

系统设计说明书

项目名称：逐梦校友圈

组 长：

221802222 郑茂基

组 员：

221801104 谢语涵

221801125 许渊博

221801209 蔡建斌

221801215 陈亮亮

221801230 翁炜华

221801231 邹普凡

221801321 黄海翔

221801411 党朝媚

2021 年 04 月 18 日

目录

1. 文档介绍.....	4
1.1 文档目的与范围.....	4
1.2 读者对象.....	4
1.3 术语与缩写解释.....	4
1.4 文档约定.....	4
2. 开发 测试与运行环境.....	5
稳定版 Stable Build (1.05.2103200).....	5
3. 系统概述.....	5
3.1 帖文.....	5
3.2 组局.....	6
3.3 个人中心.....	6
1. 软件系统结构设计.....	6
4.1 软件结构设计.....	6
4.1.1 总体设计结构.....	7
4.1.2 系统设计类图.....	7
4.1.3 系统 ER 图.....	10
5. 功能模块设计.....	10
5.1 功能设计模块概述.....	10
5.2 账号管理模块.....	10
5.2.1 思维导图.....	11
5.2.2 设计说明.....	11
5.3 校友圈模块.....	11
5.3.1 思维导图.....	12
5.3.2 设计说明.....	12
5.3 消息模块.....	13
5.3.1 思维导图.....	13
5.3.2 设计说明.....	13
5.4 发帖模块.....	14
5.4.1 思维导图.....	14
5.4.2 设计说明.....	15
5.5 组局模块.....	15
5.5.1 思维导图.....	16
5.5.2 设计说明.....	17
5.6 个人中心.....	18
5.6.1 思维导图.....	18
5.6.2 设计说明.....	18
6. 接口设计.....	19
6.1 用户管理.....	19
6.1.1 上传身份验证照片.....	19
6.1.2 提交验证信息.....	20
6.1.3 获取审核状态.....	21
6.2 功能目录.....	22
6.2.1 校友圈.....	22

6.2.1 帖子操作.....	27
6.2.2 消息管理.....	32
6.2.3 发帖管理.....	39
6.3 新建组局.....	40
6.4 修改组局信息.....	41
6.5 查看我的组局.....	41
6.6 根据 Type 获取组局.....	42
6.7 组局详情接口.....	43
6.8 参加组局.....	44
6.9 退出组局.....	45
6.10 将参与者移出组局.....	45
6.11 删除组局接口.....	46
6.12 组局搜索.....	46
6.13 组局评论接口.....	47
6.14 获取组局评论列表.....	48
6.15 查询个人信息.....	49
6.16 修改个人信息.....	50
6.17 查询个人收藏.....	51
6.18 删除个人收藏.....	52
6.19 查询个人帖文.....	52
6.20 删除个人帖文.....	53
6.21 查询个人评论.....	54
6.22 删除个人评论.....	55
6.23 查询黑名单.....	55
6.24 删除黑名单.....	56
6.25 查询树洞内容.....	57
6.26 更新树洞内容.....	58
7. 安全设计.....	58
7.1 总览.....	58
7.2 系统安全分层.....	59
7.3 系统（平台）层面的安全.....	59
7.4 框架层面的安全.....	59
7.5 代码层面的安全.....	60
7.6 传输层面的安全.....	60
7.7 客户端层面的安全.....	60
7.8 安全设计结构图.....	60

1.文档介绍

1.1 文档目的与范围

本文档将对逐梦校友圈微信小程序做更为详细分析，对项目做出系统设计概要，以便于对下次的 α 冲刺做好准备。

以下文档，将从功能模块层级设计 接口设计 系统类图 数据库设计 系统安全和权限设计等方面来具体描述系统的设计概要。

1.2 读者对象

本说明书作为用户与该系统软件开发维护人员共同遵守的软件系统概要设计说明。

(1) 产品经理：对后续项目开发中保持文档的完善，依据本文档，对后续项目管理进行任务统筹分配

(2) 开发人员：开发人员通过本文档，了解到系统设计，从而根据此进行编码

(3) 用户：用户通过阅读本文档，能够了解本项目使用环境

1.3 术语与缩写解释

1.4 文档约定

按照以下要求书写文档

1. 标题最多四级标题

(1) 第一级标题，宋体加粗，二号字体

(2) 第二级标题，黑体加粗，三号字体

(3) 第三级标题，宋体加粗，三号字体

(4) 第四级标题，黑体加粗，四号字体

2. 正文字体为宋体小四字体，单倍行距

3. 图标注释为宋体五号字体

4. 正文当中有小点，直接分成(1) (2)，英文括号

5. 段落首行缩进两个字符

2. 开发 测试与运行环境

软件平台名称	发行厂商	版本号	用途
IDEA	JetBrains	2019	java 集成开发环境（服务器端）
微信小程序平台	腾讯	稳定版 Stable Build (1.05.2103200)	微信小程序前端平台搭建

3. 系统概述

3.1 帖文

1. 帖文发布
根据帖文类别，发布对应信息
2. 帖文评论
对帖文进行评论，且对被评论者发布消息
3. 帖文吃瓜
对帖文进行收藏，可以在个人中心里的瓜田查看
4. 帖文点赞
对帖文点赞
5. 帖文赞赏
对帖文帖主贡献自己的一点人品值，可用于进行别的内容
6. 帖文筛选
对应帖文筛选展示
7. 帖文搜索
模糊搜索匹配帖文

3.2 组局

1. 发起组局
限定人数，性别，类别，发布组局
2. 组局筛选
3. 组局搜索
4. 加入组局
已有的组局可以选择对其加入，同时可以在对话框中进行交流

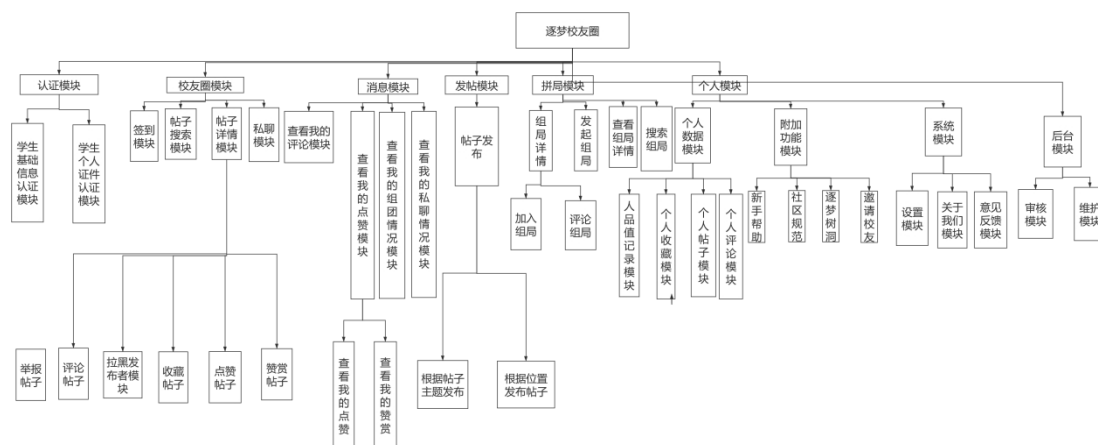
3.3 个人中心

1. 个人设置
个人认证，个人信息更改
2. 瓜田
个人关注帖文页面
3. 我的帖文
我发布过的帖文
4. 我的人品
显示人品记录
5. 树洞
一个可以随意倾吐心事的记录点
6. 新手指引
每个页面功能指引
7. 联系我们
列出联系方式

1. 软件系统结构设计

4.1 软件结构设计

4.1.1 总体设计结构



整个软件共分为 7 个模块，分别是

1. 认证模块
2. 校友圈模块
3. 消息模块
4. 发帖模块
5. 拼局模块
6. 个人模块
7. 后台模块

4.1.2 系统设计类图

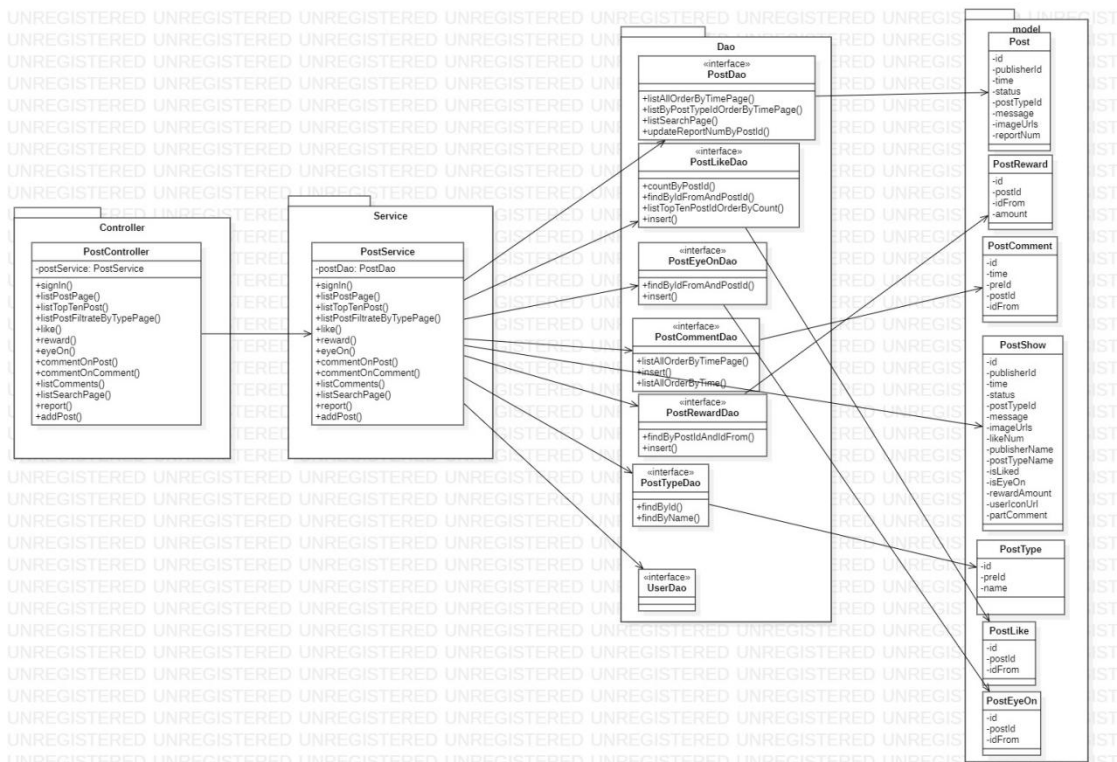
UML 类图用于描述系统所包含的类以及它们之间的相互关系，帮助人们简化对系统的理解，它是系统分析和设计阶段的重要产物，也是系统编码和测试的重要模型依据。一个好的类图设计能够给编码开发一个好的开始。

