**目录**

[READ ME 13](#_Toc521342355)

[路径规则: 13](#_Toc521342356)

[**示例数据都为PHP格式, 不是JS** 13](#_Toc521342357)

[关于前端获取PHP变量: 13](#_Toc521342358)

[前端检测用户是否登陆|注册? 14](#_Toc521342359)

[返回数据格式 14](#_Toc521342360)

[社群Group 14](#_Toc521342361)

[获取社群列表 14](#_Toc521342362)

[获取社群列表 无page 14](#_Toc521342363)

[获取社群列表带page 15](#_Toc521342364)

[获取社群用户带page 15](#_Toc521342365)

[编辑社群 15](#_Toc521342366)

[操作社群 15](#_Toc521342367)

[创建社群 15](#_Toc521342368)

[加入社群 16](#_Toc521342369)

[退出社群/将他人踢出社群 16](#_Toc521342370)

[将多人批量踢出社群 16](#_Toc521342371)

[获取社群详情 16](#_Toc521342372)

[获取member没参加的社群/其他社群list 17](#_Toc521342373)

[获取带奖金池pool的groupList和会员在该pool下的对应日期打卡次数,不分页 17](#_Toc521342374)

[社群申请group\_apply 18](#_Toc521342375)

[获取申请列表page 18](#_Toc521342376)

[不带page 18](#_Toc521342377)

[[邀请加入|申请加入,写入一条]|[同意加入|拒绝加入] 18](#_Toc521342378)

[奖金池/擂台Pool 18](#_Toc521342379)

[获取擂台列表 18](#_Toc521342380)

[获取擂台列表 无page 19](#_Toc521342381)

[获取擂台列表带page 19](#_Toc521342382)

[获取擂台用户带page 19](#_Toc521342383)

[编辑擂台 19](#_Toc521342384)

[创建擂台 20](#_Toc521342385)

[打卡Punch 20](#_Toc521342386)

[获取打卡列表 20](#_Toc521342387)

[获取打卡列表 无page 20](#_Toc521342388)

[获取打卡列表带page 20](#_Toc521342389)

[获取社群的打卡记录带page 21](#_Toc521342390)

[创建打卡 21](#_Toc521342391)

[打卡评论 22](#_Toc521342392)

[获取评论列表 22](#_Toc521342393)

[获取评论列表byPage 22](#_Toc521342394)

[点赞 22](#_Toc521342395)

[评论 22](#_Toc521342396)

[打赏 23](#_Toc521342397)

[运动计划sport\_plan 23](#_Toc521342398)

[获取运动计划主表列表带page 23](#_Toc521342399)

[获取运动计划主表列表 不带page 23](#_Toc521342400)

[提交计划,写入\修改一条 23](#_Toc521342401)

[添加运动日程(批量) 24](#_Toc521342402)

[获取运动日程(不分页) 24](#_Toc521342403)

[会员荣誉 24](#_Toc521342404)

[创建会员荣誉 24](#_Toc521342405)

[编辑会员荣誉 25](#_Toc521342406)

[会员荣誉列表 25](#_Toc521342407)

[会员荣誉列表（页码） 25](#_Toc521342408)

[获取会员个人荣誉与球队荣誉集合列表 25](#_Toc521342409)

[比赛技术数据 26](#_Toc521342410)

[球员赛季数据 26](#_Toc521342411)

[球队录入比赛球员技术数据 26](#_Toc521342412)

[保存（联赛）单场比赛双方球队的统计数据+球员出席比赛数据 27](#_Toc521342413)

[根据球队ID获取联赛单场比赛球员技术统计 27](#_Toc521342414)

[以球队单位获取单项联赛技术统计排名 28](#_Toc521342415)

[以球员单位获取单项联赛技术统计排名 28](#_Toc521342416)

[以球队单位获取多项联赛技术统计排名 28](#_Toc521342417)

[以球员单位获取多项联赛技术统计排名 29](#_Toc521342418)

[球队荣誉 29](#_Toc521342419)

[创建球队荣誉 29](#_Toc521342420)

[编辑球队荣誉 29](#_Toc521342421)

[球队荣誉列表（页码） 30](#_Toc521342422)

[球队荣誉列表 30](#_Toc521342423)

[球队最近荣誉列表 31](#_Toc521342424)

[删除球队荣誉 31](#_Toc521342425)

[获取球队成员的荣誉列表（页码） 31](#_Toc521342426)

[获取球队成员的荣誉列表 32](#_Toc521342427)

[球队比（约）赛 32](#_Toc521342428)

[创建球队比赛 32](#_Toc521342429)

[编辑球队比赛 33](#_Toc521342430)

[直接录入球队比赛 33](#_Toc521342431)

[保存球队友谊赛信息+比赛战绩数据（不使用了） 33](#_Toc521342432)

[发送比赛认领信息给对方球队 34](#_Toc521342433)

[比赛列表（有页码） 34](#_Toc521342434)

[比赛列表 35](#_Toc521342435)

[比赛列表（所有数据） 35](#_Toc521342436)

[比赛列表包含比赛战绩、出赛球员（有页码） 35](#_Toc521342437)

[球队认领比赛列表（分页） 36](#_Toc521342438)

[球队认领比赛列表 36](#_Toc521342439)

[球队认领比赛数据提交 37](#_Toc521342440)

[比赛管理操作 37](#_Toc521342441)

[球队成员报名参加比赛 37](#_Toc521342442)

[提交约赛应战信息 38](#_Toc521342443)

[回复约赛应战 38](#_Toc521342444)

[球队参加比赛申请列表（页码） 39](#_Toc521342445)

[球队参加比赛申请列表 39](#_Toc521342446)

[球队战绩列表（有页码） 39](#_Toc521342447)

[球队战绩列表 40](#_Toc521342448)

[球队战绩列表（所有数据） 40](#_Toc521342449)

[球队认领完成比赛列表（分页） 41](#_Toc521342450)

[球队认领完成比赛列表 41](#_Toc521342451)

[历史对手球队列表（有页码） 42](#_Toc521342452)

[历史对手球队列表 42](#_Toc521342453)

[历史对手球队列表（所有数据） 43](#_Toc521342454)

[比赛战绩-会员关联列表（页码） 43](#_Toc521342455)

[比赛战绩-会员关联列表 43](#_Toc521342456)

[比赛战绩-会员关联列表（所有数据） 44](#_Toc521342457)

[分页获取lat,lng最近数据 44](#_Toc521342458)

[球员参赛列表（页码） 45](#_Toc521342459)

[球员参赛列表 45](#_Toc521342460)

[已登记技术数据的比赛列表 45](#_Toc521342461)

[已登记技术数据的比赛列表（页码） 46](#_Toc521342462)

[印象标签 46](#_Toc521342463)

[印象标签列表（所有数据无分页） 46](#_Toc521342464)

[发表印象标签评论（一个标签只能评论一次） 47](#_Toc521342465)

[获取印象评论记录列表 47](#_Toc521342466)

[球队 48](#_Toc521342467)

[创建球队 48](#_Toc521342468)

[编辑球队 48](#_Toc521342469)

[球队列表（有页码） 48](#_Toc521342470)

[球队列表（无页码） 49](#_Toc521342471)

[搜索球队列表（有页码） 49](#_Toc521342472)

[搜索球队列表 50](#_Toc521342473)

[我的球队列表 50](#_Toc521342474)

[球队成员列表 50](#_Toc521342475)

[球队成员列表（不分页） 51](#_Toc521342476)

[移除球队成员（只有领队可操作） 51](#_Toc521342477)

[球队成员申请离队（成员自己操作） 51](#_Toc521342478)

[申请加入球队 52](#_Toc521342479)

[回复申请加入球队结果 52](#_Toc521342480)

[添加平台会员（学生）进入球队 52](#_Toc521342481)

[平台会员回复加入球队邀请 53](#_Toc521342482)

[添加非平台会员进入球队 53](#_Toc521342483)

[离队成员回到球队 53](#_Toc521342484)

[创建球队活动 54](#_Toc521342485)

[修改球队活动 54](#_Toc521342486)

[直接创建并录入活动 55](#_Toc521342487)

[球队活动列表（有页码） 55](#_Toc521342488)

