

系统详细设计文档

Let’s Move !



2016-11-1

NJu-SE

更新历史

|  |  |  |  |
| --- | --- | --- | --- |
| 修改人员 | 日期 | 变更原因 | 版本号 |
| 董本超 | 2016-11-02 | 初始建档 | V1.0 |
|  |  |  |  |

目录

[1 引言 3](#_Toc465879622)

[1.1 编制目的 3](#_Toc465879623)

[1.2 参考资料 3](#_Toc465879624)

[2 总体设计 3](#_Toc465879625)

[2.1 开发环境 3](#_Toc465879626)

[2.2 模块结构 3](#_Toc465879627)

[2.3 系统设计模式 4](#_Toc465879628)

[2.4 软件架构 5](#_Toc465879629)

[3 数据库设计 5](#_Toc465879630)

[4 导航设计 6](#_Toc465879631)

[4.1 导航内容 6](#_Toc465879632)

[4.2 原型界面 6](#_Toc465879633)

[5 接口设计 9](#_Toc465879634)

[5.1 首页 9](#_Toc465879635)

[5.2.1 提供的接口 9](#_Toc465879636)

[5.2.2 需要的接口 9](#_Toc465879637)

[5.2 登录/注册 9](#_Toc465879638)

[5.2.1 提供的接口 9](#_Toc465879639)

[5.2.2 需要的接口 10](#_Toc465879640)

[5.3 身体数据 10](#_Toc465879641)

[5.3.1 提供的接口 10](#_Toc465879642)

[5.3.2 需要的接口 11](#_Toc465879643)

[5.4 健身活动 11](#_Toc465879644)

[5.4.1 提供的接口 11](#_Toc465879645)

[5.4.2 需要的接口 12](#_Toc465879646)

[5.5 粉丝管理 13](#_Toc465879647)

[5.5.1 提供的接口 13](#_Toc465879648)

[5.5.2 需要的接口 13](#_Toc465879649)

[5.6 个人信息管理 14](#_Toc465879650)

[5.6.1 提供的接口 14](#_Toc465879651)

[5.6.2 需要的接口 14](#_Toc465879652)

[6 附录 14](#_Toc465879653)

# 1 引言

## 1.1 编制目的

本文档详细完成对“Let’s Move !”系统的概要设计，达到指导开发的目的，同时实现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. 《web程序设计 第8版》

2. Let’s Move ! 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php＋sqlite+Apache，客户端使用html+css+js。用Apache作为服务器，采用sqlite作为后台数据管理系统。