我们按照系统的功能模块划分，将类图设计部分分为 4 个部分单独设计（用户部分、管理员部分、贴文部分和组局部分），之后进行整合。当然，能够这样进行划分设计的前提是各个模块之间的关联并非很多，往往只是一个模块和另一个模块中的某个类有关联。所以，模块化的好处在于能够更清晰地发现类和拟定类的方法，并且不至于使类图过于庞大而影响最终的阅读。

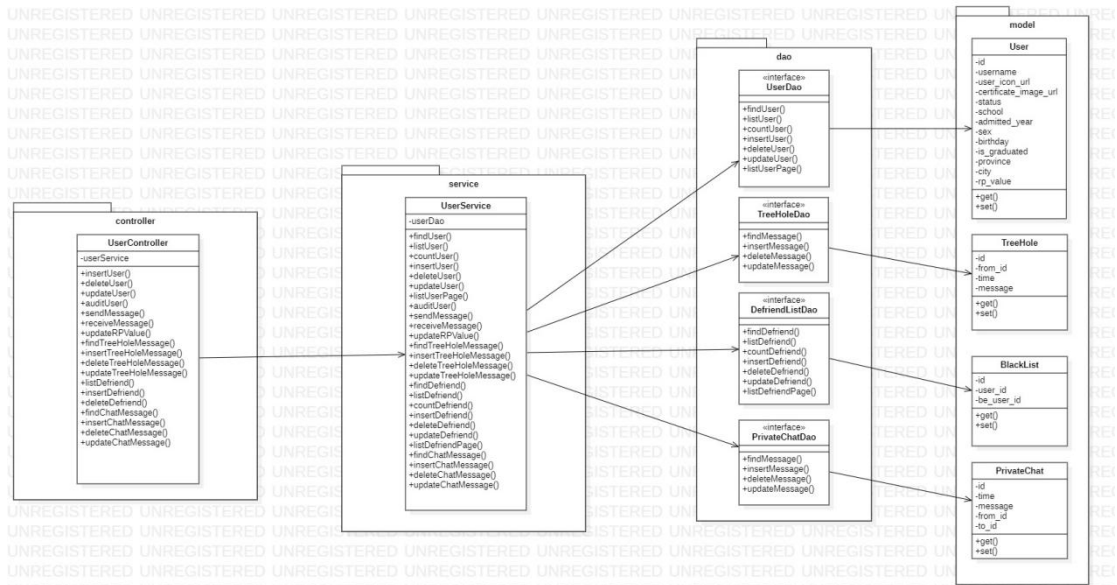
我们在类图设计中结合了开发阶段所利用的语言、框架、操作系统等等因素。在这些开发环境的基础上组织我们的类。依据 SpringBoot 框架，我们将类分为 Model, Dao, Service 和 Controller 四大部分。其中，Model 是系统中的实体类，负责规范数据结构，方便数据的传输和使用；Dao 部分则为与数据库交互的接口，一个 Model 往往对应一个 Dao；Service 类则是系统的处理核心，负责起系统的大部分计算；Controller 类则为前端提供接口。我们遵循设计为开发原则，杜绝无用功。

