袁阳阳

Nanjing University, Gulou Campus

摘要

RunStyle健康社交管理系统详细设计文档

Runstyle详细设计文档

更新历史

|  |  |  |  |
| --- | --- | --- | --- |
| 修改人员 | 日期 | 变更原因 | 版本号 |
| 袁阳阳 | 2016-11-01 | 初始文档 | V1.0 |
| 袁阳阳 | 2016-11-02 | 添加模块 | V1.0 |
| 袁阳阳 | 2016-11-04 | 完成 | V1.0 |

目录

[1.引言 2](#_Toc466051906)

[1.1编制目的 2](#_Toc466051907)

[1.2参考资料 2](#_Toc466051908)

[2.总体设计 2](#_Toc466051909)

[2.1开发环境 2](#_Toc466051910)

[2.2模块结构 3](#_Toc466051911)

[2.3系统设计模式 3](#_Toc466051912)

[2.4软件架构 3](#_Toc466051913)

[3.数据库设计 4](#_Toc466051914)

[4.导航设计 4](#_Toc466051915)

[4.1总体导航 4](#_Toc466051916)

[4.2运动页面导航 5](#_Toc466051917)

[4.3社交页面导航 5](#_Toc466051918)

[4.4活动页面导航 6](#_Toc466051919)

[5.详细设计 6](#_Toc466051920)

[5.1登录/注册 6](#_Toc466051921)

[5.1.1提供的接口 6](#_Toc466051922)

[5.1.2需要的接口 7](#_Toc466051923)

[5.2运动分析 7](#_Toc466051924)

[5.2.1提供的接口 7](#_Toc466051925)

[5.2.2需要的接口 8](#_Toc466051926)

[5.3睡眠分析 8](#_Toc466051927)

[5.3.1提供的接口 8](#_Toc466051928)

[5.3.2需要的接口 9](#_Toc466051929)

[5.4身体管理 10](#_Toc466051930)

[5.4.1提供的接口 10](#_Toc466051931)

[5.4.2需要的接口 10](#_Toc466051932)

[5.5动态管理 10](#_Toc466051933)

[5.5.1提供的接口 10](#_Toc466051934)

[5.5.2需要的接口 13](#_Toc466051935)

[5.6好友管理 13](#_Toc466051936)

[5.6.1提供的接口 13](#_Toc466051937)

[5.6.2需要的接口 14](#_Toc466051938)

[5.7活动管理 14](#_Toc466051939)

[5.7.1提供的接口 14](#_Toc466051940)

[5.7.2需要的接口 16](#_Toc466051941)

[5.8消息管理 16](#_Toc466051942)

[5.8.1提供的接口 16](#_Toc466051943)

[5.8.2需要的接口 17](#_Toc466051944)

[5.9账户进阶 17](#_Toc466051945)

[5.9.1提供的接口 17](#_Toc466051946)

[5.9.2需要的接口 17](#_Toc466051947)

[6.交互设计 18](#_Toc466051948)

# 1.引言

## 1.1编制目的

本文档是RunStyle运动社交系统的详细设计文档，用于指导后续的软件开发，同时实现与测试人员及用户的沟通。

本文档面向开发人员、测试人员及最终用户编写，是了解系统的导航。

## 1.2参考资料

1）IEEE标准

2）《web程序设计 第8版》

3）《RunStyle运动社交系统需求文档》

# 2.总体设计

## 2.1开发环境

本系统采用php作为主要开发语言，服务端主要使用Apache+php+sqlite, 客户端使用html+css+js。使用Apache作为网络服务器，sqlite作为后台数据管理系统。

