5](#_Toc30012)

[五、安全设计 5](#_Toc1591)

[六、功能设计 5](#_Toc21344)

[首页 5](#_Toc29387)

[文章详情 6](#_Toc10818)

[用户基础信息 6](#_Toc869)

[个人首页 6](#_Toc30550)

[文章评论 6](#_Toc25505)

[分类查询 6](#_Toc13276)

[存档查询 7](#_Toc31627)

[文章发布 7](#_Toc21576)

[文章管理 7](#_Toc28639)

[文件列表 7](#_Toc19491)

[文件上传 7](#_Toc2286)

[文件下载 7](#_Toc438)

[七、通用异常设计 8](#_Toc24086)

[八、项目地址 8](#_Toc14393)

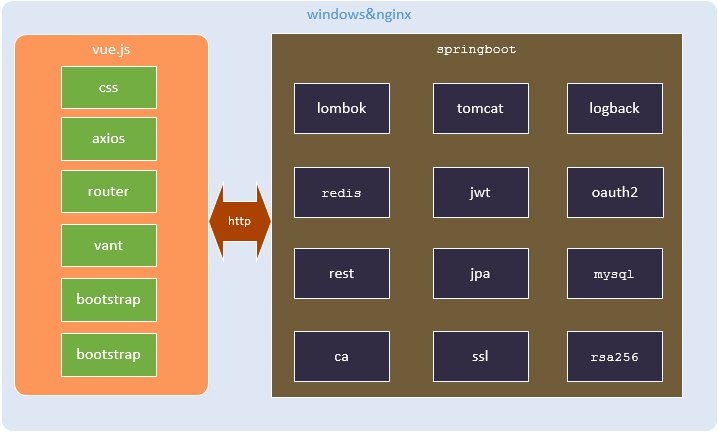
# 一、系统概述

tzxblog博客系统旨在创建一个可持续迭代的，以博客系统为基础并辅以文件下载、论坛讨论、商城等功能为一体的综合性站点。<br/>

重在持续性迭代，巩固工作中主流技术的同时，练习工作中未使用但主流或者不可或缺的技术。

# 二、整体架构

系统整体采用前后台分离思想，后台以springboot为基础框架，前台以vue为基础技术，辅以redis缓存、mysql存储、nginx转发以及ssl通信，力求软件生命周期内涉及开发的内容均可得到提现。



# 三、版本控制

## 代码版本

前后台代码均托管github，以不同分支管理大的版本，遵循如下要求：

* master：最新可运行版本，从develop合并；
* develop: 最新开发版本，可能不能运行，主要作用是代码备份，防止本地最新文件丢失；
* realese-X.X: X.X为数字泛指，如1.0、1.1、2.0，作为稳定大功能的发布版本，由master拉出；
* tags：tag版本作为废弃不用的版本存档。

## 文档版本

项目周期内各种文档，尽量以两种方式呈现，分别是博客文章和docx文档，类似接口文档、数据库这种不太好博客展示的，就仅以附件文档的格式输出。<br/>

其中docx文档存放于项目doc目录下，和代码一起托管于github，同时docx文档每次修改变更均要变更版本号并标注具体变更说明。

# 四、数据库设计

详见doc中数据库设计和sql脚本，后续补充。

\*\*\*

### 缓存设计

为提高系统性能，使用频率高的数据需要存放于redis缓存中。

# 五、安全设计

对于接口层面，前后台接口需要有签名校验，防止网络攻击。<br/>

对于如密码这种敏感数据，传输和存储均应加密处理。<br/>

系统前期可使用http协议，功能稳定后采用自签名证书和https交互。

# 六、功能设计

## 首页

1. 用户访问首页，前台向后台页面发起首页需要的各种请求：分类列表、文章列表、推荐列表，列表条数和推荐列表条数均可配置，默认分别为10和5；
2. 后台收到请求后查询redis缓存或者mysql数据库，组装数据响应前台；
3. 前台展示首页相关内容；
4. 成功进入首页后，分类列表、文章列表、推荐列表、导航按钮均可进行点击后进入具体的文章页、功能页。

## 文章详情

1. 用户点击文章列表、推荐列表具体文章，前台根据文章id向后台发起文章详情查询；
2. 后台收到请求后查询redis或者mysql，组装数据后响应前台；
3. 文章详情包含文章内容和阅读量、评论数、评论列表分页；
4. 文章详情页面三栏布局，顶部导航，下左方用户信息，下右方文章详情。

## 用户基础信息

1. 文章详情左侧显示用户基本信息，用户头像、昵称、简介、签名、以及用户文章分类和归档等信息；
2. 点击用户头像或昵称均可进入用户个人首页。

## 个人首页

1. 个人首页整体布局和文章详情类似，把文章详情区域的内容替换为文章列表。

## 文章评论

1. 文章详情末尾展示评论框，可输入文字评论；
2. 评论框下方展示评论列表，评论列表条数可配置，默认展示5条，评论必须登录。

## 分类查询

1. 个人首页或首页点击分类列表中的具体分类，前台向后台发起分类文章查询；
2. 后台查询具体分类的文章分页列表后，返回前台；
3. 前台在个人首页的下右方主区域或者首页的下中区展示存档文章列表。

## 存档查询

1. 个人首页点击存档列表中的具体月份，前台向后台发起存档文章查询；
2. 后台查询具体月份的文章分页列表后，返回前台；
3. 前台在个人首页的下右方主区域展示存档文章列表。

## 文章发布

1. 用户登录后，在顶部导航栏展示发布文章按钮，点击后进入文章编辑页面；
2. 文章内容使用markdown格式，编辑完成后点击发布；
3. 编辑页面包含文章标题、主内容、分类、是否原创、是否公开等信息。

## 文章管理

1. 用户登录后，在顶部导航栏展示文章管理按钮，点击后进入文章列表页面；
2. 对列表后展示删除、修改、分类、权限等按钮，均可进行编辑。

## 文件列表

1. 用户点击顶部导航中的下载页面，进入下载资源列表页面；
2. 下载页面展示可下载资源列表，同时提供上传按钮。

## 文件上传

1. 用户在文件列表页面可以点击文件上传按钮上传文件；
2. 上传文件必须登录。

## 文件下载

1. 文件列表中的文件可以直接下载，不需登录，后期考虑引入积分制。

# 七、通用异常设计

系统中的异常均需要人工捕获，避免直接出现系统类异常。<br/>

500、404等均需要有自定义的友好提示，自定义异常尽可能详细，使得前台提示更加清晰。

# 八、项目地址

项目代码和文档均以github托管，地址如下：

<https://github.com/tuzongxun/tzxblog>