此外，我们在类图中加入了包，是类图组织有序，更易于观看

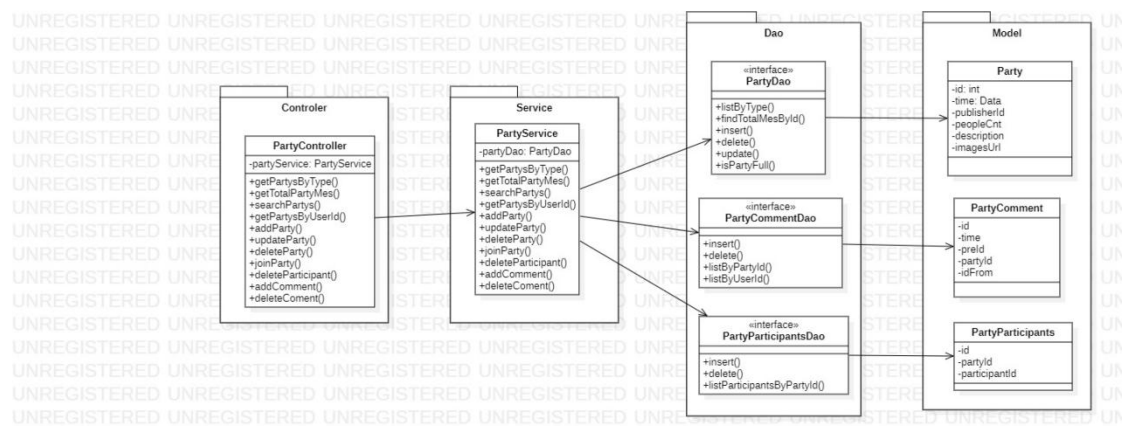
(1) 发帖类图



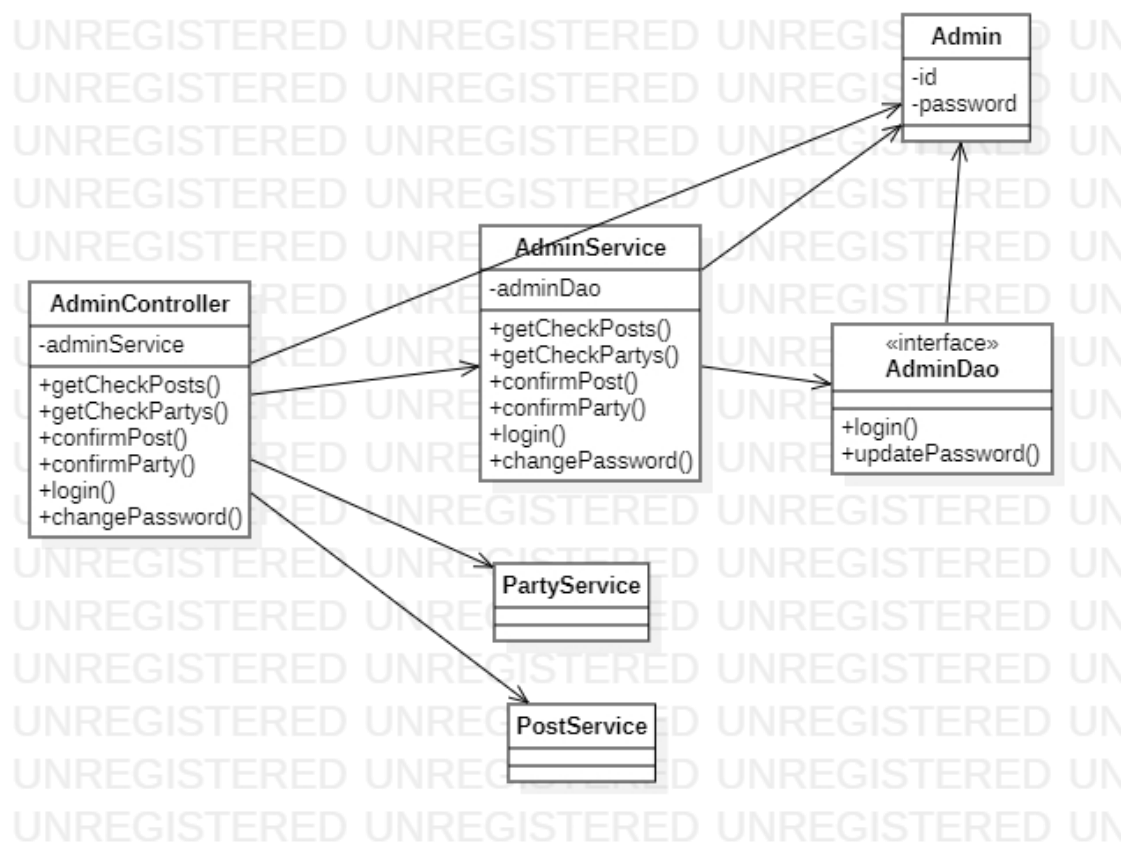
(2) 用户类图



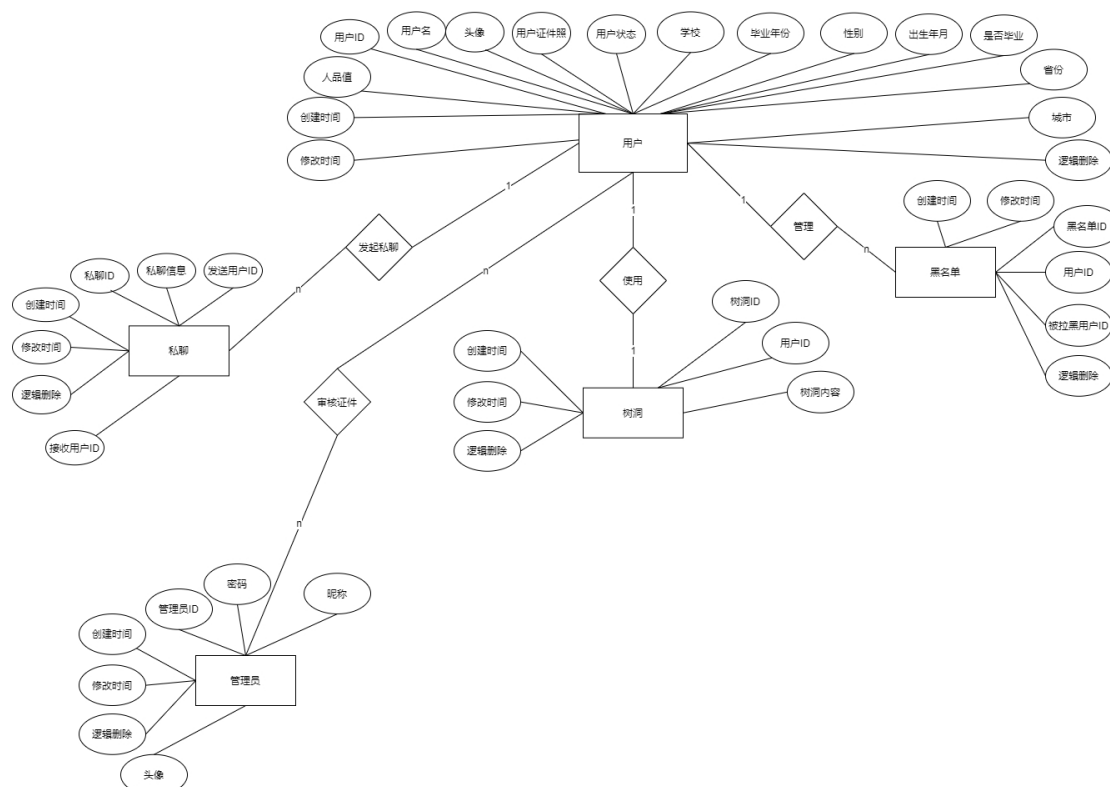
(3) 组局类图



(4) 管理员类图



4.1.3 系统 ER 图



从用户角度出发，寻找用户拥有的属性，寻找用户与其他类（帖文、组局等）的关联，确定之间的关系：1 对多、1 对 1 等，再补充其他类的拥有的属性

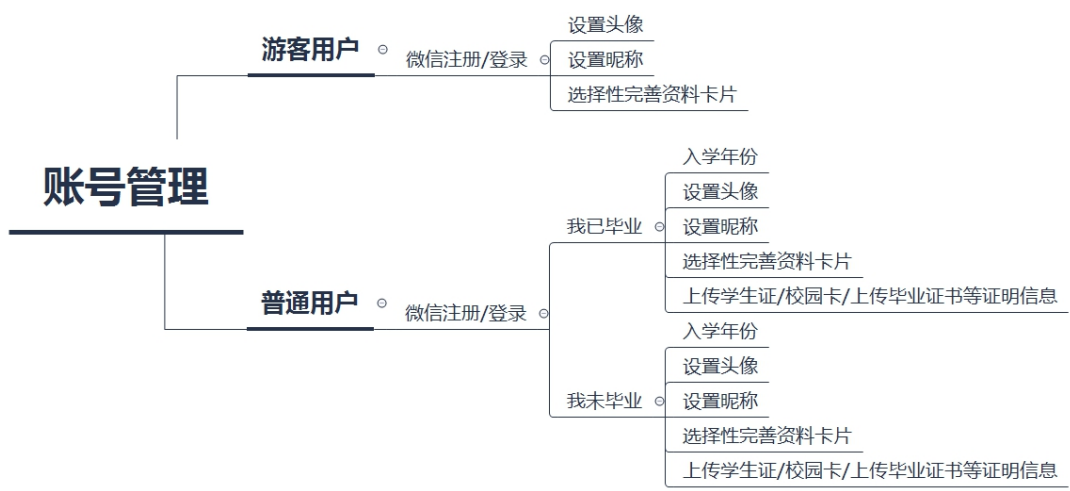
5. 功能模块设计

5.1 功能设计模块概述

该部分对各功能模块，包括账号管理模块 校友圈模块 消息模块 发帖模块 组局模块 个人中心模块共六个功能模块通过绘制各功能模块思维导图 功能模块描述等对系统的功能模块进行具体设计。

5.2 账号管理模块

5.2.1 思维导图

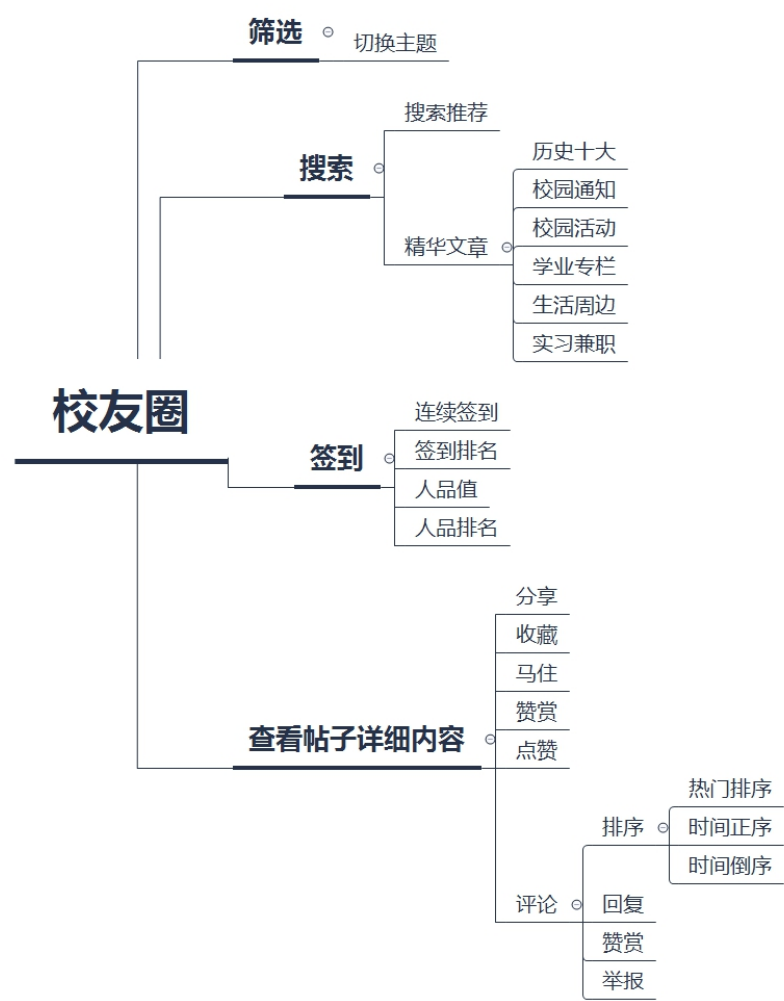


5.2.2 设计说明

- 1. 游客用户
 - 游客用户微信直接登录，初次登录需设置头像 昵称并且选择性完善资料卡片
 - 游客用户不能参与发帖 组局等多项功能
- 2. 普通用户
 - 普通用户微信直接登录，游客用户微信直接登录，初次登录需设置是否毕业 头像 昵称 入学年份并且选择性完善资料卡片
 - 普通用户完善信息越详细将奖励越多的人品值

5.3 校友圈模块

5.3.1 思维导图



5.3.2 设计说明

1. 筛选功能
 - 可以切换最新 投稿 求助 投票 闲置 租房 帮转 兼职招聘 找人 寻物/招领 公告等主题对感兴趣的内容进行筛选
2. 搜索功能
 - 支持用户模糊搜索，搜索框部分下方为搜索推荐部分，按全体用户搜索关键词的热度进行推荐
 - 精华文章包含历史十大 校园通知 校园活动 学业专栏 生活周边 实习兼职六大固定部分
3. 签到功能
 - 根据持续签到天数，用户可以获得相应的人品值
 - 向用户展示连续签到 签到排名 人品值 人品排名四点信息
4. 查看帖子功能

- 用户可以对每个帖子进行分享 收藏 马住 赞赏 点赞 评论
- 其中对于用户的评论其他用户可以进行回复 赞赏 举报或者按照热门排序 时间正序 时间倒序对评论进行排序

5.3 消息模块

5.3.1 思维导图



5.3.2 设计说明

1. 评论功能

- 用户可以对每个帖子进行分享 收藏 马住 赞赏 点赞 评论
- 其中对于用户的评论其他用户可以进行回复 赞赏 举报或者按照热门排序 时间正序 时间倒序对评论进行排序

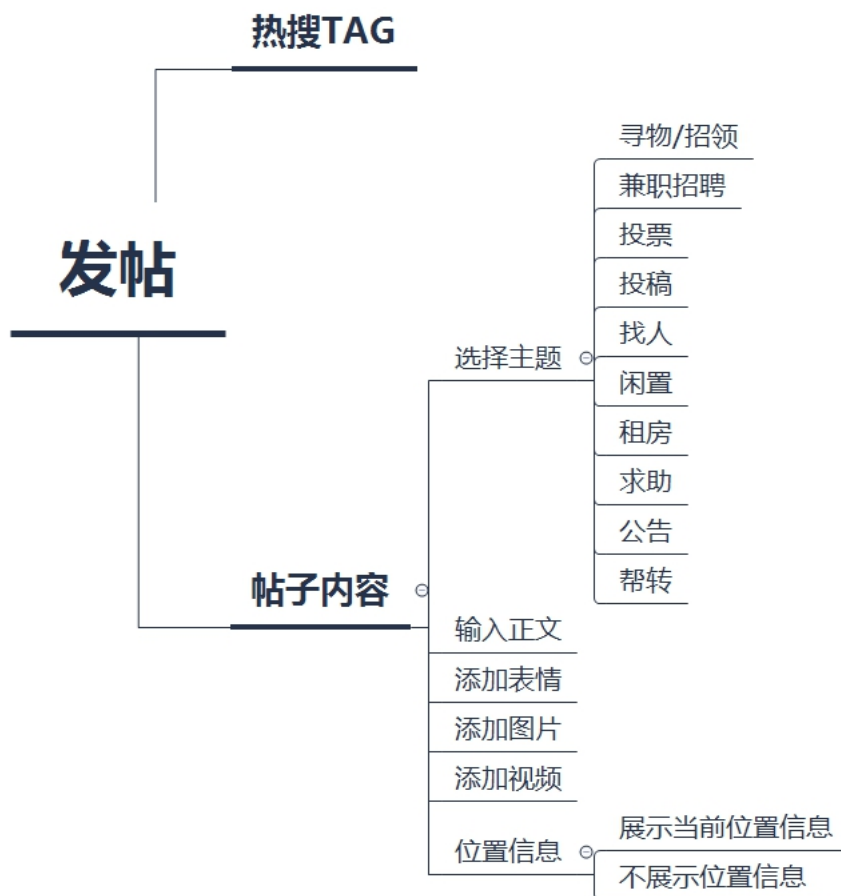
2. 组局消息功能

- 消息列表展示了用户对组局的评论，点击可进入组局详情页面，其中包括组局的活动主题 组内成员 组情 我想去和分享

- 点击帖子列表进入帖子详情，其中同样包括分享 收藏 马住 赞赏 点赞 评论功能
3. 私聊
- 用户可以对私聊按自己的需求进行设置，其中包括接受私聊的方式 同步聊天记录和拉黑聊天记录
 - 通过私聊设置还可以查看自己的拉黑列表，对拉黑用户进行处理
4. 赞/赞赏
- 点击赞/赞赏列表同样进入帖子详情，其中包括分享 收藏 马住 赞赏 点赞 评论功能