[球队活动列表 55](#_Toc521342489)

[球队最新活动记录 56](#_Toc521342490)

[训练营球队最新活动记录 56](#_Toc521342491)

[球队热门活动记录 57](#_Toc521342492)

[报名参加球队活动 57](#_Toc521342493)

[球队活动-会员关联列表（有页码） 58](#_Toc521342494)

[球队活动-会员关联列表 58](#_Toc521342495)

[球队活动-会员关联列表（无分页） 58](#_Toc521342496)

[球队活动、比赛评论列表（有页码） 59](#_Toc521342497)

[球队活动、比赛评论列表 59](#_Toc521342498)

[发布球队活动、比赛评论 60](#_Toc521342499)

[删除球队评论 60](#_Toc521342500)

[球队活动、比赛点赞/取消点赞 61](#_Toc521342501)

[获取球队活动、比赛点赞状态 61](#_Toc521342502)

[最新比赛记录 62](#_Toc521342503)

[训练营球队最新比赛记录 62](#_Toc521342504)

[热门比赛记录 63](#_Toc521342505)

[联赛 League 63](#_Toc521342506)

[注册联赛组织 63](#_Toc521342507)

[编辑联赛组织 63](#_Toc521342508)

[创建联赛 64](#_Toc521342509)

[编辑联赛 64](#_Toc521342510)

[平台展示联赛列表 64](#_Toc521342511)

[平台展示联赛列表（页码） 65](#_Toc521342512)

[联赛球队列表（不分页） 65](#_Toc521342513)

[联赛球队列表 65](#_Toc521342514)

[联赛球队列表（页码） 66](#_Toc521342515)

[联赛球队报名申请列表 66](#_Toc521342516)

[联赛球队列表（页码） 66](#_Toc521342517)

[球队报名申请联赛 67](#_Toc521342518)

[回复联赛球队报名申请 67](#_Toc521342519)

[通知报名联赛球队完善参赛球员名单 67](#_Toc521342520)

[联赛报名球员列表 68](#_Toc521342521)

[联赛报名球员列表（页码） 68](#_Toc521342522)

[创建联赛分组 68](#_Toc521342523)

[编辑联赛分组 69](#_Toc521342524)

[联赛分组列表 69](#_Toc521342525)

[联赛分组列表（页码） 69](#_Toc521342526)

[联赛分组包含组内球队列表 70](#_Toc521342527)

[联赛分组包含组内球队列表（页码） 70](#_Toc521342528)

[保存联赛球队分组数据 70](#_Toc521342529)

[清空联赛分组信息 71](#_Toc521342530)

[获取联赛球队成员列表（无分页） 71](#_Toc521342531)

[获取联赛球队成员列表 71](#_Toc521342532)

[获取联赛球队成员列表（页码） 72](#_Toc521342533)

[球队提交参数成员名单 72](#_Toc521342534)

[设置联赛章程 72](#_Toc521342535)

[获取联赛组织人员列表 73](#_Toc521342536)

[获取联赛组织人员列表（不分页） 73](#_Toc521342537)

[获取联赛组织人员列表（页码） 73](#_Toc521342538)

[检查会员有无联赛组织人员信息 74](#_Toc521342539)

[邀请会员加入联赛组织 74](#_Toc521342540)

[会员回复联赛组织邀请 74](#_Toc521342541)

[删除联赛组织人员 75](#_Toc521342542)

[联赛工作人员列表 75](#_Toc521342543)

[联赛工作人员列表（不分页） 75](#_Toc521342544)

[联赛工作人员列表（页码） 76](#_Toc521342545)

[检查会员有无联赛工作人员信息 76](#_Toc521342546)

[邀请会员联赛工作人员 76](#_Toc521342547)

[会员回复联赛工作人员邀请 76](#_Toc521342548)

[会员申请联赛工作人员 77](#_Toc521342549)

[回复联赛工作人员申请 77](#_Toc521342550)

[删除联赛工作人员 77](#_Toc521342551)

[获取联赛工作人员（裁判）列表 78](#_Toc521342552)

[获取联赛工作人员（裁判）列表 页码 78](#_Toc521342553)

[创建赛程 78](#_Toc521342554)

[编辑赛程 78](#_Toc521342555)

[赛程列表（无分页） 79](#_Toc521342556)

[赛程列表 79](#_Toc521342557)

[赛程列表（页码） 79](#_Toc521342558)

[删除赛程 80](#_Toc521342559)

[清空联赛所有赛程 80](#_Toc521342560)

[获取根据阶段分组的赛程列表 80](#_Toc521342561)

[获取赛程列表（主页展示，可选择查询条件） 80](#_Toc521342562)

[根据联赛分组（小组）生成预览赛程 81](#_Toc521342563)

[批量保存比赛赛程数据 81](#_Toc521342564)

[比赛阶段列表 81](#_Toc521342565)

[比赛阶段列表（页码） 82](#_Toc521342566)

[新建比赛阶段 82](#_Toc521342567)

[修改比赛阶段 82](#_Toc521342568)

[删除比赛阶段 83](#_Toc521342569)

[联赛积分排名 83](#_Toc521342570)

[创建比赛结果数据 83](#_Toc521342571)

[更新比赛结果数据 84](#_Toc521342572)

[发布比赛结果 84](#_Toc521342573)

[比赛结果列表（根据比赛时间、阶段、分组） 84](#_Toc521342574)

[创建联赛文章 85](#_Toc521342575)

[编辑联赛文章 85](#_Toc521342576)

[删除联赛文章 85](#_Toc521342577)

[联赛文章列表 86](#_Toc521342578)

[联赛文章列表（页码） 86](#_Toc521342579)

[联赛模块评论列表（有页码） 86](#_Toc521342580)

[联赛模块评论列表 87](#_Toc521342581)

[发布联赛模块评论 88](#_Toc521342582)

[删除联赛模块评论 88](#_Toc521342583)

[联赛模块点赞/取消点赞 88](#_Toc521342584)

[获取会员在联赛模块点赞信息 89](#_Toc521342585)

[保存比赛阶段晋级球队数据 89](#_Toc521342586)

[比赛阶段晋级球队列表（无分页） 90](#_Toc521342587)

[比赛阶段晋级球队列表 90](#_Toc521342588)

[比赛阶段晋级球队列表（页码） 90](#_Toc521342589)

[裁判员Referee 91](#_Toc521342590)

[获取裁判员列表(有页码) 91](#_Toc521342591)

[获取裁判员列表 91](#_Toc521342592)

[注册裁判员信息 91](#_Toc521342593)

[编辑裁判员信息 92](#_Toc521342594)

[获取会员裁判员数据 92](#_Toc521342595)

[评论裁判 93](#_Toc521342596)

[获取裁判制裁列表（页码） 93](#_Toc521342597)

[获取裁判制裁列表 93](#_Toc521342598)

[邀请裁判加入比赛 93](#_Toc521342599)

[裁判申请|受邀比赛列表（分页） 94](#_Toc521342600)

[裁判申请|受邀比赛列表 94](#_Toc521342601)

[获取裁判申请|邀请比赛详情 94](#_Toc521342602)

[裁判回复比赛邀请 95](#_Toc521342603)

[裁判列表与比赛邀请|申请关系（分页） 95](#_Toc521342604)

[裁判列表与比赛邀请|申请关系 95](#_Toc521342605)

[裁判最近比赛列表（裁判最近订单） 96](#_Toc521342606)

# READ ME

## 路径规则:

#### url(‘地址表达式’,[‘参数’],[‘URL后缀’],[‘域名’])

如

{:url('模块名/方法名',['参数名'=>'参数值'])}

{:url('index/blog/read','id=5&name=thinkphp')}

以上方法会生成下面的URL地址：

/index.php/blog/5/name/thinkphp.html

我们API:

{:url('api/Member/logout',['参数名'=>'参数值'])}

{:url('api/Member/logout')}

lesson->lessonList->lesson

{:url(,[lesson=>'大热'])}

## **示例数据都为PHP格式, 不是JS**

## 关于前端获取PHP变量:

如果要获取一个详情页的班级表的id字段,那么PHP变量为:{$grandeInfo['id']}或者{$grandeInfo.id}

如果要获取一个列表的字段值,则为:{$vol.字段名}或者{$vol['字段名']}

注意:vol只有在当前列表dom有效,有时候vol也为v

## 前端检测用户是否登陆|注册?

js直接读取"{$memberInfo['id']}",如果有值,则已经登录/注册

## 返回数据格式

type:json

{['code'=>200,'msg'=>{data}]}

# 社群Group

## 获取社群列表

api: getGroupListApi

post:

get:page

## 获取社群列表 无page

api: getGroupListNoPageApi

post:

## 获取社群列表带page

api: getGroupListByPageApi

post:

## 获取社群用户带page

api: getGroupMemberListByPageApi

post:

## 编辑社群

api: updateGroupApi

post:

get:group\_id

## 操作社群

api: editGroupApi

post:

get:group\_id

## 创建社群

api: createGroupApi

post:

## 加入社群

api: createGroupMemberApi

get:group\_id

## 退出社群/将他人踢出社群

api: dropGroup

post: group\_id,member\_id,id

注:自己退出可不传member\_id

注: idList的格式为9;

数字为group\_member表数据的id

## 将多人批量踢出社群

api: dropGroups

post: idList,group\_id

注: idList的格式为[1,2,3,4,5,6,7,8,9,10];

数字为group\_member表数据的id

## 获取社群详情

api: getGroupInfoApi

post:

## 获取member没参加的社群/其他社群list

api: getOtherGroupListApi

post:

get:keyword,page

## 获取带奖金池pool的groupList和会员在该pool下的对应日期打卡次数,不分页

api: getGroupListJoinPoolApi

post|get: punch\_time

**注:**

**返回示例**

array(1) {

[0] => array(16) {

["id"] => int(48)

["group"] => string(6) "荣光"

["group\_id"] => int(1)

["member"] => string(7) "wayen\_z"

["avatar"] => string(26) "/static/default/avatar.png"

["member\_id"] => int(6)

["status"] => int(1)

["create\_time"] => int(1527247409)

["update\_time"] => int(1527247409)

["delete\_time"] => NULL

["stake"] => NULL →奖金池需要的打卡金

["pool"] => NULL →奖金池名称

["p\_status"] => NULL →奖金池状态

["p\_id"] => NULL →奖金池id

["times"] => NULL →奖金池日可打卡数

["punchs"] => int(0) →本人在该奖金池下的日打卡数

}

}

# 社群申请group\_apply

## 获取申请列表page

api: getGroupApplyListByPageApi

post:

get:keyword

## 不带page

api:getGroupApplyListNoPageApi

post:

get:keyword

## [邀请加入|申请加入,写入一条]|[同意加入|拒绝加入]

api: createGroupApplyApi

post:

get: group\_apply\_id

注意:如果传group\_apply\_id,则post传status,为同意加入|拒绝加入操作;

status为2或者-1

# 奖金池/擂台Pool

## 获取擂台列表

api: getPoolListApi

get:page

post:

## 获取擂台列表 无page

api: getPoolListNoPageApi

post:

## 获取擂台列表带page

api: getPoolListByPageApi

get:page

post:

## 获取擂台用户带page

api: getPoolMemberListByPageApi

post:

get:page

## 编辑擂台

api: updatePoolApi

post:

get: pool\_id

注:

开始时间starts,格式为2018-5-24

开始时间ends,格式为2018-5-24

必须大于7天

## 创建擂台

api: createPoolApi

post:

注:

开始时间starts,格式为2018-5-24

开始时间ends,格式为2018-5-24

必须大于7天

# 打卡Punch

## 获取打卡列表

api: getPunchListApi

get:page

post:

## 获取打卡列表 无page

api: getPunchListNoPageApi

post:

## 获取打卡列表带page

api: getPunchListByPageApi

get:page

post:

## 获取社群的打卡记录带page

api: getGroupPunchListByPageApi

post:

get:page

## 创建打卡

api:createPunchApi

post:**[ groupList]**

**注:** **groupList为关联的社群数组,格式为:**

[

{student: "荣", student\_id: "51", id: "64", grade: "13213", grade\_id: "26"}

{student: "woo", student\_id: "54", id: "61", grade: "13213", grade\_id: "26"}

];

**内容与group\_punch字段对应;不需要create\_time,udpdate\_time,delete\_time和id**

## 打卡评论

api: commentApi

post:

get:comment\_id(注:如果传comment\_id,则为修改评论)

## 获取评论列表

api:getCommentListApi

post:

## 获取评论列表byPage

api: getCommentListByPageApi

post:

## 点赞

api:likesApi

post|get:punch\_id,punch

## 评论

api: commentApi

post:

**注意: 如果是在punch下面评论别的的评论,需要传to\_member\_id和to\_member和to\_member\_avatar;否则不需要传**

## 打赏

api: rewardApi

post: reward, punch\_id

**注意:reward就是打赏的总金额,相当于stakes;**

# 运动计划sport\_plan

## 获取运动计划主表列表带page

api: getSportPlanListByPageApi

get:keyword

post:

## 获取运动计划主表列表 不带page

**api:** **getSportPlanListNoPageApi**

get:keyword

**post:**

## 提交计划,写入\修改一条

api: createSportPlanApi

post:

get: sport\_plan\_id

**注意: 如果传输sport\_plan\_id,则为更新**

## 添加运动日程(批量)

api: createSportPlanScheduleListApi

post: sportPlanScheduleData

注意: sportPlanScheduleData需要用JSON.stringfy()处理;

## 获取运动日程(不分页)

api: getSportPlanScheduleListApi

post:

# 会员荣誉

## 创建会员荣誉

api:api/member/creatememberhonor

参数：member\_honor表字段

成功 code=>200, msg=>， data=>（member\_honor数据表自增id）

## 编辑会员荣誉

api:api/member/updatememberhonor

参数：member\_honor表字段

成功 code=>200, msg=>

## 会员荣誉列表

api:api/member/getmemberhonorlist

参数：member\_honor表字段，page（第x页）

成功 code=>200, msg=>'', data=>''

## 会员荣誉列表（页码）

api:api/member/getmemberhonorpage

翻页传参：{:url('api/member/getmemberhonorpage')} /?page=页数

参数：member\_honor表字段

成功 code=>200, msg=>'', data=>''

## 获取会员个人荣誉与球队荣誉集合列表

api： api/member/getmemberandteamhonorlist

翻页传参：{:url('api/member/getmebmerandteawmhonorlist')}/?page=页数

参数：member\_id

成功 code=>200, msg=>'',data=>''

# 比赛技术数据

## 球员赛季数据

api: api/matchdata/playerseasonstatis

参数：team\_member\_id（球员id）

数据表：match\_statistics

response:

成功 code=>200, msg=>， data=>[match\_number比赛场次,efficiency效率值,avgdata赛季平均数据]

## 球队录入比赛球员技术数据

api: api/matchdata/savematchstatisticsbyteam

参数：match\_id（比赛id）、match（比赛名称）、match\_record\_id（比赛战绩id）、match\_time（比赛时间，日期格式或时间戳格式）、team\_id（球队id，取当前操作）、team（球队名，同team\_id）；

members（球员技术数据名单集合，json格式）：[{ match\_record\_member\_id, team\_member\_id, name, member\_id, member, number, fg, fga, 3pfg, 3pfga, reb, ast, stl, blk, ft, fta, turnover, foul, lineup}]

response:

成功 code=>200, msg=>

## 保存（联赛）单场比赛双方球队的统计数据+球员出席比赛数据

api: api/ matchdata/savematchstaticsbyleague

参数：同上

response:

成功 code=>200, msg=>

## 根据球队ID获取联赛单场比赛球员技术统计

api： api/ matchdata/ getleaguerecordstaticsbyteam

参数：match \_id（match表id）、match\_record\_id（match\_record表id）、team\_id（team表id）

数据表：match\_statistics表，

response:

成功 code=>200, msg=>， data=> [..... is\_attend: 1|-1] PS.is\_attend为球员出场标识（1出场|-1未出场）

## 以球队单位获取单项联赛技术统计排名

api: api/matchdata/getleaguestaticsrankbyteam

参数： match\_id（联赛id match表id）、field（要获取的技术统计项字段名，参考match\_statistics表字段）、page（页码）

response:

成功 code=>200, msg=>

## 以球员单位获取单项联赛技术统计排名

api: api/matchdata/getleaguestaticsrankbyteammember

参数： match\_id（联赛id match表id）、field（要获取的技术统计项字段名，参考match\_statistics表字段）、page（页码）

response:

成功 code=>200, msg=>

## 以球队单位获取多项联赛技术统计排名

api: api/matchdata/getleaguestaticsallrankbyteam

参数： match\_id（联赛id match表id）

response:

成功 code=>200, msg=>

## 以球员单位获取多项联赛技术统计排名

api: api/matchdata/getleaguestaticsallrankbyteammember

参数： match\_id（联赛id match表id）

response:

成功 code=>200, msg=>

# 球队荣誉

## 创建球队荣誉

api: api/team/createteamhonor

参数：team\_honor数据表字段

response:

成功 code=>200, msg=>， data=>（team\_honor数据表自增id）

## 编辑球队荣誉

api: api/team/editteamhonor

参数：team\_honor数据表字段

response:

成功 code=>200, msg=>，

## 球队荣誉列表（页码）

api: api/team/getteamhonorpage

翻页传参 {:url('api/team/ getteamhonorpage)}/?page=页数

post: team\_honor数据表字段

response:

成功 code=>200, msg=>, data=>（3层数据结构）

## 球队荣誉列表

api: api/team/getteamhonorlist

post：team\_honor数据表字段、page(第x页) 一页10条数据

response:

成功 code=>200, msg=>, data=>（2层数据结构）

## 球队最近荣誉列表

api: api/team/lastteamhonorlist

post：team\_honor数据表字段、page(第x页) 一页10条数据、

response:

成功 code=>200, msg=>, data=>（2层数据结构）

## 删除球队荣誉

api: api/team/ delteamhonor

post：team\_honor数据表id

response:

成功 code=>200, msg=>

## 获取球队成员的荣誉列表（页码）

api: api/team/getteammemberhonorpage

翻页传参 {:url('api/team/ getteammemberhonorpage)}/?page=页数

post: tm\_id（team\_member表id）

response:

成功 code=>200, msg=>, data=>（3层数据结构）

## 获取球队成员的荣誉列表

api: api/team/getteammemberhonorlist

post：team\_honor数据表字段、page(第x页) 一页10条数据、

response:

成功 code=>200, msg=>, data=>（2层数据结构）

# 球队比（约）赛

## 创建球队比赛

api: api/match/createteammatch

post: match数据表字段、record[]：match\_record表字段

type（比赛类型） 1友谊赛 2联赛

友谊赛提交内容：

team\_id （友谊赛主队球队id）

opponent\_id（友谊赛对手球队id）

match\_time（比赛时间）

home\_team\_color、home\_team\_colorstyle （主队球队球衣颜色）

away\_team\_color、away\_team\_colorstyle（对手球队球衣颜色）

response:

成功 code=>200, msg=>'', data=>（新增数据id）

## 编辑球队比赛

api: api/match/updateteammatch

post: match数据表字段、record[]：match\_record表字段

response:

成功 code=>200, msg=>''

## 直接录入球队比赛

api: api/match/ directteammatch

post: match数据表字段、record[]：match\_record表字段

response:

成功 code=>200, msg=>''

## 保存球队友谊赛信息+比赛战绩数据（不使用了）

api: api/match/storefriendlymatchrecord()

post: match表字段（记录比赛主要信息），record[]:match\_record表字段（记录比赛战绩、比赛参赛人员）

response:

成功 code=>200, msg=>'', data=>（新增match表id）

## 发送比赛认领信息给对方球队

api : api/match/sendteamclaimfinishmatch

参数：match\_id(match表id)、id（match\_record表id）

## 比赛列表（有页码）

api: api/match/matchlistpage

翻页传参: {:url('api/match/matchlsitpage')}?page=页数

post: match数据表字段

year（选择年份有传入正确年份才查询创建时间在区间内的数据）；

keyword（搜索关键字，发布比赛的球队名）；

choice\_time（比赛时间选择 1：近7天|2：近15天|3：近1个月）

response:

成功 code=>200, msg=>, data=>（3层数据结构）

## 比赛列表

api: api/match/matchlist

post：match数据表字段、page(第x页) 一页10条数据

year（选择年份有传入正确年份才查询完成时间在区间内的数据）；

keyword（搜索关键字，发布比赛的球队名）；

choice\_time（比赛时间选择 1：近7天|2：近15天|3：近1个月）

response:

成功 code=>200, msg=>'', data=>（2层数据结构）

## 比赛列表（所有数据）

api: api/match/matchlistall

post：match数据表字段

response:

成功 code=>200, msg=>'', data=>（2层数据结构）

## 比赛列表包含比赛战绩、出赛球员（有页码）

api: api/match/matchpagewithrecord

翻页传参: {:url('api/match/ matchpagewithrecord)}?page=页数

post: match数据表字段

keyword（搜索关键字，比赛名称）；

response:

成功 code=>200, msg=>, data=>（3层数据结构）

data里有record是比赛战绩数据，record\_members是比赛球员

## 球队认领比赛列表（分页）

api: api/match/ teamclaimfinishmatchpage  
翻页传参: {:url('api/match/ teamclaimfinishmatchpage)}?page=页数

参数：team\_id（球队id）

response:

成功 code=>200, msg=>, data=>（3层数据结构）

## 球队认领比赛列表

api: api/match/ teamclaimfinishmatchlist

参数：team\_id（球队id）、page（第x页）

response:

成功 code=>200, msg=>, data=>（2层数据结构）

## 球队认领比赛数据提交

api: api/match/ claimfinishmatch

参数：

response:

成功 code=>200, msg=>'',

## 比赛管理操作

api: api/match/removematch

post：match\_id（比赛id），action：editstatus（修改状态status字段）/del（删除比赛信息）

response:

成功 code=>200, msg=>'',

## 球队成员报名参加比赛

api: api/match/joinmatch

post: match\_id（match表id）

response:

成功 code=>200, msg=>'',

## 提交约赛应战信息

api: api/match/joinmatchapply

post:

match\_apply表字段：match\_id、match、team\_id（选择的team\_id）、team（选择的team）、telphone（提交的手机号）、contact（提交的联系人信息）

response:

成功 code=>200, msg=>'', data=>（新增match\_apply表id）

## 回复约赛应战

api:api/match/replymatchapply

post: apply\_id（match\_apply表id）、reply（2 同意|3拒绝）

response:

成功 code=>200, msg=>'', data=>（新增match\_apply表id）

## 球队参加比赛申请列表（页码）

api: api/match/matchapplypage

翻页传参: {:url('api/match/matchapplypage')}?page=页数

post: match\_apply数据表字段

response:

成功 code=>200, msg=>, data=>（3层数据结构）

## 球队参加比赛申请列表

api: api/match/matchapplylist

post：match\_apply数据表字段、page(第x页) 一页10条数据

response:

成功 code=>200, msg=>'', data=>（2层数据结构）

## 球队战绩列表（有页码）

api: api/match/matchrecordlistpage

翻页传参 {:url('api/match /matchrecordlistpage)}/?page=页数

post: team\_id（球队id）、match表字段

year（选择年份有传入正确年份才查询完成时间在区间内的数据）

response:

成功 code=>200, msg=>, data=>（3层数据结构）：match与match\_record表字段同一层结构

## 球队战绩列表

api: api/match/matchrecordlist

post：team\_id（球队id）、match表字段、page(第x页) 一页10条数据

year（选择年份有传入正确年份才查询完成时间在区间内的数据）

response:

成功 code=>200, msg=>'', data=>（2层数据结构）：match与match\_record表字段同一层结构

## 球队战绩列表（所有数据）

api: api/match/matchrecordlistall

post：team\_id（球队id）、match表字段

response:

成功 code=>200, msg=>'', data=>（2层数据结构）：match与match\_record表字段同一层结构

## 球队认领完成比赛列表（分页）

api:api/match/teamclaimfinshmatchpage

翻页传参: {:url('api/match/ teamclaimfinishmatchpage')}?page=页数

post: team\_id（球队id）

response:

成功 code=>200, msg=>, data=>（3层数据结构）

## 球队认领完成比赛列表

api:api/match/teamclaimfinishmatchlist

post: team\_id（球队id）、page(第x页)

response:

成功 code=>200, msg=>, data=>（2层数据结构）

## 历史对手球队列表（有页码）

api: api/match/historyteampage

翻页传参: {:url('api/match/historyteampage ')}?page=页数

post: team\_id（球队id）

keyword（搜索关键字，历史对手球队名）；

response:

成功 code=>200, msg=>, data=>（3层数据结构）

## 历史对手球队列表

api: api/match/historyteamlist

post: team\_id（球队id）、page(第x页) 一页10条数据

keyword（搜索关键字，历史对手球队名）；

response:

成功 code=>200, msg=>'', data=>（2层数据结构）

## 历史对手球队列表（所有数据）

api: api/match/historyteamall

post：team\_id（球队id）

response:

成功 code=>200, msg=>'', data=>（2层数据结构）

## 比赛战绩-会员关联列表（页码）

api: api/match/recordmemberpage

翻页传参: {:url('api/match/recordmemberpage')}?page=页数

post: match\_id （match表id）、status：1报名|2出席比赛|-1无效（剔除）

response:

成功 code=>200, msg=>, data=>（3层数据结构）

## 比赛战绩-会员关联列表

api: api/match/recordmemberlist

post： page(第x页) 一页10条数据、

match\_id （match表id）、status：1报名|2出席比赛|-1无效（剔除）

response:

成功 code=>200, msg=>'', data=>（2层数据结构）

## 比赛战绩-会员关联列表（所有数据）

api: api/match/recordmemberall

post：match\_id （match表id）、status：1报名|2出席比赛|-1无效（剔除）

response:

成功 code=>200, msg=>'', data=>（2层数据结构）

## 分页获取lat,lng最近数据

api: getMatchListOrderByDistanceApi

post|get: lat, lng,page,keyword,choice\_time, year,city,province,area, orderby

PS:返回的distance字段为距离,单位为km'千米';

orderby 默认为距离最近;

choice\_time为1,2,3,对应未来7天,未来15天,未来30天;

## 球员参赛列表（页码）

api: api/match/playermatchpage

翻页传参：{:url('api/match/playermatchpage')}/?page=x

post: team\_member\_id（球员id）

response:

成功 code=>200, msg=>, data=>（3层数据结构），line\_up首发次数，win\_rate球员出场比赛胜率

## 球员参赛列表

api: api/match/playermatchlist

post: team\_member\_id（球员id），page（第x页）

response:

成功 code=>200, msg=>, data=>（2层数据结构），line\_up首发次数，win\_rate球员出场比赛胜率

## 已登记技术数据的比赛列表

api：api/match/matchlisthasstatics

post: team\_id（球队id）、page（第x页）

response:

成功 code=>200, msg=>, data=>（2层数据结构）

## 已登记技术数据的比赛列表（页码）

api：api/match/matchpagehasstatics

post: team\_id（球队id）、page（第x页）

response:

成功 code=>200, msg=>, data=>（3层数据结构）

# 印象标签

## 印象标签列表（所有数据无分页）

api: api/tag/tagall

post:comment\_type（印象评论类型1会员|2球队|3教练|4训练营|5裁判员）

response:

成功 code=>200, msg=>'', data=>（1层数据结构）

## 发表印象标签评论（一个标签只能评论一次）

api: api/tag/addtagcomment

post：

comment\_type 评论类型（1会员|2球队|3教练|4训练营|5裁判员）

commented 被评论的实体名称（球队名之类）

commented\_id 被评论的实体id （球队id之类）

tag\_ids 提交的印象标签id集合，字符串，以逗号拼接（如1,2,3）

tags 提交的印象标签名集合，字符串，以逗号拼接（如球风良好,诚信守时）

response:

成功 code=>200, msg=>''

## 获取印象评论记录列表

api: api/tag/commentlist

post:

comment\_type 评论类型（1会员|2球队）

commented\_id 被评论的实体id （球队id之类）

response:

成功 code=>200,msg=>'', data=>（1层结构），其中data[].count是印象评论次数

# 球队

## 创建球队

api: api/team/createteam

post: 数据表字段对应

response:

成功:code => 200, msg=>'', insid=>id （insid为新增数据id）

## 编辑球队

api: api/team/updateteam

post:数据表字段对应

response:

成功 code=>200, msg => ''

## 球队列表（有页码）

api: api/team/teamlistpage

翻页传参: {:url('api/team/teamlistpage')}/?page=x

post：（查询条件字段）

response:

成功:code=>200, msg=>'', data=>（3层数据结构）

## 球队列表（无页码）

api:api/team/teamlist

post：（查询条件字段）、page(第x页) 一页10条数据

response:

成功 code=>200, msg=>'', data=>（2层数据结构）

## 搜索球队列表（有页码）

api: api/team/searchteamlistpage

翻页传参: {:url('api/team/searchteamlistpage')}/?page=x

post：（查询条件字段）、搜索关键字keyword

response:

成功:code=>200, msg=>'', data=>（3层数据结构）

## 搜索球队列表

api: api/team/searchteamlist

post：（查询条件字段）、搜索关键字keyword 、page(第x页) 一页10条数据

response:

成功 code=>200, msg=>'', data=>（2层数据结构）

## 我的球队列表

api: api/team/myteamlist

post:team\_member表字段，iscamp（是否输出训练营球队 1/0，可不传）

response:

成功code=>200, msg=>'',data=>（2层数据结构）team详情信息在team{}中

## 球队成员列表

api: api/team/teammemberlist

post: team\_id

response:

成功 code=>200, msg=>'', data=>(2层数据结构),team详情信息在team{}中

## 球队成员列表（不分页）

api: api/team/teammembers

post: team\_id

response:

成功 code=>200, msg=>'', data=>(2层数据结构),team详情信息在team{}中

## 移除球队成员（只有领队可操作）

api: api/team/removeteammember

post: team\_id（球队id）、member\_id（会员id）

response:

成功 code=>200, msg=>''

## 球队成员申请离队（成员自己操作）

api: api/team/applyleaveteam

post: team\_id（球队id）、member\_id（会员id）

response:

成功 code=>200, msg=>''

## 申请加入球队

api: api/team/applyjointeam

post: team\_id（球队id）、remarks（申请说明）

response:

成功 code=>200,msg=>'',insid=>xxx

## 回复申请加入球队结果

api: api/team/applyreply

post: apply\_id(apply表id)、reply(2 同意 3拒绝)

response:

成功 code=>200, msg=>'',

## 添加平台会员（学生）进入球队

api: api/team/addmember

post: team\_id、member\_id、student\_id（加学生就必须要）

response:

成功 code=>200, msg=>''

## 平台会员回复加入球队邀请

api: api/team/replyteaminvitation

post: apply\_id(apply表id)、reply(2 同意 3拒绝)

response:

成功 code=>200, msg=>''

## 添加非平台会员进入球队

api: api/team/addnoregmember

post:team\_id、name（姓名）、telephone（手机号）

response:

成功 code=>200, msg=>''

## 离队成员回到球队

api: api/team/returnteammember

post: id(team\_member表id)、team\_id（球队id）

response:

成功 code=>200, msg=>''

## 创建球队活动

api: api/team/createteamevent

post: 页面填写字段对应数据字段名

response:

成功 code=>200, msg=>'', data=>新增id(team\_event表id)

## 修改球队活动

api: api/team/updateteamevent

post: 页面填写字段对应数据字段名、参加活动人员(memberdata 格式形式与训练营活动一样)

response:

成功 code=>200, msg=>'',

## 直接创建并录入活动

api: api/team/directcreateteamevent

post: 页面填写字段对应数据字段名、参加活动人员(memberdata 格式形式与训练营活动一样)

response:

成功 code=>200, msg=>'', 新增id(team\_event表id)

## 球队活动列表（有页码）

api: api/team/teameventlistpage

翻页传参 {:url('api/team/teameventlistpage')}/?page=页数

post: 查询字段条件

year（选择年份有传入正确年份才查询创建时间在区间内的数据）

response:

成功 code=>200, msg=>, data=>（3层数据结构）

## 球队活动列表

api: api/team/teameventlist

post：（查询条件字段）、page(第x页) 一页10条数据

year（选择年份有传入正确年份才查询创建时间在区间内的数据）

response:

成功 code=>200, msg=>'', data=>（2层数据结构）

## 球队最新活动记录

api: api/team/lastevent

post： team\_id（球队id）、status=1（上架活动）、is\_finished=-1（未完成）

response:

成功 code=>200, msg=>'', data=>（一维数组）

其中memberlist是活动人员名单列表，start\_time\_stamp是活动开始时间戳，is\_finished\_num是活动是否完成（-1未完成|1已完成）

## 训练营球队最新活动记录

api: api/team/lastcampteamevent

post： camp\_id（训练营id）、status=1（上架活动）

response:

成功 code=>200, msg=>'', data=>（一维数组）

其中memberlist是活动人员名单列表，start\_time\_stamp是活动开始时间戳，is\_finished\_num是活动是否完成（-1未完成|1已完成）

## 球队热门活动记录

api: api/team/hotevent

post： team\_id（球队id）、status=1（上架活动）、is\_finished=-1（未完成）、hot=1（热门标识）

response:

成功 code=>200, msg=>'', data=>（一维数组）

其中memberlist是活动人员名单列表，start\_time\_stamp是活动开始时间戳，is\_finished\_num是活动是否完成（-1未完成|1已完成）

## 报名参加球队活动

api: api/team/jointeamevent

post: event\_id（球队活动id）

response:

成功 code=>200, msg=>'', data=>新增id(team\_event\_member表id)

## 球队活动-会员关联列表（有页码）

api: api/team/teameventmemberlistpage

翻页传参 {:url('api/team/teameventlistpage')}/?page=页数

post: 查询字段条件

response:

成功 code=>200, msg=>, data=>（3层数据结构）

## 球队活动-会员关联列表

api: api/team/teameventmemberlist

post：（查询条件字段）、page(第x页) 一页10条数据

response:

成功 code=>200, msg=>'', data=>（2层数据结构）

## 球队活动-会员关联列表（无分页）

api: api/team/teameventmemberall

post:（查询条件字段）

response:

成功code=>200, msg=>'', data=>（2层数据结构）

## 球队活动、比赛评论列表（有页码）

api: api/team/teamcommentlistpage

翻页传参 {:url('api/team/teameventlistpage')}/?page=页数

post:

comment\_type 评论类型 1球队 | 2球队活动 | 3球队比赛

comment\_id 被评论实体id 根据评论类型所对应数据表的id字段

response:

成功 code=>200, msg=>, data=>（3层数据结构）, thumbup\_count=>（点赞数），can\_delete=>0/1 （1可删除）

## 球队活动、比赛评论列表

api: api/team/teamcommentlist

post:

comment\_type 评论类型 1球队 | 2球队活动 | 3球队比赛|4球员

comment\_id 被评论实体id 根据评论类型所对应数据表的id字段

page(第x页) 一页10条数据

response:

成功 code=>200, msg=>, data=>（2层数据结构）, thumbup\_count=>（点赞数），can\_delete=>0/1 （1可删除）

## 发布球队活动、比赛评论

api: api/team/addteamcomment

post:

comment\_type 评论类型 1球队 | 2球队活动 | 3球队比赛|4球员

comment\_id 被评论实体id 根据评论类型所对应数据表的id字段

comment 评论文字内容

team\_id 被评论实体所有team\_id（有就传）

response:

成功 code=>200, msg=>'', data=>新增id(team\_comment表id)

## 删除球队评论

api： api/team/delteamcomment

post: id（team\_comment表id）

response:

成功 code=>200, msg=>'',

## 球队活动、比赛点赞/取消点赞

api： api/team/dianzan

post:

comment\_type 评论类型 1球队 | 2球队活动 | 3球队比赛|4球员

comment\_id 被评论实体id 根据评论类型所对应数据表的id字段

response:

成功 code=>200, msg=>'',thumbup\_count=>（最新的点赞数字统计）

## 获取球队活动、比赛点赞状态

api: api/team/isthumbup

post:

comment\_type 评论类型 1球队 | 2球队活动 | 3球队比赛|4球员

comment\_id 被评论实体id 根据评论类型所对应数据表的id字段

response:

成功 code=>200, msg=>'', thumbsup =>1（有点赞）|0（没点赞）

## 最新比赛记录

api: api/team/lastmatch

post： team\_id（球队id）、status=1（上架活动）、is\_finished=-1（未完成）

response:

成功 code=>200, msg=>'', data=>（一维数组）

其中memberlist是比赛人员名单列表，is\_finished\_num是活动是否完成（-1未完成|1已完成）

## 训练营球队最新比赛记录

api: api/team/lastcampteammatch

参数：camp\_id（训练营id）、member\_id（会员id）

esponse:

成功 code=>200, msg=>'', data=>（一维数组）

其中memberlist：比赛人员名单列表，is\_finished\_num：比赛是否完成（-1未完成|1已完成）

## 热门比赛记录

api: api/team/hotmatch

post： team\_id（球队id）、status=1（上架活动）、is\_finished=-1（未完成）、

response:

成功 code=>200, msg=>'', data=>（一维数组）

其中memberlist是比赛人员名单列表，start\_time\_stamp是活动开始时间戳，is\_finished\_num是活动是否完成（-1未完成|1已完成）

# 联赛 League

## 注册联赛组织

api: api/league/creatematchorg

参数：match\_organization数据表字段

response:

成功 code=>200, msg=>， data=>（match\_organization数据表自增id）

## 编辑联赛组织

api: api/league/editmatchorg

参数：match\_organization数据表字段，证件信息

response:

成功 code=>200, msg=>''

## 创建联赛

api: api/league/creatematch

post: match数据表字段

response:

成功 code=>200, msg=>, data=>match表新增id

## 编辑联赛

api：api/league/updatematch

post: match数据表字段

response:

成功 code=>200, msg=>''

## 平台展示联赛列表

api: api/league/platformleaguelist

参数： keyword（关键字，联赛名称）、page(第x页) 一页10条数据

成功 code=>200, msg=>'', data=>（一维数组）

## 平台展示联赛列表（页码）

api: api/league/platformleaguepage

翻页传参 {:url('api/ league/platformleaguepage ')}/?page=页数

参数： keyword（关键字，联赛名称）

成功 code=>200, msg=>'', data=>（二维数组）

## 联赛球队列表（不分页）

api: api/league/ leagueteams

参数： league\_id（联赛id）

成功 code=>200, msg=>'', data=>（一维数组）

## 联赛球队列表

api: api/league/ leagueteamlist

参数： keyword（关键字，球队名称）、league\_id（联赛id）、page(第x页) 一页10条数据

成功 code=>200, msg=>'', data=>（一维数组）

## 联赛球队列表（页码）

api: api/league/ leagueteampage

翻页传参 {:url('api/ league/ leagueteampage ')}/?page=页数

参数：keyword（关键字，球队名称）、league\_id（联赛id）、page(第x页) 一页10条数据

成功 code=>200, msg=>'', data=>（二维数组）

## 联赛球队报名申请列表

api: api/league/ leaguesignupteamlist

参数： league\_id（联赛id）、page(第x页) 一页10条数据

成功 code=>200, msg=>'', data=>（一维数组）

## 联赛球队列表（页码）

api: api/league/ leaguesignupteampage

翻页传参 {:url('api/ league/ leaguesignupteampage')}/?page=页数

参数：league\_id（联赛id）、page(第x页) 一页10条数据

成功 code=>200, msg=>'', data=>（二维数组）

## 球队报名申请联赛

api: api/league/signupleague

参数：league\_id（联赛id）、team\_id（会员所在球队id）

成功 code=>200, msg=>'',

## 回复联赛球队报名申请

api: api/league/replyleaguesignupteam

参数：apply\_id（match\_apply表id）、status（2同意，3拒绝）

成功 code=>200, msg=>'',

## 通知报名联赛球队完善参赛球员名单

api： api/league/noticeteamcompleteplayer

参数: apply\_id（match\_apply表id）

成功code=> 200，msg=>''