开发环境：windows 10

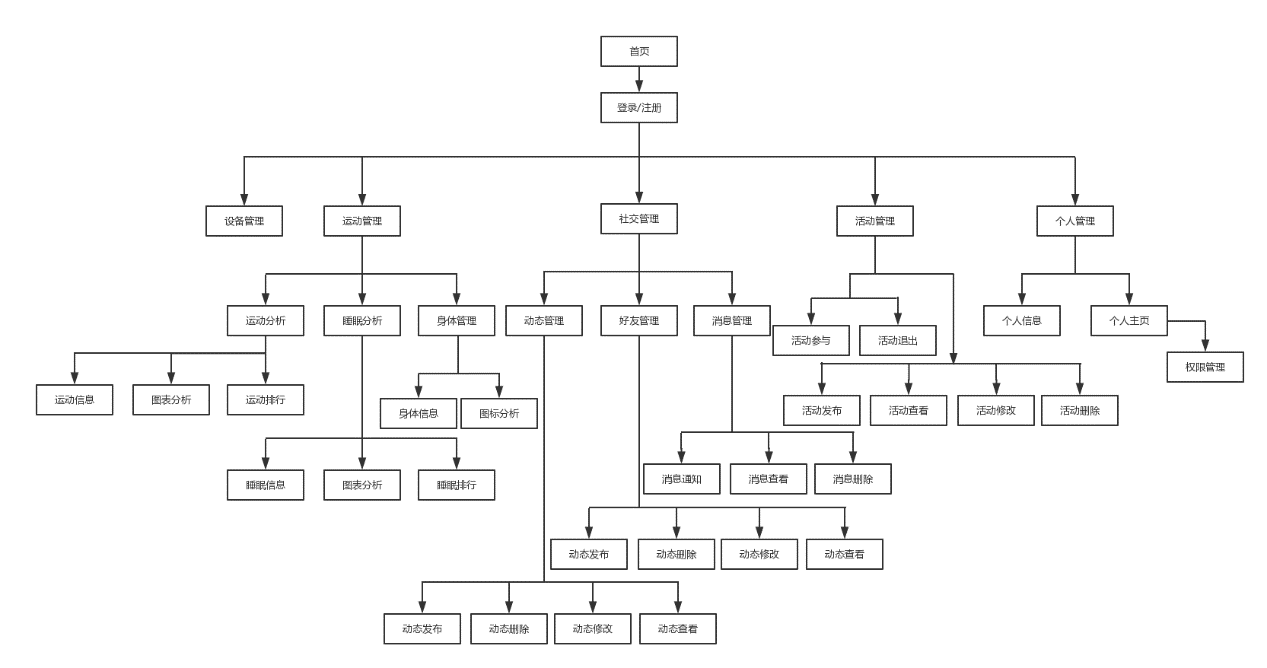
开发工具：phpstorm

测试浏览器：chrome

服务器：aphache2.4

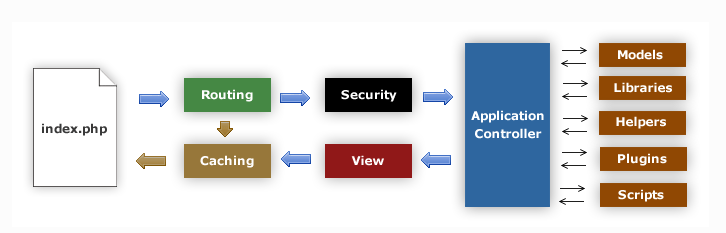
数据库：sqlite3

## 2.2模块结构



## 2.3系统设计模式

本系统采用CodeIgniter框架进行开发，基于MVC设计模式。

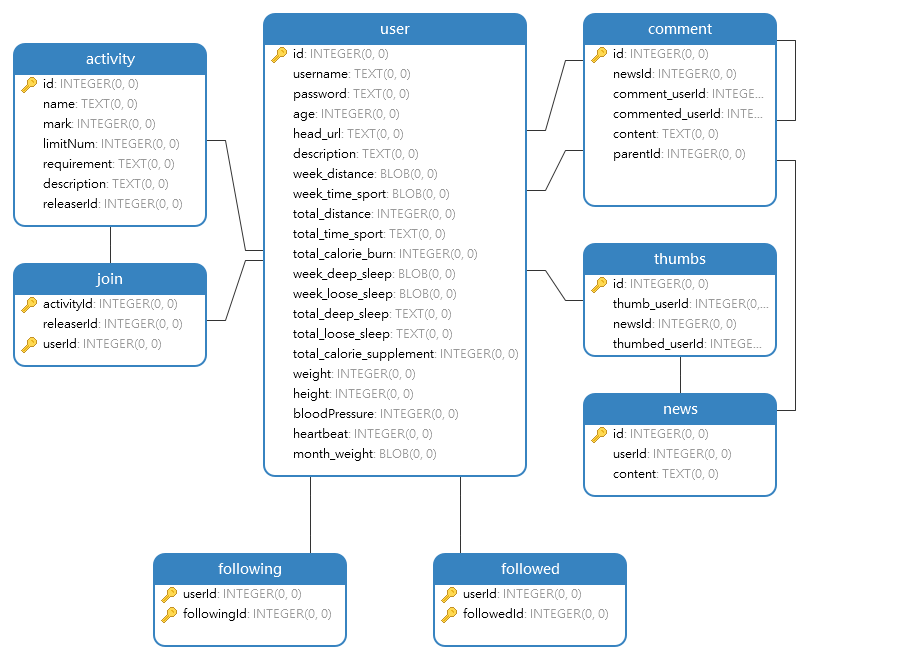


1. index.php 文件作为前端控制器，初始化运行 CodeIgniter 所需的基本资源；
2. Router 检查 HTTP 请求，以确定如何处理该请求；
3. 如果存在缓存文件，将直接输出到浏览器，不用走下面正常的系统流程；
4. 在加载应用程序控制器之前，对 HTTP 请求以及任何用户提交的数据进行安全检查；
5. 控制器加载模型、核心类库、辅助函数以及其他所有处理请求所需的资源；
6. 最后一步，渲染视图并发送至浏览器，如果开启了缓存，视图被会先缓存起来用于 后续的请求。

## 2.4软件架构

Restful架构

# 3.数据库设计



# 4.导航设计

## 4.1总体导航



## 4.2运动页面导航



## 4.3社交页面导航



## 4.4活动页面导航



# 5.详细设计

## 5.1登录/注册

### 5.1.1提供的接口

|  |  |  |
| --- | --- | --- |
| 登录 | url | /user/login |
| 方法类型 | Get |
| 控制器 | UserController |
| 方法 | Login(Request$request) |
| 参数 | Username,password |
| 前置条件 | 用户输入用户名、密码 |
| 后置条件 | 返回登录信息，若失败返回相应错误信息 |
| 注册 | url | /user/signUp |
| 方法类型 | Post |
| 控制器 | UserController |
| 方法 | SignUp(Request$request) |
| 参数 | Username,password |
| 前置条件 | 用户输入注册用户名和密码 |
| 后置条件 | 返回注册结果 |
| 修改账户信息 | url | /user/modify |
| 方法类型 | Post |
| 控制器 | UserController |
| 方法 | Modify(Request$request) |
| 参数 | Username,password,head-url |