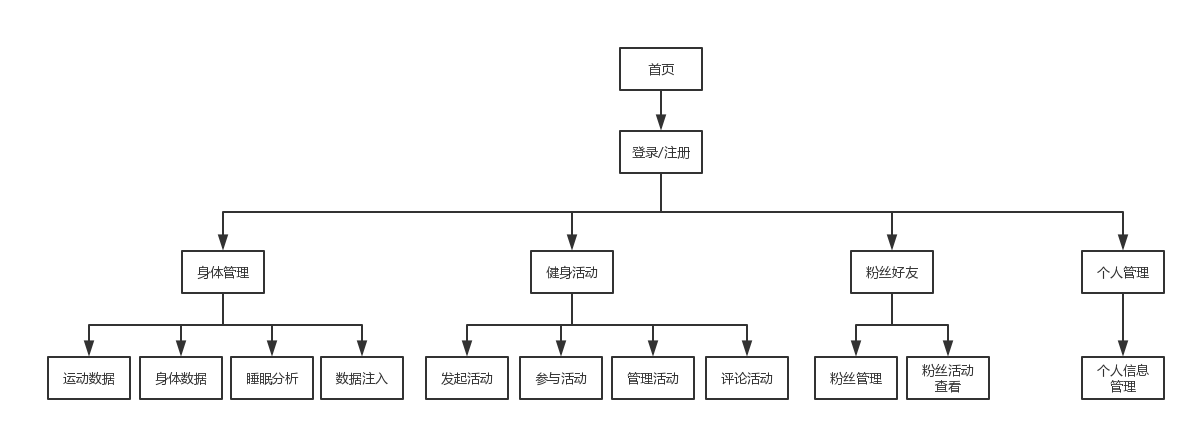
开发环境：Windows 10

开发工具：phpstorm

测试浏览器：Chrome

服务器：Apache2

## 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。



图1 MVC设计模式

CodeIgniter框架在处理请求时采用以下的数据流程处理方式：

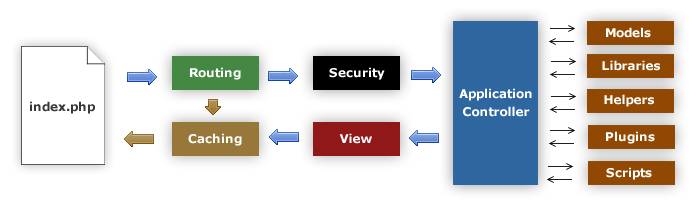


图2 CodeIgniter数据流程图

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

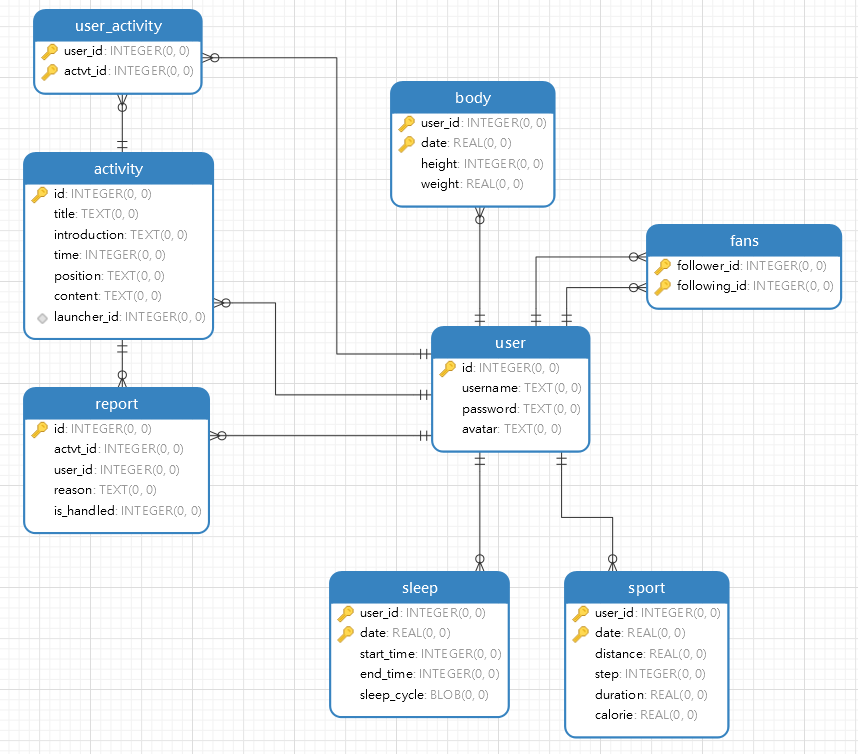


图3 数据库模式

# 4 导航设计

## 4.1 导航内容

|  |  |
| --- | --- |
| 导航栏 | 侧边栏 |
| 首页 | 无 |
| 身体数据 | 我的运动，身体管理，睡眠分析，数据注入 |
| 健身活动 | 所有活动，我发起的，我参与的 |
| 粉丝好友 | 关注我的，我关注的 |