## 联赛报名球员列表

api：api/league/leaguesignupteamlist

参数：league\_id（match表id）、page(第x页) 一页10条数据

成功 code=>200, msg=>'', data=>（一维数组）

## 联赛报名球员列表（页码）

api：api/league/leaguesignupteampage

翻页传参 {:url('api/ league/ leaguesignupteampage ')}/?page=页数

参数：league\_id（match表id）、page(第x页) 一页10条数据

成功 code=>200, msg=>'', data=>（二维数组）

## 创建联赛分组

api： api/league/addleaguegroup

参数：match\_id（联赛在match表id）、name（分组名称）

成功 code=>200, msg=>'',

## 编辑联赛分组

api： api/league/updategleaguegroup

参数：id（match\_group表id）、match\_id（联赛在match表id）、name（分组名称）

成功 code=>200, msg=>'',

## 联赛分组列表

api：api/league/leaguegrouplist

参数：league\_id（联赛match表id）、page(第x页) 一页10条数据

成功 code=>200, msg=>'', data=>（一维数组）

## 联赛分组列表（页码）

api: api/league/leaguegrouppage

参数：league\_id（match表id）、page(第x页) 一页10条数据

成功 code=>200, msg=>'', data=>（二维数组）

## 联赛分组包含组内球队列表

api：api/league/leaguegroupwithteamlist

参数：league\_id（联赛 match表id）、page(第x页) 一页10条数据

成功 code=>200，msg=>''， data=>

## 联赛分组包含组内球队列表（页码）

api：api/league/leaguegroupwithteampage

参数：league\_id（联赛 match表id）、page(第x页) 一页10条数据

成功 code=>200，msg=>''， data=>

## 保存联赛球队分组数据

api：api/league/saveleaguegroup

参数：league\_id（联赛match表id）

成功 code=>200，msg=>''， data=>

## 清空联赛分组信息

api: api/league/clearmatchgroup

参数：league\_id（联赛match表id）

成功 code=>200，msg=>''， data=>

## 获取联赛球队成员列表（无分页）

api：api/league/getmatchteammembers

参数：league\_id（联赛match表id）

成功 code=>200，msg=>''， data=>

## 获取联赛球队成员列表

api：api/league/getmatchteamemberlist

参数：league\_id（联赛 match表id）、page(第x页) 一页10条数据

成功 code=>200，msg=>''， data=>

## 获取联赛球队成员列表（页码）

api：api/league/getmatchteamemberpage

参数：league\_id（联赛 match表id）、page(第x页) 一页10条数据

成功 code=>200，msg=>''， data=>

## 球队提交参数成员名单

api：api/league/savematchteammemberfromteam

参数：match\_id（联赛match表id）、TeamMemberData（参赛球员名单json数据）

成功 code=>200，msg=>''， data=>

## 设置联赛章程

api：api/league/setregulation

参数：id（联赛match表id）、setregulation（章程）

成功 code=>200，msg=>''， data=>

## 获取联赛组织人员列表

api：api/league/getmatchorgmemberlist

参数：league\_id（联赛 match表id）、page(第x页) 一页10条数据

成功 code=>200，msg=>''， data=>

## 获取联赛组织人员列表（不分页）

api：api/league/getmatchorgmembers

参数：league\_id（联赛 match表id）

成功 code=>200，msg=>''， data=>

## 获取联赛组织人员列表（页码）

api：api/league/getmatchorgmemberpage

参数：league\_id（联赛 match表id）、page(第x页) 一页10条数据

成功 code=>200，msg=>''， data=>

## 检查会员有无联赛组织人员信息

api：api/league/checkmatchorgmember

参数：match\_org\_id（联赛组织match\_org表id）、member\_id（会员id）

返回：code=>200 有数据，code=>100没数据

## 邀请会员加入联赛组织

api：api/league/invitematchorgmember

参数：match\_org\_id（联赛组织match\_org表id）、member\_id（会员id）

成功 code=>200

## 会员回复联赛组织邀请

api:api/league/replymatchorginvitation

参数：apply\_id（apply申请表id， organization\_type=5 、apply\_type=2），status回复状态（2同意|3拒绝）

成功 code=>200

## 删除联赛组织人员

api：api/league/delmatchorgmember

参数：id（match\_org\_member表id）

成功 code=>200

## 联赛工作人员列表

api：api/league/getmatchmemberlist

参数：league\_id（联赛 match表id）、page(第x页) 一页10条数据

成功 code=>200，msg=>''， data=>

## 联赛工作人员列表（不分页）

api：api/league/getmatchmembers

参数：league\_id（联赛 match表id）

成功 code=>200，msg=>''， data=>

## 联赛工作人员列表（页码）

api：api/league/getmatchmemberpage

参数：league\_id（联赛 match表id）、page(第x页) 一页10条数据

成功 code=>200，msg=>''， data=>

## 检查会员有无联赛工作人员信息

api：api/league/checkmatchmember

参数：match\_id（联赛match表id）、member\_id（会员id）

返回：code=>200 有数据，code=>100没数据

## 邀请会员联赛工作人员

api：api/league/inviematchmember

参数：match\_id（联赛match表id）、member\_id（会员id）、type（工作人员角色）

成功 code=>200

## 会员回复联赛工作人员邀请

api：api/league/replymatchworkerinvitation

参数：apply\_id（apply申请表id， organization\_type=4, apply\_type=2）

成功 code=>200

## 会员申请联赛工作人员

api： api/league/applyleagueworker

参数： match\_id（联赛match表id）、member\_id（会员id）、type（工作人员角色）

成功 code=>200

## 回复联赛工作人员申请

api：api/league/replymatchworkerapply

参数：apply\_id（apply申请表id， organization\_type=5 、apply\_type=2），status回复状态（2同意|3拒绝）

成功 code=>200

## 删除联赛工作人员

api：api/league/delmatchmember

参数：id（match\_ member表id）

成功 code=>200

## 获取联赛工作人员（裁判）列表

api: api/league/getleaguerefereelist

参数：league\_id（联赛 match表id）、page(第x页) 一页10条数据

成功 code=>200，msg=>''， data=>

## 获取联赛工作人员（裁判）列表 页码

api: api/league/getleaguerefereepage

参数：league\_id（联赛 match表id）、page(第x页) 一页10条数据

成功 code=>200，msg=>''， data=>

## 创建赛程

api：api/league/createschedule

参数：match\_schedule表字段

成功 code=>200，msg=>''， data=>

## 编辑赛程

api：api/league/updateschedule

参数：match\_schedule表字段、id（match\_schedule表id）

成功 code=>200，msg=>''， data=>

## 赛程列表（无分页）

api：api/league/getmatchschedules

参数：league\_id（联赛 match表id）

成功 code=>200，msg=>''， data=>

## 赛程列表

api：api/league/getmatchschedulelist

参数：league\_id（联赛 match表id）、page(第x页) 一页10条数据

成功 code=>200，msg=>''， data=>

## 赛程列表（页码）

api：api/league/getmatchschedulepage

参数：league\_id（联赛 match表id）、page(第x页) 一页10条数据

成功 code=>200，msg=>''， data=>

## 删除赛程

api：api/league/delmatchschedule

参数：id（match\_schedule表id）

成功 code=>200，msg=>''，

## 清空联赛所有赛程

api：api/league/clearmatchschedule

参数：match\_id（match表id）

成功 code=>200，msg=>''，

## 获取根据阶段分组的赛程列表

api：api/league/getmatchschedulelistbystageandgroup

参数：league\_id（联赛 match表id）

成功 code=>200，msg=>''，data=>[]

## 获取赛程列表（主页展示，可选择查询条件）

api: api/league/getshowmatchschedulelist

参数：league\_id（联赛match表id）、sort（查询排序：1最近7天|2未开始的赛程|3已完成有比分信息的赛程|0或4全部）

成功 code=>200，msg=>''，data=>[]