| 前置条件 | 用户修改账户信息 |
| 后置条件 | 返回修改结果 |

### 5.1.2需要的接口

|  |  |
| --- | --- |
| 服务名 | 服务 |
| User::find | 查找用户 |
| User::create | 创建用户并加入数据库 |
| User::save | 保存用户信息并更改数据库 |

## 5.2运动分析

### 5.2.1提供的接口

|  |  |  |
| --- | --- | --- |
| 获取运动数据 | url | /sport/basicInfo |
| 方法类型 | Get |
| 控制器 | SportController |
| 方法 | GetBasicInfo(Request$request) |
| 参数 | Username |
| 前置条件 | 用户查看基本运动信息 |
| 后置条件 | 返回用户基本运动信息，若错误返回错误信息 |
| 获取步频分析图 | url | /sport/stepChart |
| 方法类型 | Get |
| 控制器 | SportController |
| 方法 | GetStepChart(Request$request) |
| 参数 | Username |
| 前置条件 | 用户查看步频分析 |
| 后置条件 | 返回用户步频分析信息，若错误返回错误信息 |
| 获取周运动图表 | url | /sport/weekChart |
| 方法类型 | Get |
| 控制器 | SportController |
| 方法 | GetWeekChart(Request$request) |
| 参数 | Username |
| 前置条件 | 用户查看周运动图表 |
| 后置条件 | 返回用户周运动图表数据，若错误返回错误信息 |
| 获取总运动数据 | url | /sport/totalInfo |
| 方法类型 | Get |
| 控制器 | SportController |
| 方法 | GetTotalInfo(Request$request) |
| 参数 | Username |
| 前置条件 | 用户查看总运动信息 |
| 后置条件 | 返回总运动信息，若错误返回错误信息 |
| 获取运动排名 | url | /sport/ranking |
| 方法类型 | Get |
| 控制器 | SportController |
| 方法 | GetRanking(Request$request) |
| 参数 | rankType,rangeType |
| 前置条件 | 用户查看日排名或周排名或总排名 |
| 后置条件 | 返回相应的排名，若错误返回错误信息 |