## 4.2 原型界面

UI界面原型如图4-图7所示。

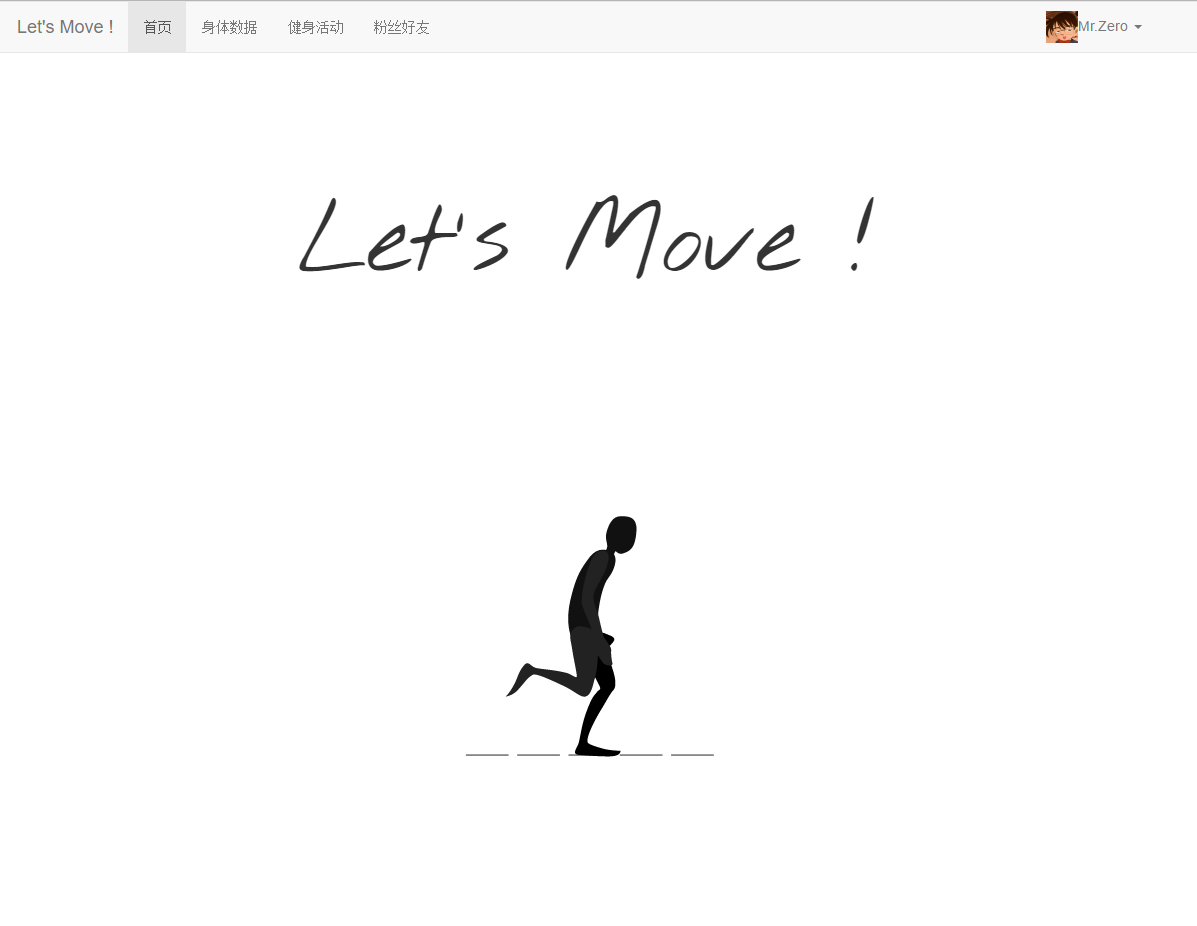


图4 首页

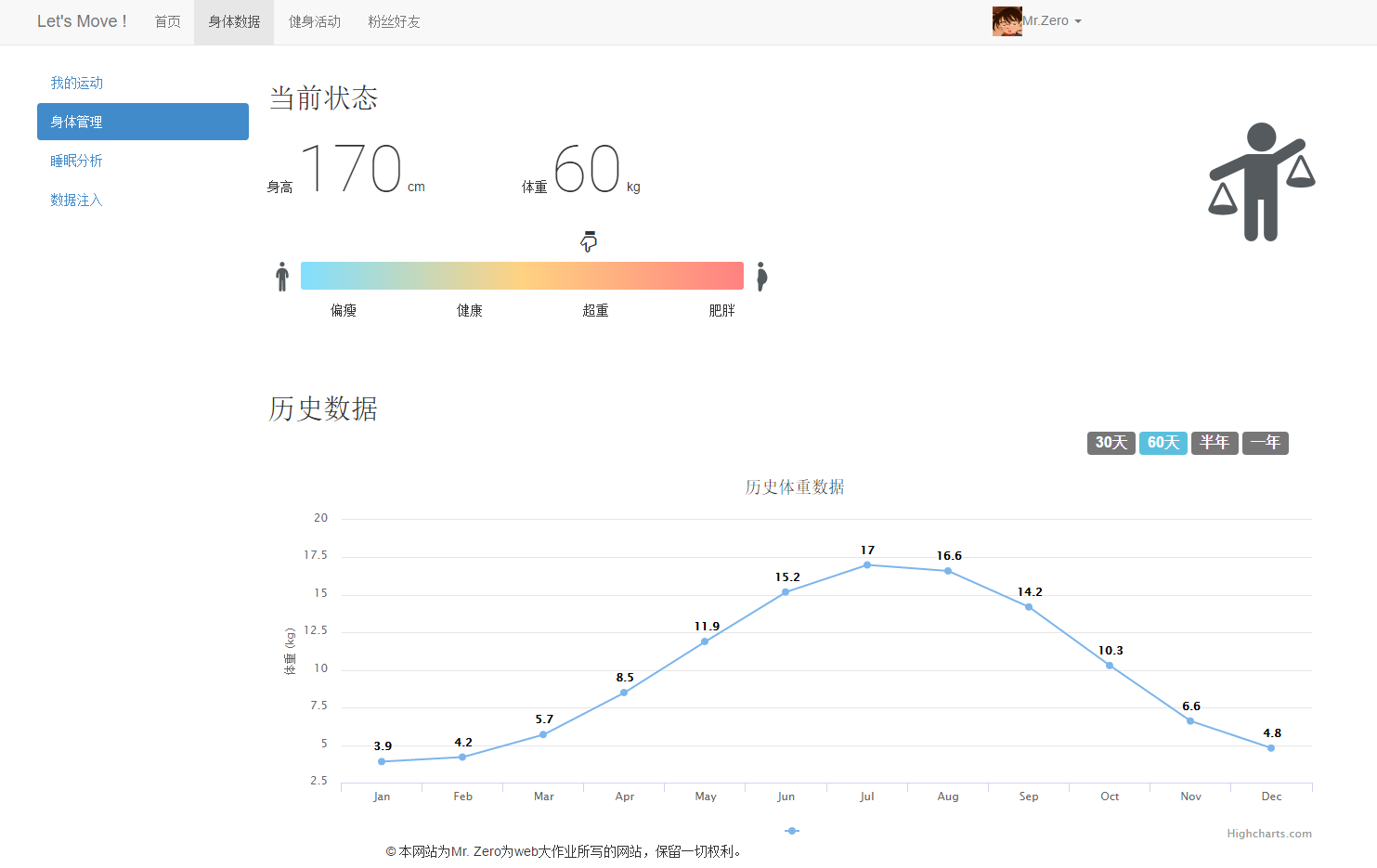


图5 身体数据



