
极客学院大前端毕业设计

Front end Engineer Graduation design

详细设计说明书

项目名称	〈X-SONGTAO 个人博客搭建〉
班 级	〈极客学院第一期 1 班〉
姓 名	〈向松涛〉
版 本	〈V1.0〉

二〇一六年三月十三日

版本修订记录

版本	日期	描 述	作者	审核
V1.0	2016-3-13	第一次修订	向松涛	A

目录

1	引言	4
1.1.	编写目的	4
1.2.	项目概况	4
2	系统概述	4
2.1.	系统技术选型	4
2.2.	系统简要架构和核心功能点	4
3	公用接口程序设计说明	5
3.1.	全局变量	5
3.2.	主要公用接口和类库	6
	附件	7

1 引言

1.1. 编写目的

通过对个人博客的编写，将大前端整体知识体系串起来完成毕业设计。

1.2. 项目概况

1. 个人博客项目为自己所用，设计目标朝向简明扼要为主。
2. 项目分为以下几个部分：博客页面设计、全响应式页面搭建、后台管理系统界面设计、管理系统页面搭建、nodejs 后台服务搭建、域名申请备案、阿里云服务器环境配置等部分。

2 系统概述

2.1. 系统技术选型

项目整体的技术选型为 MEAN 框架：Mongodb+Express+Angularjs+Nodejs。接口采用 Restfull 模式。博客页面使用的 Angularjs+BootstrapUI 库进行全响应式设计（1500px、1200px、991px、768px）。

2.2. 系统简要架构和核心功能点

项目使用的是阿里云服务器，服务端使用的是 CentOS 64 位操作系统，配置为：cpu 1 核、内存 1024mg、带宽 1Mbps。架构环境如下：nodejs+nginx+mongodb，外网监听 80 端口的请求，并将本地 127.0.0.1:8088 端口数据映射出去。Mongodb 的配置信息在 app/config 中，使用的 mongoose 作为数据库连接中间件。后台 node 服务使用 pm2 管理。已将服务器环境配置好，登陆域名为：xiangsongtao.com。后台用户名和密码均为：admin。目前并未设置用户访问权限。Ps，点击头像会有我的个人陈述。

程序组织的层次结构关系如下表：

文件夹名	子文件夹名	文件夹作用	功能说明
app	config	Mongodb 配置文件	配置文件
	controllers	路由信息处理文件	控制器
	models	Mongoose 模型定义	模型定义
	routes	路由控制	路由
	services	公共数据服务	公共数据
	utils	工具类	工具
	views	存放视图（blog+后台）	视图页面
Public	adminPages	后台管理静态资源	静态文件
	Fonts	字体	
	Home	前台 Blog 静态资源	
	Images	公共图片	
	javascripts	公共 js 脚本	
	stylesheets	公共样式文件夹	
	uploads	上传文件存放处	
bin	/	Nodejs 启动配置	
app.js	/	项目入口	

3 公用接口程序设计说明

3.1. 全局变量

Angularjs: 为前端页面的 mvc 载体，操作整个 DOM。

公共脚本库均使用外网 CDN。

Mongodb 数据库需要提前安装，没有设置密码。

3.2. 主要公用接口和类库

项目一共设计了三个数据模型：文章模型、个人信息模型、标签模型。因为不涉及到用户管理，故登陆信息写在了 `app/services/basicData.server.js` 中。文章模型和标签模型通过标签 id 关联，数据整合在 `controller` 中进行。项目一共分为两个大类：Front-End 和 LiftStyle，两个类别的补充信息也在 `app/services/basicData.server.js` 中。

1. 后台管理 Restfull 请求接口及说明，文件在 `app/routes` 中。

编号	请求方式	接口地址	说明
1	GET	<code>http://xiangsongtao.com/admin/api/myinfo</code>	个人信息列表
2	POST	<code>http://xiangsongtao.com/admin/api/myinfo</code>	修改个人信息
3	POST	<code>http://xiangsongtao.com/admin/api/myinfo/inguoload</code>	头像上传
4	POST	<code>http://xiangsongtao.com/admin/api/tag/add</code>	增加标签
5	POST	<code>http://xiangsongtao.com/admin/api/tag/edit</code>	标签修改
6	GET	<code>http://xiangsongtao.com/admin/api/tags</code>	标签列表
7	GET	<code>http://xiangsongtao.com/admin/api/article</code>	文章列表
8	GET	<code>http://xiangsongtao.com/admin/api/article/:id</code>	获得单个文章
9	POST	<code>http://xiangsongtao.com/admin/api/article/:id</code>	修改文章
10	DELETE	<code>http://xiangsongtao.com/admin/api/article/:id</code>	删除文章

2. 博客页 Restfull 请求接口及说明，文件在 `app/routes` 中。

编号	请求方式	接口地址	说明
1	GET	<code>http://xiangsongtao.com/api/:catalogueName/articles</code>	不同栏目文章列表
2	GET	<code>http://xiangsongtao.com/api/:catalogueName/articles/:id</code>	单独文章
3	GET	<code>http://xiangsongtao.com/api/:catalogueName/historys</code>	时间轴
4	GET	<code>http://xiangsongtao.com/api/:catalogueName/tags</code>	标签列表
5	GET	<code>http://xiangsongtao.com/api/:catalogueName/tags/:id</code>	标签获取
6	GET	<code>http://xiangsongtao.com/admin/api/myinfo</code>	个人信息列表

3. 公共服务

根据标签 id 查找标签名的公共服务，文件在 app/utils/commom.js 中。

4. 数据模型：文章模型

```
var Article = new mongoose.Schema({
  catalogueName: String,
  title: String,
  time: {type: Date, default: Date.now},
  readNum: Number,
  articleType: Number,
  tags: Array,
  state: Boolean,
  content: String
});
```

5. 数据模型：标签模型

```
var Tags = new mongoose.Schema({
  name: String,
  catalogueName: String,
  markClass: Number,
  hot: Number,
  state: Boolean
});
```

6. 数据模型：个人信息模型

```
var MyInfo = new mongoose.Schema({
  id: Number,
  name: String,
  position: String,
  address: String,
  mood: String,
  personalStateHTML: String,
  imgurl: String
});
```

附件

无