### 5.2.2需要的接口

|  |  |
| --- | --- |
| 服务名 | 服务 |
| Sport::getBasicInfo | 返回日运动基本信息 |
| Sport::getStepChart | 返回日运动步频信息 |
| Sport::weekInfo | 返回周运动信息 |
| Sport::totalInfo | 返回总运动信息 |
| Sport::ranking | 返回日、周、总运动的排名信息 |

## 5.3睡眠分析

### 5.3.1提供的接口

|  |  |  |
| --- | --- | --- |
| 获取睡眠基本数据 | url | /sleep/basicInfo |
| 方法类型 | Get |
| 控制器 | SleepController |
| 方法 | GetBasicInfo(Request$request) |
| 参数 | Username |
| 前置条件 | 用户查看当日睡眠信息 |
| 后置条件 | 返回用户睡眠数据 |
| 获取睡眠曲线 | url | /sleep/sleepChart |
| 方法类型 | get |
| 控制器 | sleepController |
| 方法 | GetSleepChart(request$request) |
| 参数 | Username |
| 前置条件 | 用户查看睡眠曲线 |
| 后置条件 | 返回用户睡眠曲线数据 |
| 获取周睡眠图表 | url | /sleep/weekSleepChart |
| 方法类型 | Get |
| 控制器 | SleepController |
| 方法 | GetSleepWeekChart(Request$request) |
| 参数 | Username |
| 前置条件 | 用户查看周睡眠图表 |
| 后置条件 | 返回用户周睡眠图表 |
| 获取总的睡眠信息 | url | /sleep/totalInfo |
| 方法类型 | Get |
| 控制器 | SleepController |
| 方法 | GetTotalInfo(Request$request) |
| 参数 | Username |
| 前置条件 | 用户查看睡眠汇总信息 |
| 后置条件 | 返回睡眠汇总信息 |
| 获取睡眠排名 | url | /sleep/ranking |
| 方法类型 | Get |
| 控制器 | SleepController |
| 方法 | GetRanking(Request$request) |
| 参数 | RankType,rangeType |
| 前置条件 | 用户查看排名信息 |
| 后置条件 | 返回相应的排行信息 |

### 5.3.2需要的接口

|  |  |
| --- | --- |
| 服务名 | 服务 |
| Sleep::findBasicInfo | 查找睡眠基本信息 |
| Sleep::getSleepChart | 返回睡眠曲线信息 |
| Sleep::weekInfo | 返回周睡眠信息 |
| Sleep::totalInfo | 返回总睡眠信息 |
| Sleep::ranking | 返回日、周、总睡眠的排名信息 |

## 5.4身体管理

### 5.4.1提供的接口

|  |  |  |
| --- | --- | --- |
| 获取身体基本数据 | url | /body/basicInfo |
| 方法类型 | Get |
| 控制器 | BodyController |
| 方法 | GetBasicInfo(Request$request) |
| 参数 | Username |
| 前置条件 | 用户查看身体信息 |
| 后置条件 | 返回用户基本身体信息 |
| 获取身高体重变化图表 | url | body/weightHeightChart |
| 方法类型 | get |
| 控制器 | BodyController |
| 方法 | GetWeightHeightChart(request$request) |
| 参数 | Username |
| 前置条件 | 用户查看身高体重变化图表 |
| 后置条件 | 返回用户身高体重图表信息 |
| 更改身体基本数据 | url | /body/editInfo |
| 方法类型 | post |
| 控制器 | BodyController |
| 方法 | editBasicInfo(Request$request) |
| 参数 | Username,weight,height，age |
| 前置条件 | 用户修改身体信息 |
| 后置条件 | 返回修改操作结果 |