5.4 发帖模块

5.4.1 思维导图



5.4.2 设计说明

1. 热搜 Tag 功能

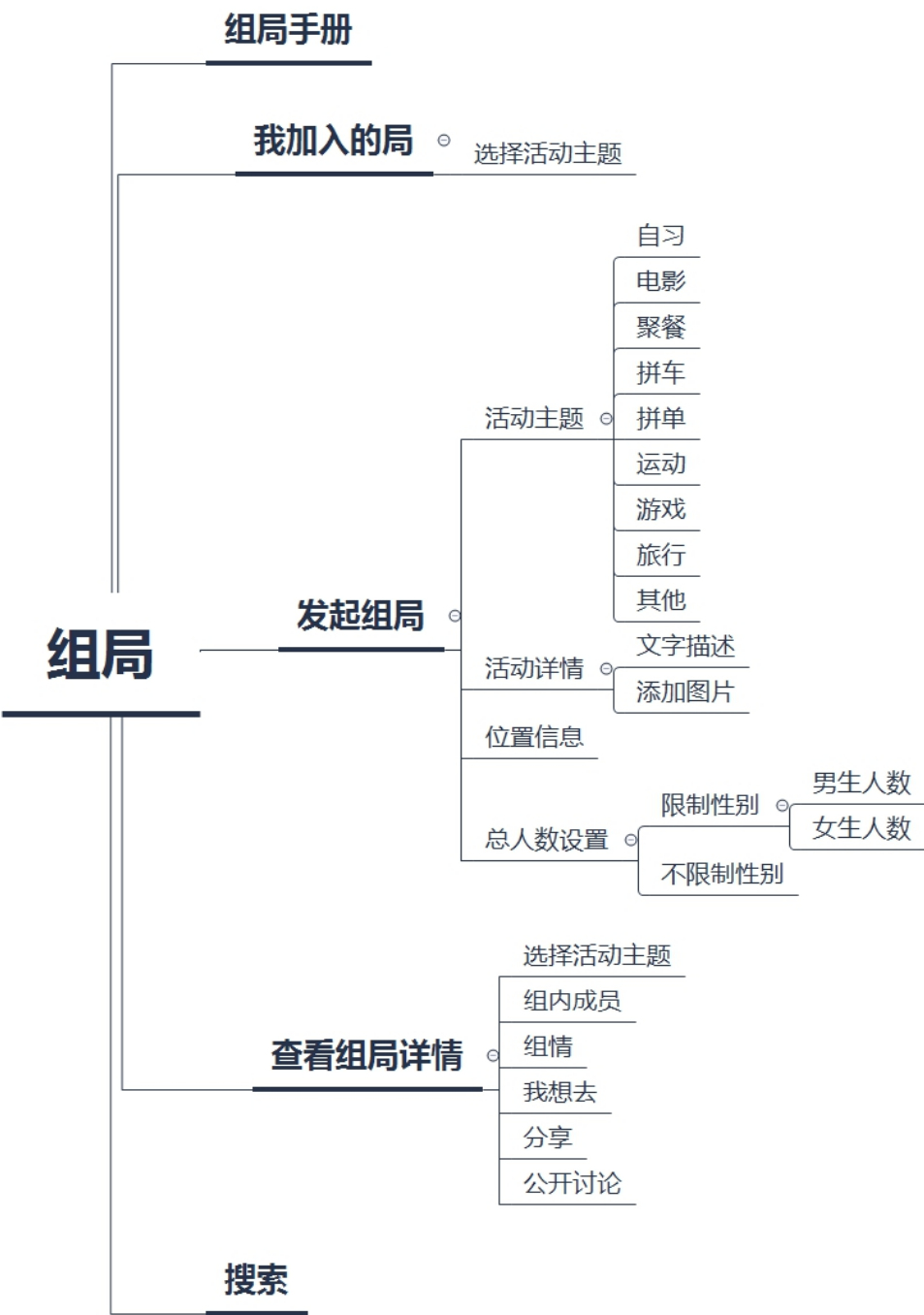
- 按照近期用户使用关键词的热度生成热搜 Tag，用户发帖时可以按需求选择 Tag 使帖子更有趣味性

2. 发帖功能

- 用户需要选择一个或者多个主题，其中主题包括寻物/招领 兼职招聘 投票 投稿 找人 闲置 租房 求助 公告 帮转
- 同时用户还可以以文字 表情 图片 视频等形式来丰富帖子内容
- 用户可以根据需求选择是否展示位置信息
- 每次发帖用户将消耗 100 人品值，人品值过低时不支持发帖功能

5.5 组局模块

5.5.1 思维导图



5.5.2 设计说明

1. 组局手册功能

- 组局手册以图文形式详细介绍了逐梦校友圈组局功能的使用

2. 发起组局功能

- 用户需要选择一个活动的主题,包括自习 电影 聚餐 拼车 拼单 运动 游戏 旅行 其他
- 用户需要描述活动详情,其中可以包括文字描述和图片描述
- 用户需要设置活动的总人数,可以限制性别
- 每次发帖用户将消耗 200 人品值,人品值过低时不支持组局功能

3. 查看组局功能

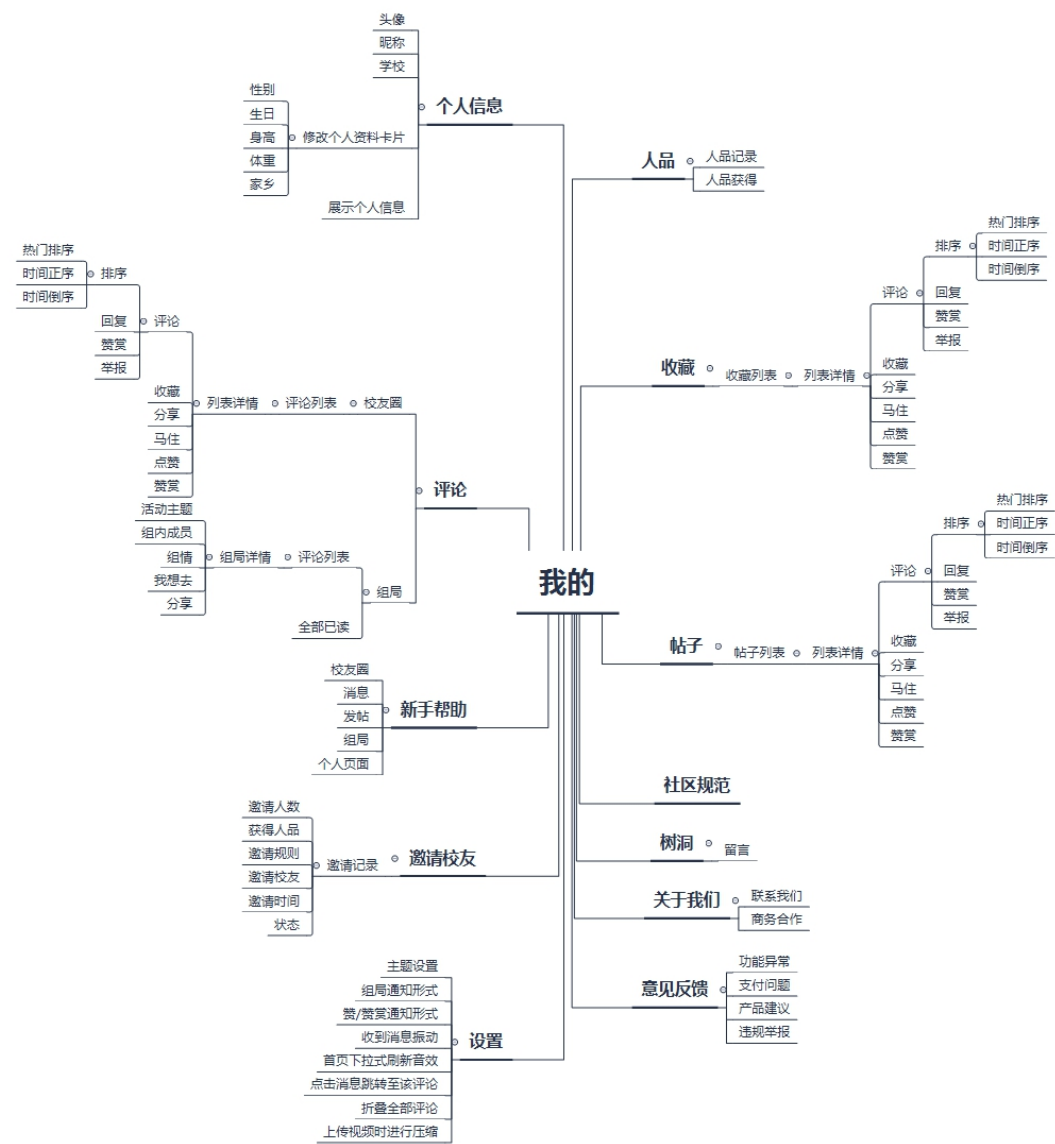
- 用户可以选择要查看的某类型的组局
- 组局的详情包括活动主题 组内成员 组情 我想去和分享