图6 健身活动

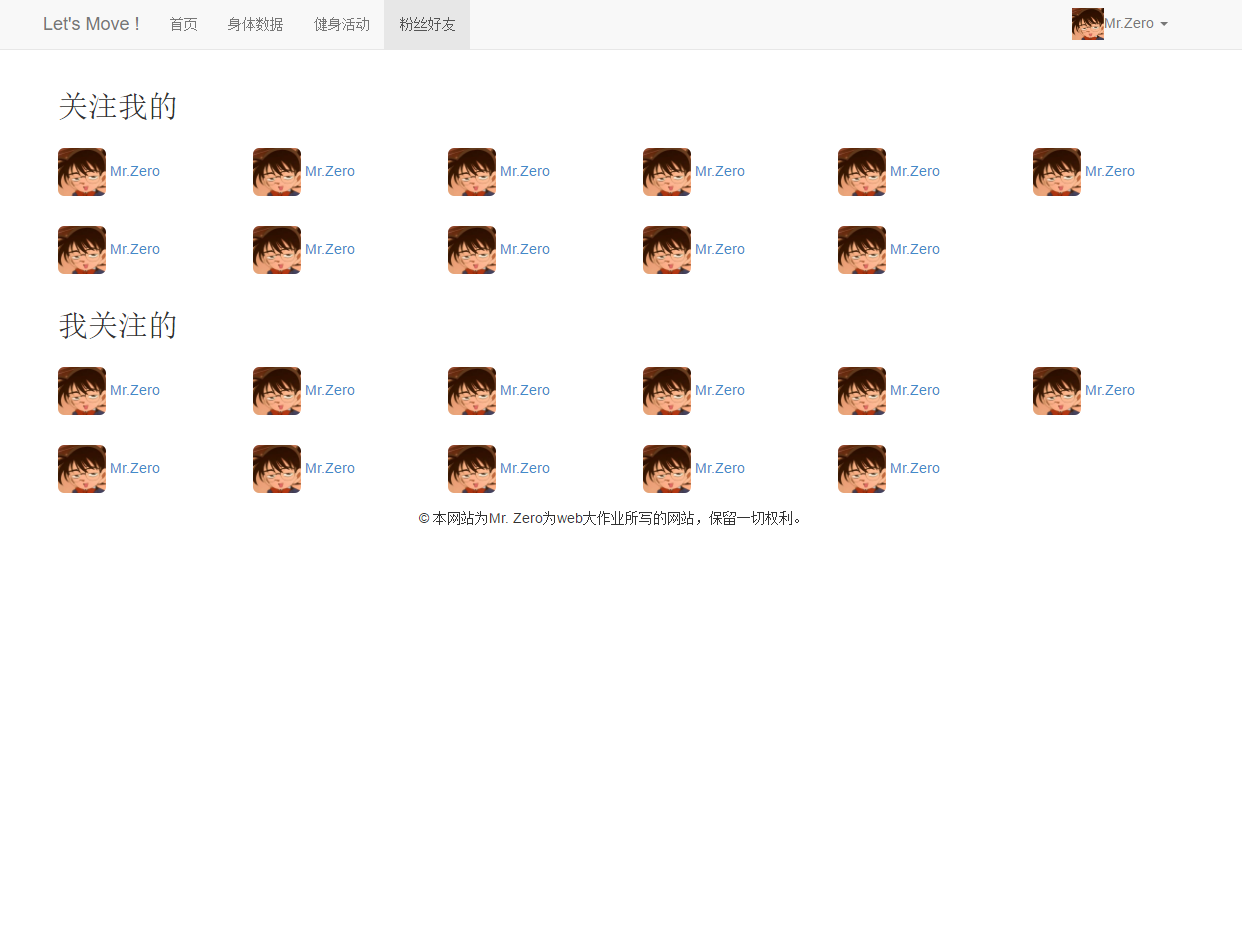


图7 粉丝好友

# 5 接口设计

## 5.1 首页

### 5.2.1 提供的接口

|  |  |  |
| --- | --- | --- |
| 首页 | url | /home |
| 方法类型 | get |
| 控制器 | HomeController |
| 方法 | getHome() |
| 参数 | 无 |
| 前置条件 | 无 |
| 后置条件 | 返回主页界面 |