### 5.4.2需要的接口

|  |  |
| --- | --- |
| 服务名 | 服务 |
| Body::findInfo | 查找用户身体基本信息 |
| Body::save | 更改用户身体信息并保存到数据库 |

## 5.5动态管理

### 5.5.1提供的接口

|  |  |  |
| --- | --- | --- |
| 获取好友动态 | url | /news/friendNews |
| 方法类型 | Get |
| 控制器 | NewsController |
| 方法 | GetFriendNews(Request$request) |
| 参数 | Username,date |
| 前置条件 | 用户查看好友动态 |
| 后置条件 | 返回好友动态信息 |
| 发布动态 | url | /news/releaseNews |
| 方法类型 | post |
| 控制器 | NewsController |
| 方法 | ReleaseNews(Request$request) |
| 参数 | Username,date,content |
| 前置条件 | 用户发布动态 |
| 后置条件 | 数据库保存动态信息，返回发布结果 |
| 编辑动态 | url | /news/editNews |
| 方法类型 | Post |
| 控制器 | NewsController |
| 方法 | EditNews(Request$request) |
| 参数 | Username,date,content,newsId |
| 前置条件 | 用户编辑动态 |
| 后置条件 | 数据库修改动态信息，返回编辑结果 |
| 删除动态 | url | /news/deleteNews |
| 方法类型 | Post |
| 控制器 | NewsController |
| 方法 | DeleteNews(Request$request) |
| 参数 | Username,newsId |
| 前置条件 | 用户删除自己的动态 |
| 后置条件 | 数据库删除该条动态，返回删除结果 |
| 点赞动态 | url | /news/thumbNews |
| 方法类型 | Post |
| 控制器 | newsController |
| 方法 | ThumbNews(Request$request) |
| 参数 | thumb\_user, news\_user, newsId |
| 前置条件 | 用户点赞另一位用户的动态 |
| 后置条件 | 返回点赞结果，通知被点赞的用户 |
| 添加评论 | url | /news/comment |
| 方法类型 | Post |
| 控制器 | NewsController |
| 方法 | Comment(request$request) |
| 参数 | comment\_user, commented\_user, newsId, commented\_id, content |
| 前置条件 | 用户评论另一位用户的动态 |
| 后置条件 | 返回评论结果并通知被评论的用户 |
| 删除评论 | url | /news/deleteComment |
| 方法类型 | Post |
| 控制器 | NewsController |
| 方法 | DeleteComment(request$request) |
| 参数 | NewsId,comment\_id |
| 前置条件 | 用户删除评论 |
| 后置条件 | 在数据库中删除该条评论并返回删除结果 |
| 取消点赞 | url | /news/cancelThumb |
| 方法类型 | Post |
| 控制器 | NewsController |
| 方法 | CancelThumb(request$request) |
| 参数 | Thumb\_user,newsId |
| 前置条件 | 用户取消点赞 |
| 后置条件 | 返回取消结果 |
| 获取点赞数 | url | /news/thumbNum |
| 方法类型 | Get |
| 控制器 | newsController |
| 方法 | GetThumbNum(request$request) |
| 参数 | NewsId, news\_user |
| 前置条件 | 用户查看动态点赞数 |
| 后置条件 | 返回点赞数 |
| 查看个人动态 | url | /news/selfNews |
| 方法类型 | Get |
| 控制器 | NewsController |
| 方法 | GetSelfNews(Request$request) |
| 参数 | Username |
| 前置条件 | 用户查看个人所有状态 |
| 后置条件 | 返回所有状态信息 |
|  | url |  |
| 方法类型 |  |
| 控制器 |  |
| 方法 |  |
| 参数 |  |
| 前置条件 |  |
| 后置条件 |  |