4. 搜索功能

- 用户可以根据需求输入关键词来查找内容中有该关键字的局

5.6 个人中心

5.6.1 思维导图



5.6.2 设计说明

1. 个人信息功能

- 个人信息包括头像 昵称 学校 性别 生日 身高 体重 家乡
- 用户可以更改个人资料卡片，但是性别一旦确定不能更改

2. 评论功能

- 用户可以对每个帖子进行分享 收藏 马住 赞赏 点赞 评论
 - 其中对于用户的评论其他用户可以进行回复 赞赏 举报或者按照热门排序 时间正序 时间倒序对评论进行排序
3. 新手帮助功能
 - 组局手册以图文形式详细介绍了逐梦校友圈各大功能的使用
 4. 邀请校友功能
 - 根据邀请校友的人数奖励相应的人品值
 5. 设置功能
 - 设置包括主题设置 组局通知形式 赞/赞赏通知形式 收到消息是否振动 首页下拉式刷新音效形式 点击消息是否跳转至该评论 是否折叠全部评论 上传视频时是否进行压缩
 6. 人品功能
 - 人品记录包含所有人品值获得的来源，其中包括在线时长 连续签到天数 分享别人帖子 提供 bug 反馈和意见等获得的人品值
 - 人品获得包含人品值获得的规则 如何使用人品以及一些备注
 -
 7. 我的收藏功能
 - 点击收藏列表进入帖子详情，其中同样包括分享 收藏 马住 赞赏 点赞 评论功能
 8. 我的帖子功能
 - 点击帖子列表进入帖子详情，其中同样包括分享 收藏 马住 赞赏 点赞 评论功能
 9. 树洞功能
 - 用户可以向树洞倾诉自己的内心，也可以记录自己的情感
 10. 关于我们 意见反馈功能
 - 关于我们包括联系我们即联系客服 商务合作以及逐梦校友圈的介绍
 - 意见反馈中向开发者的反馈包括功能异常 支付问题 产品建议等问题的反馈
 - 意见反馈中向微信平台投诉包括违规举报

6. 接口设计

6.1 用户管理

6.1.1 上传身份验证照片

简要描述

用户上传身份验证照片接口

- 请求 URL
- http://xx.com/api/identify/img
- 请求方式
- POST
- 数据类型
- JSON

参数 参数名	必选	类型	说明
img	是	file	用户名
返回示例 <pre>{ "code": 0, //返回服务器结果代码 "message": "", //上传图片结果 "data": "" //上传图片服务器 url }</pre>			

- 备注
- 更多返回错误代码请看首页的错误代码描述

6.1.2 提交验证信息

简要描述

提交验证信息接口

请求 URL

http://xx.com/api/user/indentify/submit

请求方式

POST

数据类型

JSON

参数 参数名	必选	类型	说明
username	是	string	用户名
userIconUrl	是	string	上传图片链接
status	是	int	审核状态
school	是	string	学校
admittedYear	是	int	入学年份
sex	是	int	性别
birthday	是	Date	生日
isGraduated	是	int	毕业状态
province	是	string	省份
city	是	string	城市

参数 参数名	必选	类型	说明
返回示例			
<pre>{ "code": 0, //返回服务器结果代码 "message": "", //服务器处理结果信息 "data": "" }</pre>			

备注
更多返回错误代码请看首页的错误代码描述

6.1.3 获取审核状态

简要描述

获取审核状态接口

请求 URL

http://xx.com/api/user/identify/status

请求方式

POST

数据类型

JSON

参数

参数名	必选	类型	说明
id	是	string	用户 id
返回示例			
<pre>{ "code": 0, //返回服务器结果代码 "message": "", //返回服务器处理结果信息 "data": 0 //审核状态值 }</pre>			
返回参数说明			
参数名		类型	说明
groupid		int	用户组 id, 1: 超级管理员; 2: 普通用户

备注
更多返回错误代码请看首页的错误代码描述

6.2 功能目录

6.2.1 校友圈

6.2.1.1 签到

简要描述
校友圈签到接口

请求 URL
`http://xx.com/api/user/alumnicircle/signin`

请求方式
POST

数据类型
JSON

参数 参数名	必选	类型	说明
id	是	Long	用户 id

返回示例

```
{
  "code": 0, //返回服务器结果代码
  "message": "",
  "data": {
  }
}
```

备注
更多返回错误代码请看首页的错误代码描述

6.2.1.2 获取校友圈首页帖子

简要描述
获取校友圈帖子接口

请求 URL
`http://xx.com/api/alumnicircle/posts/all`

请求方式
POST

数据类型
JSON

参数 参数名	必选	类型	说明
-----------	----	----	----

参数 参数名	必选	类型	说明
userId	是	Long	用户 ID
pageNum	是	int	帖子页数
pageSize	是	int	每页帖子数
返回示例 <pre> { "code": 0, //返回服务器结果代码 "message": "", //服务器处理结果信息 "data": { data: { publisherId: 0, publisherName: "", postType: "", message: "", imageUrls: [], gmtCreate: "", isEyeOn: "", eyeOnNum: 0, isLike: 0, likeNum: 0, rewardNum: 0, comments: [{ commentId: 0, commentUserId: 0, commentUsername: "", message: "", preId: 0, },] } } }</pre>			

备注
 更多返回错误代码请看首页的错误代码描述

6.2.1.3 用户搜索帖子

简要描述
 用户搜索帖子接口

请求 URL

http://xx.com/api/alumnicycle/search

请求方式

POST

数据类型

JSON

参数 参数名	必选	类型	说明
content	是	string	搜索内容
<div>返回示例</div> <pre>{ "code": 0, //返回服务器结果代码 "message": "", //返回服务器接口结果信息 "data": { data: { publisherId: 0, publisherName: "", postType: "", message: "", imageUrls: [], gmtCreate: "", isEyeOn: "", eyeOnNum: 0, isLike: 0, likeNum: 0, rewardNum: 0, comments: [{ commentId: 0, commentUserId: 0, commentUsername: "", message: "", preId: 0, },], }, }, }</pre>			

备注

更多返回错误代码请看首页的错误代码描述

6.2.1.4 切换主题接口

简要描述

帖子显示切换主题接口

请求 URL

http://xx.com/api/alumnicircle/changetheme

请求方式

POST

数据类型

JSON

参数 参数名	必选	类型	说明
typeId	是	Long	板块 ID
返回示例			
<pre>{ "code": 0, //返回服务器结果代码 "message": "", //返回服务器接口结果信息 "data": { data: { publisherId: 0, publisherName: "", postType: "", message: "", imageUrls: [], gmtCreate: "", isEyeOn: "", eyeOnNum: 0, isLike: 0, likeNum: 0, rewardNum: 0, comments: [{ commentId: 0, commentUserId: 0, commentUsername: "", message: "", preId: 0, },], } } }</pre>			

备注
更多返回错误代码请看首页的错误代码描述

6.2.1.5 获取热门帖子

简要描述
获取十大热门帖子接口
请求 URL
http://xx.com/api/alumnicircle/heatposts
请求方式
POST
数据类型
JSON

参数 参数名	必选	类型	说明
userId	是	Long	用户 ID
<div>返回示例</div> <pre>{ "code": 0, //返回服务器结果代码 "message": "", //服务器处理结果信息 "data": { data: { publisherId: 0, publisherName: "", postType: "", message: "", imageUrls: [], gmtCreate: "", isEyeOn: "", eyeOnNum: 0, isLike: 0, likeNum: 0, rewardNum: 0, comments: [{ commentId: 0, commentUserId: 0, commentUsername: "", message: "", preId: 0, },] } } }</pre>			

参数 参数名	必选	类型	说明

备注
更多返回错误代码请看首页的错误代码描述

6.2.1 帖子操作

6.2.1.1 帖子点赞接口

简要描述
帖子点赞接口
请求 URL
`http://xx.com/api/alumnicycle/posts/like`
请求方式
POST
数据类型
JSON

参数 参数名	必选	类型	说明
userId	是	Long	点赞用户 id
postId	是	Long	被点赞帖子 id
返回示例 <pre>{ "code": 0, //返回服务器结果代码 "message": "", //返回服务器接口结果信息 "data": "" }</pre>			

备注
更多返回错误代码请看首页的错误代码描述

6.2.1.2 帖子评论接口

简要描述
帖子评论接口
请求 URL
`http://xx.com/api/alumnicycle/posts/comment`
请求方式

POST

数据类型

JSON

参数 参数名	必选	类型	说明
content	是	string	评论内容
userId	是	Long	评论用户 ID
postId	是	Long	帖文 ID
preId	是	Long	父级评论 ID
返回示例			
<pre>{ "code": 0, //返回服务器结果代码 "message": "", //返回服务器接口结果信息 "data": "" }</pre>			

备注

更多返回错误代码请看首页的错误代码描述

6.2.1.3 帖子收藏接口

简要描述

帖子收藏接口

请求 URL

http://xx.com/api/alumnicycle/posts/collect

请求方式

POST

数据类型

JSON

参数

参数名	必选	类型	说明
userId	是	Long	收藏用户 ID
postId	是	Long	用户名
返回示例			
<pre>{ "code": 0, //返回服务器结果代码 "message": "", //返回服务器接口结果信息 "data": "", }</pre>			

备注

更多返回错误代码请看首页的错误代码描述

6.2.1.4 用户拉黑接口

简要描述

用户拉黑接口

请求 URL

http://xx.com/api/block

请求方式

POST

数据类型

JSON

参数 参数名	必选	类型	说明
userIdFrom	是	Long	用户 ID
userIdTo	是	Long	被拉黑用户 ID
isBlock	是	int	是否拉黑用户
返回示例			
<pre>{ "code": 0, //返回服务器结果代码 "message": "", //返回服务器接口结果信息 "data": "" }</pre>			