### 5.2.2 需要的接口

|  |  |
| --- | --- |
| 服务名 | 服务 |
| 无 | 无 |

## 5.2 登录/注册

### 5.2.1 提供的接口

|  |  |  |
| --- | --- | --- |
| 登录界面 | url | /login |
| 方法类型 | get |
| 控制器 | AuthController |
| 方法 | getLogin() |
| 参数 | 无 |
| 前置条件 | 用户未登录 |
| 后置条件 | 返回登录界面 |
| 登录请求 | url | /login |
| 方法类型 | post |
| 控制器 | AuthController |
| 方法 | postLogin($username, $password) |
| 参数 | 用户名，密码（密文） |
| 前置条件 | 用户未登录 |
| 后置条件 | 返回登录结果 |
| 注册界面 | url | /register |
| 方法类型 | get |
| 控制器 | AuthController |
| 方法 | getRegister() |
| 参数 | 无 |
| 前置条件 | 用户未登录 |
| 后置条件 | 返回注册界面 |
| 注册请求 | url | /register |
| 方法类型 | post |
| 控制器 | AuthController |
| 方法 | postRegister($username, $password) |
| 参数 | 用户名，密码（密文） |
| 前置条件 | 用户未登录 |
| 后置条件 | 返回注册结果 |

### 5.2.2 需要的接口

|  |  |
| --- | --- |
| 服务名 | 服务 |
| UserModel::create | 创建新用户 |
| UserModel::login | 查找用户并核对密码 |

## 5.3 身体数据

### 5.3.1 提供的接口

|  |  |  |
| --- | --- | --- |
| 运动数据界面 | url | /analyze/sport |
| 方法类型 | get |
| 控制器 | AnalyzeController |
| 方法 | getSport() |
| 参数 | 无 |
| 前置条件 | 用户已登录 |
| 后置条件 | 返回我的运动界面 |
| 身体管理界面 | url | /analyze/body |
| 方法类型 | get |
| 控制器 | AnalyzeController |
| 方法 | getBody($days) |
| 参数 | 历史体重数据天数 |
| 前置条件 | 用户已登录 |
| 后置条件 | 返回身体管理界面 |
| 睡眠分析界面 | url | /analyze/sleep |
| 方法类型 | get |
| 控制器 | AnalyzeController |
| 方法 | getSleep($date) |
| 参数 | 需要获取睡眠数据的日期 |
| 前置条件 | 用户已登录 |
| 后置条件 | 返回睡眠分析界面 |
| 数据注入界面 | url | /analyze/inject |
| 方法类型 | get |
| 控制器 | AnalyzeController |
| 方法 | getInject() |
| 参数 | 无 |
| 前置条件 | 用户已登录 |
| 后置条件 | 返回数据注入界面 |
| 数据注入界面 | url | /analyze/inject |
| 方法类型 | post |
| 控制器 | AnalyzeController |
| 方法 | postInject($data) |
| 参数 | 需要注入的数据 |
| 前置条件 | 用户已登录 |
| 后置条件 | 返回数据注入结果 |

### 5.3.2 需要的接口

|  |  |
| --- | --- |
| 服务名 | 服务 |
| AnalyzeModel::getSport | 获取运动数据 |
| AnalyzeModel::getBody | 获取身高体重数据 |
| AnalyzeModel::getSleep | 获取睡眠数据 |
| AnalyzeModel::injectData | 注入数据 |