### 5.5.2需要的接口

|  |  |
| --- | --- |
| 服务名 | 服务 |
| News::findNews | 查找好友动态 |
| News::saveNews | 创建动态并存储到数据库 |
| News::editNews | 更改动态信息并存储到数据库 |
| News::deleteNews | 将动态信息从数据库删除 |

## 5.6好友管理

### 5.6.1提供的接口

|  |  |  |
| --- | --- | --- |
| 获取关注的用户列表 | url | /friend/followingList |
| 方法类型 | Get |
| 控制器 | friendController |
| 方法 | GetFollowingList(Request$request) |
| 参数 | Username |
| 前置条件 | 用户查看关注好友列表 |
| 后置条件 | 返回用户关注好友列表 |
| 获取关注我的用户列表 | url | /friend/followedList |
| 方法类型 | Get |
| 控制器 | friendController |
| 方法 | GetFollowedList(Request$request) |
| 参数 | Username |
| 前置条件 | 用户查看关注自己的用户列表 |
| 后置条件 | 返回关注该用户的用户列表 |
| 关注好友 | url | /friend/addFreind |
| 方法类型 | post |
| 控制器 | friendController |
| 方法 | addFriend(Request$request) |
| 参数 | Username,username\_friend |
| 前置条件 | 用户关注某用户 |
| 后置条件 | 更改数据库信息，并返回关注操作是否成功 |
| 取关好友 | url | /friend/deleteFriend |
| 方法类型 | post |
| 控制器 | friendController |
| 方法 | deleteFriend(Request$request) |
| 参数 | Username,username\_friend |
| 前置条件 | 用户取关好友 |
| 后置条件 | 更改数据库信息，返回取关操作是否成功 |
| 查看好友详细信息 | url | /user/detailInfo |
| 方法类型 | get |
| 控制器 | userController |
| 方法 | getDetail(Request$request) |
| 参数 | username\_friend |
| 前置条件 | 用户查看好友详细信息 |
| 后置条件 | 返回详细信息 |

### 5.6.2需要的接口

|  |  |
| --- | --- |
| 服务名 | 服务 |
| friend::findFollowingList | 查找关注的用户列表 |
| Friend::findFollowedList | 查找关注本用户的用户列表 |
| Friend::addFriend | 向关注用户列表中添加新关注的用户，向被关注用户的被关注列表中添加本用户 |
| Friend::deleteFriend | 从关注用户列表中删除该用户 |
| Informing::addInform | 在数据库中增加一条消息 |
| User::getInfo | 查找用户详细信息 |

## 5.7活动管理

### 5.7.1提供的接口

|  |  |  |
| --- | --- | --- |
| 获取全部活动列表 | url | /activity/totalActivity |
| 方法类型 | Get |
| 控制器 | ActivityController |
| 方法 | GetTotalActivity(Request$request) |
| 参数 | pageNum |
| 前置条件 | 用户查看全部活动列表 |
| 后置条件 | 根据页码返回全部活动列表 |
| 发布活动 | url | /activity/releaseActivity |
| 方法类型 | post |
| 控制器 | avtivityController |
| 方法 | releaseActivity(Request$request) |
| 参数 | Username, act\_name, mark, numLimit, requirement, description， startTime, endTime |
| 前置条件 | 用户发布活动 |
| 后置条件 | 向数据库中添加活动信息，返回操作是否成功 |
| 编辑活动 | url | /activity/editActivity |
| 方法类型 | post |
| 控制器 | activityController |
| 方法 | editActivity(Request$request) |
| 参数 | Username, activityId, act\_name, mark, numLimit, requirement, description， startTime, endTime |
| 前置条件 | 用户编辑已发布的活动 |
| 后置条件 | 更改数据库信息，返回操作是否成功 |
| 删除活动 | url | /activity/deleteActivity |
| 方法类型 | post |
| 控制器 | activityController |
| 方法 | deleteActivity(Request$request) |
| 参数 | Username,activityId |
| 前置条件 | 用户删除已发布的活动 |
| 后置条件 | 数据库删除该活动信息，通知已参加该活动的用户，返回操作是否成功 |
| 获取发布的活动列表 | url | /activity/releaseList |
| 方法类型 | Get |
| 控制器 | activityController |
| 方法 | GetReleaseList(Request$request) |
| 参数 | Username，dateType |
| 前置条件 | 用户查看自己发布的活动列表 |
| 后置条件 | 返回符合dateType的用户发布的活动列表 |
| 获取参与的活动列表 | url | /activity/joinList |
| 方法类型 | Get |
| 控制器 | activityController |
| 方法 | GetJoinList(Request$request) |
| 参数 | Username，dateType |
| 前置条件 | 用户查看自己参加的活动列表 |
| 后置条件 | 返回符合dateType的用户参加的活动列表 |
| 搜索活动 | url | /activity/searchList |
| 方法类型 | Get |
| 控制器 | activityController |
| 方法 | searchActivity(Request$request) |
| 参数 | keyword |
| 前置条件 | 用户输入关键词搜索活动 |
| 后置条件 | 返回符合关键字的活动列表 |
| 获取活动详情 | url | /activity/detailActivity |
| 方法类型 | Get |
| 控制器 | activityController |
| 方法 | GetDetailActivity(Request$request) |
| 参数 | activityId |
| 前置条件 | 用户查看自己发布的活动列表 |
| 后置条件 | 返回符合dateType的用户发布的活动列表 |