备注

更多返回错误代码请看首页的错误代码描述

6.2.1.5 帖子赞赏接口

简要描述

用户注册接口

请求 URL

http://xx.com/api/alumnicycle/posts/reward

请求方式

POST

数据类型

JSON

参数 参数名	必选	类型	说明
userIdFrom	是	Long	打赏用户 ID
userIdTo	是	Long	被打赏用户 ID
postId	是	Long	被打赏帖文 ID
amount	是	int	打赏人气值数量

参数 参数名	必选	类型	说明
返回示例			
<pre>{ "code": 0, //返回服务器结果代码 "message": "", //服务器处理结果信息 "data": "", }</pre>			

备注

更多返回错误代码请看首页的错误代码描述

6.2.1.6 帖子举报接口

简要描述

帖子举报接口

请求 URL

http://xx.com/api/alumnicycle/posts/tipoff

请求方式

POST

数据类型

JSON

参数 参数名	必选	类型	说明
postId	是	Long	被举报帖子 id
userId	是	Long	举报用户 id
返回示例			
<pre>{ "code": 0, //返回服务器结果代码 "message": "", //服务器处理结果信息 "data": "", }</pre>			

备注

更多返回错误代码请看首页的错误代码描述

6.2.1.7 获取发帖人信息

简要描述

获取发帖人信息接口

请求 URL

http://xx.com/api/alumnicycle/posts/publishermsg

请求方式

POST

数据类型

JSON

参数 参数名	必选	类型	说明
publisherId	是	Long	发帖人 ID
返回示例 <pre>{ "code": 0, //返回服务器结果代码 "message": "", //服务器处理结果信息 "data": { userName: "", age: 0, isGraduated: 0, province: "", city: "", university: "", iconUrl: "", } }</pre>			

备注

更多返回错误代码请看首页的错误代码描述

6.2.1.8 获取帖子评论列表

简要描述

获取帖子评论列表接口

请求 URL

http://xx.com/api/alumnicycle/posts/commentlist

请求方式

POST

数据类型

JSON

参数 参数名	必选	类型	说明
postId	是	Long	帖子 ID
返回示例 <pre>{ "code": 0, //返回服务器结果代码 "message": "", //服务器处理结果信息 "data": [</pre>			

参数 参数名	必选	类型	说明
<pre> { commentId:0, commentUserId:0, commentUsername:"", message:"", preId:0, },], }</pre>			

备注

更多返回错误代码请看首页的错误代码描述

6.2.2 消息管理

6.2.2.1 获取个人相关评论

简要描述

获取用户个人相关评论接口

请求 URL

http://xx.com/api/message/comment

请求方式

POST

数据类型

JSON

参数 参数名	必选	类型	说明
userId	是	Long	用户名
返回示例 <pre> { "code": 0, //返回服务器结果代码 "message": "", //服务器处理结果信息 "data": [{ userId:0, userName:"", message:"", commentId:0, preId:0, quote:"", </pre>			

参数 参数名	必选	类型	说明
<pre>gmtCreate:"", }] }</pre>			

备注
更多返回错误代码请看首页的错误代码描述

6.2.2.2 获取个人相关赞赏

简要描述
获取个人相关赞赏接口

请求 URL
`http://xx.com/api/message/myreward`

请求方式
POST

数据类型
JSON

参数 参数名	必选	类型	说明
userId	是	Long	用户 ID
返回示例 <pre>{ "code": 0, //返回服务器结果代码 "message": "", //服务器处理结果信息 "data": [rewarderId:0, rewarderName:"", rewardAmount:0, postId:0, postContent:"", rewardTime:"",] }</pre>			

备注
更多返回错误代码请看首页的错误代码描述

6.2.2.3 获取个人相关点赞

简要描述

获取个人相关点赞接口

请求 URL

http://xx.com/api/message/like

请求方式

POST

数据类型

JSON

参数 参数名	必选	类型	说明
userId	是	id	用户 id
返回示例 <pre>{ "code": 0, //返回服务器结果代码 "message": "", //服务器处理结果信息 "data": [{ postId: "", message: "", likeNum: "", gmtCreate: "" }] }</pre>			

备注

更多返回错误代码请看首页的错误代码描述

6.2.2.4 获取个人相关拼局信息

简要描述

获取个人相关拼局信息接口

请求 URL

http://xx.com/api/message/myparty

请求方式

POST

数据类型

JSON

参数 参数名	必选	类型	说明
-----------	----	----	----

参数 参数名	必选	类型	说明
userId	是	Long	用户 Id
返回示例 <pre> { "code": 0, //返回服务器结果代码 "message": "", //服务器处理结果信息 "data": [partyId:0, content:"", users:[{ userName:"", }], gmtCreate:"",] }</pre>			

备注

更多返回错误代码请看首页的错误代码描述

6.2.2.5 获取私聊用户列表

简要描述

获取私聊用户列表接口

请求 URL

http://xx.com/api/message/chat/list

请求方式

POST

数据类型

JSON

参数 参数名	必选	类型	说明
userId	是	Long	用户 ID
返回示例 <pre> { "code": 0, //返回服务器结果代码 "message": "", //服务器处理结果信息 "data": [{ dialogUserId:"", dialogUserName:"", }] }</pre>			

参数 参数名	必选	类型	说明
<pre> imageUrl:"", latestTime:"", latestMessage:"", }] }</pre>			

备注

更多返回错误代码请看首页的错误代码描述

6.2.2.6 清空私聊列表

简要描述

清空私聊列表接口

请求 URL

http://xx.com/api/message/chat/clearlist

请求方式

POST

数据类型

JSON

参数 参数名	必选	类型	说明
userId	是	Long	用户 Id
返回示例 <pre> { "code": 0, //返回服务器结果代码 "message": "", //服务器处理结果信息 "data": "" }</pre>			

备注

更多返回错误代码请看首页的错误代码描述

6.2.2.7 用户发送消息

简要描述

用户发送消息接口

请求 URL

http://xx.com/api/message/chat/send

请求方式

POST

数据类型

JSON

参数 参数名	必选	类型	说明
userIdFrom	是	Long	发送用户 ID
userIdTo	是	Long	接受用户 ID
content	是	string	消息内容
返回示例 <pre>{ "code": 0, //返回服务器结果代码 "message": "", //服务器处理结果信息 "data": "" }</pre>			

备注

更多返回错误代码请看首页的错误代码描述

6.2.2.8 用户接受消息

简要描述

用户接受消息接口

请求 URL

http://xx.com/api/message/chat/receive

请求方式

POST

数据类型

JSON

参数 参数名	必选	类型	说明
userIdFrom	是	Long	发送用户 ID
userIdTo	是	Long	接受用户 ID
pageNum	否	int	页数
pageSize	否	int	每页容量
返回示例 <pre>{ "code": 0, //返回服务器结果代码 "message": "", //服务器处理结果信息 "data": [{ time: "", content: "", </pre>			

参数 参数名	必选	类型	说明
isFromMe:0 }] }			

备注
更多返回错误代码请看首页的错误代码描述

6.2.2.9 用户相关帖子动态获取

简要描述
用户相关帖子动态获取接口
请求 URL
`http://xx.com/api/message/poststatus`
请求方式
POST
数据类型
JSON

参数 参数名	必选	类型	说明
userId	是	Long	用户 Id
返回示例 <pre>{ "code": 0, //返回服务器结果代码 "message": "", //服务器处理结果信息 "data": [postId: "", message: "", type: 1, time: "", userNameFrom: "",] }</pre>			

备注
更多返回错误代码请看首页的错误代码描述

6.2.3 发帖管理

6.2.3.1 发布帖子接口

简要描述

发布帖子接口

请求 URL

http://xx.com/api/post/publish

请求方式

POST

数据类型

JSON

参数 参数名	必选	类型	说明
userId	是	Long	用户 Id
postTheme	是	Long	帖子主题
message	是	string	帖子内容
imgUrls	是	Array	图片数组
返回示例			
<pre>{ "code": 0, //返回服务器结果代码 "message": "", //服务器处理结果信息 "data": "" }</pre>			

备注

更多返回错误代码请看首页的错误代码描述

6.2.3.2 帖文上传图片

简要描述

帖文上传图片接口

请求 URL

http://xx.com/api/post/imgupload

请求方式

POST

数据类型

JSON

参数 参数名	必选	类型	说明
-----------	----	----	----

参数 参数名	必选	类型	说明
img	是	File	图片
返回示例 <pre>{ "code": 0, //返回服务器结果代码 "message": "", //服务器处理结果信息 "data": ["", //上传图片的 URL] }</pre>			

备注
更多返回错误代码请看首页的错误代码描述

6.3 新建组局

- 简要描述
- 用户注册接口
- 请求 URL
- http://xx.com/api//alumnicycle/party/add
- 请求方式
- POST

参数 参数名	必选	类型	说明
description	是	string	组局文字内容
image	是	File	组局图片
peopleCnt	是	int	组局人数
返回示例 <pre>{ "code": 0, //返回服务器结果代码 "message": "", //返回服务器接口结果信息 }</pre>			
返回参数说明 参数名	类型	说明	
code	int	创建成功: 0 创建失败: 1	

- 备注
- 更多返回错误代码请看首页的错误代码描述

6.4 修改组局信息

- 简要描述
- 修改组局信息
- 请求 URL
- `http://xx.com/api//alumnicycle/party/add`
- 请求方式
- POST

参数 参数名	必选	类型	说明
partyId	是	Long	组局 ID
description	是	string	组局文字内容
image	是	File	组局图片
peopleCnt	是	int	组局人数
返回示例			
<pre>{ "code": 0, //返回服务器结果代码 "message": "", //返回服务器接口结果信息 }</pre>			
返回参数说明 参数名	类型	说明	
code	int	创建成功: 0 创建失败: 1	

- 备注
- 更多返回错误代码请看首页的错误代码描述

6.5 查看我的组局

- 简要描述
- 查看我的组局接口
- 请求 URL
- `http://xx.com/api/alumnicycle/party/type`
- 请求方式
- GET
- 数据类型
- JSON