## 5.4 健身活动

### 5.4.1 提供的接口

|  |  |  |
| --- | --- | --- |
| 所有活动列表 | url | /activity/all |
| 方法类型 | get |
| 控制器 | ActivityController |
| 方法 | getAllActivity($page) |
| 参数 | 页数 |
| 前置条件 | 用户已登录 |
| 后置条件 | 返回所有活动列表 |
| 我参与的活动列表 | url | /activity/join |
| 方法类型 | get |
| 控制器 | ActivityController |
| 方法 | getJoinActivity($page) |
| 参数 | 页数 |
| 前置条件 | 用户已登录 |
| 后置条件 | 返回我参与的活动列表 |
| 我发起的活动列表 | url | /activity/launch |
| 方法类型 | get |
| 控制器 | ActivityController |
| 方法 | getLaunchActivity($page) |
| 参数 | 页数 |
| 前置条件 | 用户已登录 |
| 后置条件 | 返回我发起的活动列表 |
| 活动详细信息 | url | /activity/{id} |
| 方法类型 | get |
| 控制器 | ActivityController |
| 方法 | getActivityDetail() |
| 参数 | 无 |
| 前置条件 | 用户已登录 |
| 后置条件 | 返回活动详细信息 |
| 发起新的活动 | url | /activity/launch |
| 方法类型 | post |
| 控制器 | ActivityController |
| 方法 | launchActivity($info) |
| 参数 | 活动信息 |
| 前置条件 | 用户已登录 |
| 后置条件 | 返回发起结果（成功或失败） |
| 加入活动 | url | /activity/join |
| 方法类型 | post |
| 控制器 | ActivityController |
| 方法 | joinActivity($id) |
| 参数 | 活动id |
| 前置条件 | 用户已登录 |
| 后置条件 | 返回加入结果 |
| 举报 | url | /activity/report |
| 方法类型 | post |
| 控制器 | ActivityController |
| 方法 | reportActivity($msg) |
| 参数 | 举报理由 |
| 前置条件 | 用户已登录 |
| 后置条件 | 返回举报结果 |

### 5.4.2 需要的接口

|  |  |
| --- | --- |
| 服务名 | 服务 |
| ActivityModel::getAllActivity | 获取所有活动 |
| ActivityModel::getJoinActivity | 获取参加的活动 |
| ActivityModel::getLaunchActivity | 获取发起的活动 |
| ActivityModel::getActivityDetail | 获取活动详细信息 |
| ActivityModel::launchActivity | 发布新的活动 |
| ActivityModel::joinActivity | 加入活动 |
| ActivityModel::reportActivity | 举报活动 |

## 5.5 粉丝管理

### 5.5.1 提供的接口

|  |  |  |
| --- | --- | --- |
| 关注我的人列表 | url | /fans/follower |
| 方法类型 | get |
| 控制器 | FansController |
| 方法 | getFollower($page) |
| 参数 | 页数 |
| 前置条件 | 用户已登录 |
| 后置条件 | 返回关注我的人列表 |
| 我关注的人列表 | url | /fans/following |
| 方法类型 | get |
| 控制器 | FansController |
| 方法 | getFollowing($page) |
| 参数 | 页数 |
| 前置条件 | 用户已登录 |
| 后置条件 | 返回我关注的人列表 |
| 粉丝参与的活动 | url | /fans/activities |
| 方法类型 | post |
| 控制器 | FansController |
| 方法 | getFansActivities($userid, $page) |
| 参数 | 粉丝id，页数 |
| 前置条件 | 用户已登录 |
| 后置条件 | 返回粉丝参加的活动列表 |

### 5.5.2 需要的接口

|  |  |
| --- | --- |
| 服务名 | 服务 |
| FansModel::getFollower | 获取关注我的人 |
| FansModel::getFollowing | 获取我关注的人 |
| ActivityModel::getJoinActivity | 获取粉丝参加的活动 |

## 5.6 个人信息管理

### 5.6.1 提供的接口

|  |  |  |
| --- | --- | --- |
| 修改密码 | url | /user/updatepassword |
| 方法类型 | post |
| 控制器 | UserController |
| 方法 | updatePassword($password) |
| 参数 | 密码（密文） |
| 前置条件 | 用户已登录 |
| 后置条件 | 返回修改结果 |
| 修改头像 | url | /user/updateavatar |
| 方法类型 | post |
| 控制器 | UserController |
| 方法 | updateAvatar($avatar) |
| 参数 | 头像 |
| 前置条件 | 用户已登录 |
| 后置条件 | 返回修改结果 |

### 5.6.2 需要的接口

|  |  |
| --- | --- |
| 服务名 | 服务 |
| UserModel::updatePassword | 更新用户密码 |
| UserModel::updateAvatar | 更新用户头像 |

# 6 附录