### 5.7.2需要的接口

|  |  |
| --- | --- |
| 服务名 | 服务 |
| Activity::getTotalActivity | 查找对应页码的全部活动列表 |
| Activity::getSearchedActivity | 查找符合关键词和页码的活动列表 |
| Activity::getReleasedActivity | 返回用户发布的活动列表 |
| Activity::getDetailActivity | 查找活动详细信息 |
| Activity::addActivity | 向数据库中添加活动信息 |
| Activity::editActivity | 在数据库中修改活动信息 |
| Activity::deleteActivity | 在数据库中删除该活动信息 |
| Informing::addInforming | 在数据库中增加一条消息 |

## 5.8消息管理

### 5.8.1提供的接口

|  |  |  |
| --- | --- | --- |
| 获取消息列表 | url | /informing/informingList |
| 方法类型 | Get |
| 控制器 | informingController |
| 方法 | getInformingList(Request$request) |
| 参数 | Username,pageNum |
| 前置条件 | 用户查看消息列表 |
| 后置条件 | 返回符合要求的消息列表 |
| 删除一条消息 | url | /informing/deleteOne |
| 方法类型 | post |
| 控制器 | informingController |
| 方法 | deleteOne(Request$request) |
| 参数 | Username, informingId |
| 前置条件 | 用户删除某条消息 |
| 后置条件 | 从数据库中删除该消息，返回操作是否成功 |
| 清空消息列表 | url | /informing/deleteAll |
| 方法类型 | post |
| 控制器 | informingController |
| 方法 | deleteAll(Request$request) |
| 参数 | Username,informingList |
| 前置条件 | 用户清空消息列表 |
| 后置条件 | 从数据库中删除该用户全部消息，返回操作是否成功 |

### 5.8.2需要的接口

|  |  |
| --- | --- |
| 服务名 | 服务 |
| informing::findInformingList | 查找对应页码的消息列表 |
| informing::deleteInforming | 从数据库中删除指定的消息 |

## 5.9账户进阶

### 5.9.1提供的接口

|  |  |  |
| --- | --- | --- |
| 增加积分 | url | /user/markAdding |
| 方法类型 | post |
| 控制器 | userController |
| 方法 | addMark(Request$request) |
| 参数 | Username,mark |
| 前置条件 | 用户积分上涨 |
| 后置条件 | 在数据库中更改用户信息 |
| 权限增加 | url | /user/level |
| 方法类型 | post |
| 控制器 | userController |
| 方法 | upLevel(Request$request) |
| 参数 | Username, arriveLevel |
| 前置条件 | 用户权限升级 |
| 后置条件 | 在数据中更改用户权限信息 |

### 5.9.2需要的接口

|  |  |
| --- | --- |
| 服务名 | 服务 |
| User::editMark | 在数据库中修改用户积分 |
| User::editLevel | 在数据库中修改用户等级 |

# 6.交互设计