参数

无

返回示例

```
{
  "code": 0, //返回服务器结果代码
  "message": "", //返回服务器接口结果信息
  "data": [{
    "description": "",
    "publisher": {
      "image": "image url",
      "id": 123,
      "username": "张三"
    },
    "peopleCnt": 6,
    "nowPeopleCnt": 3,
    "partyType": "拼单",
    "gmtCreate": "2020/4/23 10:00:12"
  }]
}
```

返回参数说明

参数名

类型

说明

code

int

查询成功: 0 查询失败: 1

- 备注
- 更多返回错误代码请看首页的错误代码描述

6.6 根据 Type 获取组局

- 简要描述
- 根据 Type 获取组局
- 请求 URL
- <http://xx.com/api/alumnicycle/party/type>
- 请求方式
- GET
- 数据类型
- JSON
- 参数
-

参数名	必选	类型	说明
partyType	是	string	拼局类别, 当传入为"all"时返回所有组局信息
返回示例			
{ "code": 0, //返回服务器结果代码 "message": "", //返回服务器接口结果信息 "data": [{			

```
        "description": "",
        "publisher": {
            "image": "image url",
            "id": 123,
            "username": "张三"
        },
        "peopleCnt": 6,
        "nowPeopleCnt": 3,
        "partyType": "拼单",
        "gmtCreate": "2020/4/23 10:00:12"
    }
]
}
```

返回参数说明 参数名	类型	说明
code	int	查询成功：0 查询失败：1

- 备注
- 更多返回错误代码请看首页的错误代码描述

6.7 组局详情接口

简要描述

组局详情接口

请求 URL

http://xx.com/api/alumnicycle/party/partymes

请求方式

GET

参数 参数名	必选	类型	说明
partyId	是	Long	拼局 ID

返回示例

```
{
    "code": 0,
    "data": [
        "description": "",
        "message": "", //返回服务器接口结果信息
        "publisher": {
            "image": "url",
            "id": 123,
            "username": "张三"
        },
        "peopleCnt": 6,
```

```
        "nowPeopleCnt": 3,
        "partyType": "拼单",
        "gmtCreate": "2020/4/23 10:00:12",
        "participants": [{
            "image": "image url",
            "id": 123,
            "username": "张三"
        }]
    ]
}
```

返回参数说明 参数名	类型	说明
code	int	查询成功：0 查询失败：1

备注
更多返回错误代码请看首页的错误代码描述

6.8 参加组局

简要描述
参加组局接口
请求 URL
`http://xx.com/api/alumnicycle/party/join`
请求方式
POST

参数 参数名	必选	类型	说明
partyId	是	Long	拼局 ID
返回示例			
<pre>{ "code": 0, //返回服务器结果代码 "message": "", //返回服务器接口结果信息 }</pre>			
返回参数说明 参数名	类型	说明	
code	int	查询成功：0 查询失败：1	

备注
更多返回错误代码请看首页的错误代码描述

6.9 退出组局

简要描述

参加组局接口

请求 URL

http://xx.com/api/alumnicycle/party/join

请求方式

POST

参数 参数名	必选	类型	说明
partyId	是	Long	拼局 ID
返回示例 <pre>{ "code": 0, //返回服务器结果代码 "message": "", //返回服务器接口结果信息 }</pre>			
返回参数说明 参数名	类型	说明	
code	int	查询成功：0 查询失败：1	

备注

更多返回错误代码请看首页的错误代码描述

6.10 将参与者移出组局

简要描述

将参与者移出组局接口

请求 URL

http://xx.com/api/alumnicycle/party/moveoff

请求方式

POST

参数 参数名	必选	类型	说明
partyId	是	Long	拼局 ID
userId	是	Long	被移除的用户的 ID
返回示例 <pre>{ "code": 0, //返回服务器结果代码 "message": "", //返回服务器接口结果信息 }</pre>			

}		
返回参数说明 参数名	类型	说明
code	int	移除成功：0 移除失败：1

备注

更多返回错误代码请看首页的错误代码描述

6.11 删除组局接口

简要描述

删除组局接口

请求 URL

http://xx.com/api/alumnicycle/party/delete

请求方式

POST

参数 参数名	必选	类型	说明
partyId	是	Long	待删除拼局 ID
返回示例 <pre>{ "code": 0, //返回服务器结果代码 "message": "", //返回服务器接口结果信息 }</pre>			
返回参数说明 参数名	类型	说明	
error	int	删除成功：0 删除失败：1	

备注

更多返回错误代码请看首页的错误代码描述

6.12 组局搜索

简要描述

-搜索组局接口

请求 URL

http://xx.com/api/alumnicycle/party/search

请求方式

POST

参数 参数名	必选	类型	说明
-----------	----	----	----

message	是	string	搜索字段
返回示例 <pre> { "code": 0, //返回服务器结果代码 "message": "", //返回服务器接口结果信息 "data": [{ "description": "", "publisher": { "image": "image url", "id": 123, "username": "张三" }, "partyType": "拼单", "gmtCreate": "2020/4/23 10:00:12" }] }</pre>			
返回参数说明		类型	说明
参数名			
error		int	查询成功: 0 查询失败: 1

备注

更多返回错误代码请看首页的错误代码描述

6.13 组局评论接口

简要描述

组局评论接口

请求 URL

http://xx.com/api/alumnicycle/party/comment

请求方式

POST

参数 参数名	必选	类型	说明
content	是	string	评论内容
userId	是	Long	评论用户 ID
partyId	是	Long	组局 ID
preId	是	Long	父级评论 ID
返回示例 <pre> { "code": 0, //返回服务器结果代码 "message": "", //返回服务器接口结果信息</pre>			

<pre> "data":"" } </pre>		
返回参数说明 参数名	类型	说明
error	int	查询成功：0 查询失败：1

备注

更多返回错误代码请看首页的错误代码描述

6.14 获取组局评论列表

简要描述

组局评论接口

请求 URL

http://xx.com/api/alumnicycle/party/commentlsit

请求方式

POST

参数 参数名	必选	类型	说明
partyId	是	Long	组局 ID
返回示例 <pre> { "code": 0, //返回服务器结果代码 "message": "", //服务器处理结果信息 "data": [{ commentId:0, commentUserId:0, commentUsername:"", message:"", preId:0, "gmtCreate": "2020/4/23 10:00:12" },], } </pre>			
返回参数说明 参数名	类型	说明	
error	int	查询成功：0 查询失败：1	

备注

更多返回错误代码请看首页的错误代码描述

6.15 查询个人信息

简要描述

个人信息读取接口

请求 URL

`http://xx.com/api/alumnicycle/mine/data/select`

请求方式

GET

数据类型

JSON

参数

无

返回示例

```
{
  "code": 0, //返回服务器结果代码
  "message": "", //返回服务器接口结果信息
  "data": {
    "admittedYear": 0,
    "birthday": "2021-04-23",
    "certificateImageUrl": "string",
    "city": "string",
    "deleted": 0,
    "gmtCreate": "2021-04-23T08:13:39.622Z",
    "gmtModified": "2021-04-23T08:13:39.622Z",
    "id": 0,
    "isGraduated": 0,
    "province": "string",
    "rpValue": 0,
    "school": "string",
    "sex": 0,
    "status": 0,
    "userIconUrl": "string",
    "username": "string"
  }
}
```

备注

更多返回错误代码请看首页的错误代码描述

6.16 修改个人信息

简要描述

个人信息修改接口

请求 URL

http://xx.com/api/alumnicycle/mine/data/update

请求方式

POST

数据类型

JSON

参数: User 对象

参数名	必选	类型	说明
admittedYear	是	integer(\$int32)	毕业年份
birthday	是	string(\$date-time)	出生年月
certificateImageUrl	是	string	用户证件照
city	是	string	城市
deleted	是	integer	逻辑删除
gmtCreate	是	string(\$date-time)	创建时间
gmtModified	是	string(\$date-time)	修改时间
id	是	integer(\$int64)	用户 ID(使用微信提供的 openID)
isGraduated	是	integer(\$int32)	是否毕业
province	是	string	省份
rpValue	是	integer(\$int64)	人品值
school	是	string	用户学校
sex	是	integer(\$int32)	性别(0 为女, 1 为男, 2 为无)
status	是	integer(\$int32)	用户状态(0 为未审核, 1 为已审核)
userIconUrl	是	string	用户头像
username	是	string	用户名

返回示例

```
{
  "code": 0, //返回服务器结果代码
  "message": "", //服务器处理结果信息
  "data": {
    "admittedYear": 0,
    "birthday": "2021-04-23",
    "certificateImageUrl": "string",
    "city": "string",
    "deleted": 0,
    "gmtCreate": "2021-04-23T08:13:39.622Z",
    "gmtModified": "2021-04-23T08:13:39.622Z",
    "id": 0,
```

```
        "isGraduated": 0,
        "province": "string",
        "rpValue": 0,
        "school": "string",
        "sex": 0,
        "status": 0,
        "userIconUrl": "string",
        "username": "string"
    }
}
```

备注

更多返回错误代码请看首页的错误代码描述

6.17 查询个人收藏

简要描述

个人收藏查询接口

请求 URL

http://xx.com/api/alumnicycle/mine/collect/list

请求方式

GET

数据类型

JSON

参数

无

返回示例

```
{
  "code": 0, //返回服务器结果代码
  "message": "", //服务器处理结果信息
  "data": [
    {
      "deleted": 0,
      "gmtCreate": "2021-04-23T10:37:19.038Z",
      "gmtModified": "2021-04-23T10:37:19.038Z",
      "id": 0,
      "idFrom": 0,
      "postId": 0
    }
  ]
}
```

备注

更多返回错误代码请看首页的错误代码描述

6.18 删除个人收藏

简要描述

个人收藏删除接口

请求 URL

http://xx.com/api/alumnicycle/mine/collect/deleted

请求方式

POST

数据类型

JSON

参数: Post_like 对象

参数名	必选	类型	说明
deleted	是	integer(\$int32)	逻辑删除
gmtCreate	是	string(\$date-time)	创建时间
gmtModifiedstring	是	(\$date-time)	最近一次修改时间
id	是	integer(\$int64)	ID
idFrom	是	integer(\$int64)	点赞用户 ID, 外键
postId	是	integer(\$int64)	被赞帖子 ID, 外键

返回示例

```
{
  "code": 0, //返回服务器结果代码
  "message": "", //服务器处理结果信息
  "data": 0 //0 失败, 1 成功
}
```

备注

更多返回错误代码请看首页的错误代码描述

6.19 查询个人帖文

简要描述

个人帖文查询接口

请求 URL

http://xx.com/api/alumnicycle/mine/post/list

请求方式

GET

数据类型

JSON

参数

无

返回示例

```
{
  "code": 0, //返回服务器结果代码
  "message": "", //服务器处理结果信息
  "data": [
    {
      "deleted": 0,
      "gmtCreate": "2021-04-23T11:01:33.765Z",
      "gmtModified": "2021-04-23T11:01:33.765Z",
      "id": 0,
      "imageUrls": "string",
      "message": "string",
      "postTypeId": 0,
      "publisherId": 0,
      "status": 0
    }
  ]
}
```

备注

更多返回错误代码请看首页的错误代码描述

6.20 删除个人帖文

简要描述

个人帖文删除接口

请求 URL

<http://xx.com/api/alumnicycle/mine/post/deleted>

请求方式

POST

数据类型

JSON

参数: Post 对象

参数名	必选	类型	说明
deleted	是	integer(\$int32)	逻辑删除
gmtCreate	是	string(\$date-time)	创建时间
gmtModified	是	string(\$date-time)	最近一次修改时间
id	是	integer(\$int64)	帖文 ID
imageUrls	是	string	帖子图片 URL
message	是	string	帖子文本内容
postTypeId	是	integer(\$int64)	帖子类型 ID, 外键

publisherId	是	integer(\$int64)	发布者 ID, 外键
status	是	integer(\$int32)	帖子状态: 0(正常状态), 1(举报过多被挂起), 2(已被删除)

返回示例

```
{
  "code": 0, //返回服务器结果代码
  "message": "", //服务器处理结果信息
  "data": 0 //0 失败, 1 成功
}
```

备注

更多返回错误代码请看首页的错误代码描述

6.21 查询个人评论

简要描述

个人评论查询接口

请求 URL

http://xx.com/api/alumnicycle/mine/comment/list

请求方式

GET

数据类型

JSON

参数

无

返回示例

```
{
  "code": 0, //返回服务器结果代码
  "message": "", //服务器处理结果信息
  "data": [
    {
      "deleted": 0,
      "gmtCreate": "2021-04-23T11:15:00.847Z",
      "gmtModified": "2021-04-23T11:15:00.847Z",
      "id": 1,
      "idFrom": 1,
      "message": "string",
      "postId": 1,
      "preId": 1,
      "status": 0
    }
  ]
}
```

```
]
}
```

备注

更多返回错误代码请看首页的错误代码描述

6.22 删除个人评论

简要描述

个人评论删除接口

请求 URL

http://xx.com/api/alumnicycle/mine/comment/deleted

请求方式

POST

数据类型

JSON

参数: PostComment 对象

参数名	必选	类型	说明
deleted	是	integer(\$int32)	逻辑删除
gmtCreate	是	string(\$date-time)	创建时间
gmtModified	是	string(\$date-time)	最近一次修改时间
id	是	integer(\$int64)	ID
idFrom	是	integer(\$int64)	发布评论用户 ID, 外键
message	是	string	
postId	是	integer(\$int64)	所属帖子 ID, 外键
preId	是	integer(\$int64)	所属评论 ID, 外键
status	是	integer(\$int32)	评论状态: 0(正常状态), 1(举报过多被挂起), 2(已被删除)

返回示例

```
{
  "code": 0, //返回服务器结果代码
  "message": "", //服务器处理结果信息
  "data": 0 //0 失败, 1 成功
}
```

备注

更多返回错误代码请看首页的错误代码描述

6.23 查询黑名单

简要描述

黑名单查询接口

请求 URL

http://xx.com/api/alumnicycle/mine/black/list

请求方式

GET

数据类型

JSON

参数

无

返回示例

```
{
  "code": 0, //返回服务器结果代码
  "message": "", //服务器处理结果信息
  "data": [
    {
      "beUserId": 0,
      "deleted": 0,
      "gmtCreate": "2021-04-23T11:22:12.067Z",
      "gmtModified": "2021-04-23T11:22:12.067Z",
      "id": 0,
      "userId": 0
    }
  ]
}
```

备注

更多返回错误代码请看首页的错误代码描述

6.24 删除黑名单

简要描述

黑名单删除接口

请求 URL

http://xx.com/api/alumnicycle/mine/black/deleted

请求方式

POST

数据类型

JSON

参数: BlackList 对象

参数名	必选	类型	说明
beUserId	是	integer(\$int64)	被拉黑用户 ID
deleted	是	integer(\$int32)	逻辑删除

gmtCreate	是	string(\$date-time)	创建时间
gmtModified	是	string(\$date-time)	修改时间
id	是	integer(\$int64)	黑名单 ID
userId	是	integer(\$int64)	用户 ID

返回示例

```
{
  "code": 0, //返回服务器结果代码
  "message": "", //服务器处理结果信息
  "data": 0 //0 失败, 1 成功
}
```

备注

更多返回错误代码请看首页的错误代码描述

6.25 查询树洞内容

简要描述

树洞内容查询接口

请求 URL

http://xx.com/api/alumnicycle/mine/treehole/content

请求方式

GET

数据类型

JSON

参数

无

返回示例

```
{
  "code": 0, //返回服务器结果代码
  "message": "", //服务器处理结果信息 (数据库无此条记录, 会新增一条新的记录)
  "data": {
    "deleted": 0,
    "fromId": 0,
    "gmtCreate": "2021-04-23T11:32:52.222Z",
    "gmtModified": "2021-04-23T11:32:52.222Z",
    "id": 0,
    "message": "string"
  }
}
```

备注

更多返回错误代码请看首页的错误代码描述

6.26 更新树洞内容

简要描述

树洞内容更新接口

请求 URL

http://xx.com/api/alumnicycle/mine/treehole/update

请求方式

POST

数据类型

JSON

参数: TreeHole 对象

参数名	必选	类型	说明
deleted	是	integer(\$int32) (query)	逻辑删除
fromId	是	integer(\$int64)	用户 ID
gmtCreate	是	string(\$date-time)	创建时间
gmtModified	是	string(\$date-time)	修改时间
id	是	integer(\$int64)	树洞 ID
message	是	string	树洞内容

返回示例

```
{
  "code": 0, //返回服务器结果代码
  "message": "", //服务器处理结果信息
  "data": 0 //0 失败, 1 成功
}
```

备注

更多返回错误代码请看首页的错误代码描述

7. 安全设计

7.1 总览

不同需求对安全产生不同的定义，安全对我们来说就是数据安全，即数据不被窃取，数据不被篡改，数据不被伪造。安全设计的核心是信任域的划分，而基于安全域的原子权限则是数据的坚实保护。原子权限正是这次安全设计的核心思路。

7.2 系统安全分层

以应用结构视角对整个系统的安全进行划分：

1. 系统（平台）层面的安全
2. 应用框架层面的安全
3. 代码实现层面的安全
4. 数据传输层面的安全
5. 客户端层面的安全

7.3 系统（平台）层面的安全

软件运行所需的基础环境既是此处的系统，包括以下内容：操作系统，WEB 服务器中间件，软件运行所需的程序语言环境。

在这个层面主要从这几个方面进行防护：

1. 最小服务，通过减少与外界的接触面来降低入侵发生的可能性，只开放必要的服务端口
2. 安全版本，使用各种应用的最新 stable 版本来防止入侵
3. 合理的 iptables 设置，通过设置严格的出网规则来提高入侵的成本和难度
4. 合理的配置，对于各种应用的配置不应当直接复制网上的 demo，而应该了解每一个配置会产生的行为后进行设计
5. Firewall 的配置，禁止所有非 TCP 通信
6. 安装安全软件识别已有特征的恶意软件和行为
7. 根据入侵常用手段配置蜜罐，使得我们有可能在被入侵的时候得知
8. 数据与程序环境的分离，保证在得到程序环境的时候无法直接得到数据，对各种密钥进行加密
9. 合理的文件系统权限配置，保证得到网站权限无法修改网站代码

7.4 框架层面的安全

框架层面的安全即我们为了快速开发使用的框架

在这个层面从以下方面做好防范：

1. 不要从网上 copy 配置项，应当了解每一个配置会产生的行为，再进行框架的设置
2. 对于自己所使用的框架接口应当清楚会产生怎样的行为，而不是关注会产生怎样的结果
3. 在使用框架开发时应对开发环境和生产环境进行区分，在生产环境去除一切其他不必要的功能，比如报错

7.5 代码层面的安全

在开发时我们应当遵守以下简单的规则进行安全防护：

1. 简单了解 owasptop10 后进行开发
2. 使用框架提供的或团队提供的安全操作函数
3. 在代码开发时应遵守 OWASP 安全编码指南
4. 对所有 web 接口进行
5. 在代码上线前使用白盒安全扫描工具进行测试，比如 codeql
6. 一切用户资源文件都上传至 OSS 中

7.6 传输层面的安全

在数据传输我们使用以下方法进行防御：

1. 部署 CDN 防止发现真实 IP
2. 负载均衡增加入侵难度
3. 部署云 WAF 防御恶意攻击
4. 配置访问频率防止 DDOS 和拆解攻击
5. 使用 HTTPS 加密传输数据
6. 前端对敏感数据比如密码使用 ECDH 密钥交换+AES 加密

7.7 客户端层面的安全

在客户端层面我们要做好以下防护：

1. Csrftoken 防止跨站请求伪造
2. 验证码防止暴力拆解，资源滥用
3. CSP 防止恶意 js 脚本
4. Httponly 防止 cookie 窃取

7.8 安全设计结构图

处于成本考虑，以 docker 为单元进行服务隔离。