## 根据联赛分组（小组）生成预览赛程

api：api/league/buildschedulebygroup

参数：match\_id（联赛match表id），group\_id（联赛分组match\_group表id，不传入当联赛所有分组）

成功 code=>200，msg=>''，， data=>

## 批量保存比赛赛程数据

api：api/league/saveallmatchschedule

参数：match\_id（联赛 match表id）、scheduleList（赛程数据json）

成功 code=>200，msg=>''，

## 比赛阶段列表

api： api/league/getmatchstagelist

参数：league\_id（联赛 match表id）、page(第x页) 一页10条数据

成功 code=>200，msg=>''， data=>

## 比赛阶段列表（页码）

api：api/league/getmatchstagepage

参数：league\_id（联赛 match表id）、page(第x页) 一页10条数据

成功 code=>200，msg=>''， data=>

## 新建比赛阶段

api：api/league/addmatchstage

参数：match\_stage表字段

成功 code=>200，msg=>''

## 修改比赛阶段

api：api/league/updatematchstage

参数：match\_stage表字段，需要传入id（match\_stage表id）

成功 code=>200，msg=>''

## 删除比赛阶段

api：api/league/delmatchstage

参数：传入id（match\_stage表id）

成功 code=>200，msg=>''

## 联赛积分排名

api：api/league/getmatchranklist

参数：league\_id（联赛id）、match\_stage\_id（联赛阶段id）、match\_group\_id（联赛分组id）

成功 code=>200, msg=>'', data=>[]

## 创建比赛结果数据

api：api/league/creatematchrecord

参数：照match\_record数据表字段

## 更新比赛结果数据

api：api/league/updatematchrecord

参数：照match\_record数据表字段

参数：league\_id（联赛id）、match\_stage\_id（联赛阶段id）、match\_group\_id（联赛分组id）

成功 code=>200, msg=>'', data=>[]

## 发布比赛结果

api：api/league/publishmatchrecord

参数：照match\_record数据表字段

成功 code=>200, msg=>'',

## 比赛结果列表（根据比赛时间、阶段、分组）

api：api/league/matchrecordlistbystage

参数：

## 创建联赛文章

api：api/league/createleaguearticle

参数：artcile表字段，category:3,organization\_type:4,status:-1(下架),

成功 code=>200, msg=>'',

## 编辑联赛文章

api：api/league/updateleaguearticle

参数：artcile表字段

成功 code=>200, msg=>'',

## 删除联赛文章

api：api/league/delleaguearticle

参数：artcile表id

成功 code=>200, msg=>'',

## 联赛文章列表

api： api/league/getleaguearticlelist

参数：league\_id（联赛 match表id）或者organization\_id（传入联赛id match表）、organization\_type:4、category: 3、page(第x页) 一页10条数据

成功 code=>200，msg=>''， data=>[]

## 联赛文章列表（页码）

api： api/league/getleaguearticlepage

参数：league\_id（联赛 match表id）或者organization\_id（传入联赛id match表）、organization\_type:4、category: 3、page(第x页) 一页10条数据

成功 code=>200，msg=>''， data=>[]

## 联赛模块评论列表（有页码）

api: api/league/leaguecommentlistpage

翻页传参 {:url('api/ league/leaguecommentlistpage ')}/?page=页数

post:

comment\_type 评论类型 1联赛主页 | 2联赛单场比赛结果

match\_id（联赛id match表id）

match\_record\_id（0为联赛主页，大于0的数字为联赛单程比赛结果对应 match\_record表id）

response:

成功 code=>200, msg=>, data=>（3层数据结构）, thumbup\_count=>（点赞数），can\_delete=>0/1 （1可删除）

## 联赛模块评论列表

api: api/league/leaguecommentlist

post:

comment\_type 评论类型 1联赛主页 | 2联赛单场比赛结果

match\_id（联赛id match表id）

match\_record\_id（0为联赛主页，大于0的数字为联赛单程比赛结果对应 match\_record表id）

page(第x页) 一页10条数据

response:

成功 code=>200, msg=>, data=>（2层数据结构）, thumbup\_count=>（点赞数），can\_delete=>0/1 （1可删除）

## 发布联赛模块评论

api: api/league/addleaguecomment

post:

comment\_type 评论类型 1联赛主页 | 2联赛单场比赛结果

match\_id（联赛id match表id）

match\_record\_id（0为联赛主页，大于0的数字为联赛单程比赛结果对应 match\_record表id）

comment 评论文字内容

response:

成功 code=>200, msg=>'', data=>新增id(match\_comment表id)

## 删除联赛模块评论

api： api/league/delleaguecomment

post: id（match\_ comment表id）

response:

成功 code=>200, msg=>'',

## 联赛模块点赞/取消点赞

api： api/league/dianzan

post:

comment\_type 评论类型 1联赛主页 | 2联赛单场比赛结果

match\_id（联赛id match表id）

match\_record\_id（0为联赛主页，大于0的数字为联赛单程比赛结果对应 match\_record表id）

response:

成功 code=>200, msg=>'',thumbup\_count=>（最新的点赞数字统计）

## 获取会员在联赛模块点赞信息

api: api/league/isthumbup

post:

comment\_type 评论类型 1联赛主页 | 2联赛单场比赛结果

match\_id（联赛id match表id）

match\_record\_id（0为联赛主页，大于0的数字为联赛单程比赛结果对应 match\_record表id）

response:

成功 code=>200, msg=>'', thumbsup =>1（有点赞）|0（没点赞）

## 保存比赛阶段晋级球队数据

api：api/league/savematchstageadvteam

参数：

match\_id（联赛 match表id）、match

advteams（晋级球队名单 json数组）：{team\_id,team,match\_group\_id,match\_group,match\_stage\_id,match\_stage,adv\_num}其中adv\_num为晋级序号

## 比赛阶段晋级球队列表（无分页）

api：api/league/getmatchstageadvteams

参数：league\_id（联赛 match表id）

成功 code=>200，msg=>''， data=>

## 比赛阶段晋级球队列表

api：api/league/ getmatchstageadvteamlist

参数：league\_id（联赛 match表id）、page(第x页) 一页10条数据

成功 code=>200，msg=>''， data=>

## 比赛阶段晋级球队列表（页码）

api：api/league/ getmatchstageadvteampage

参数：league\_id（联赛 match表id）、page(第x页) 一页10条数据

成功 code=>200，msg=>''， data=>

# 裁判员Referee

## 获取裁判员列表(有页码)

api: api/referee/refereelistpage

翻页传参: {:url('api/match/matchlsitpage')}?page=页数

参数: referee数据表字段、keyword（搜索referee字段）

response:

成功 code=>200, msg=>, data=>（3层数据结构）

## 获取裁判员列表

api: api/referee/refereelist

参数：referee数据表字段、keyword（搜索referee字段）、page（第几页）

response:

成功 code=>200, msg=>, data=>（2层数据结构）

## 注册裁判员信息

api: createReferee

post: 身份证、裁判证书、referee表字段

PS：身份证书和裁判证件，也一并放到data里

response:

成功 code=>200, msg=>, data=>referee表新增id

## 编辑裁判员信息

api: updateReferee

post: 身份证、裁判证书、referee表字段

PS：身份证书和裁判证件，也一并放到data里

response:

成功 code=>200, msg=>

## 获取会员裁判员数据

api: getMemberReferee

post: member\_id （可不传， 不传就获取当前会员）

response:

成功 code=>200, msg=>, data=>''

PS：code返回200，即会员有裁判员数据，data[status\_num]区分裁判员注册审核状态（0审核未通过|1正常已通过审核）；code返回100 会员无裁判员数据

## 评论裁判

api: createRefereeCommentApi

post:

## 获取裁判制裁列表（页码）

api: api/referee/ getmatchrefereepage

翻页传参: {:url('api/referee/ getmatchrefereepage)}?page=页数

请求参数：match\_referee表字段

response:

成功 code=>200, msg=>, data=>（3层数据结构）

## 获取裁判制裁列表

api: api/referee/ getmatchrefereelist

请求参数：match\_referee表字段、page（第x页）

response:

成功 code=>200, msg=>, data=>（2层数据结构）

## 邀请裁判加入比赛

api: inviteRefereeApi

post:

PS，post的data内容为apply表字段；

## 裁判申请|受邀比赛列表（分页）

api: api/referee/matchapplypage

翻页传参: {:url('api/match/matchapplypage)}?page=页数

参数: apply\_type（1申请|2邀请）、referee\_id（裁判id）

response:

成功 code=>200, msg=>, data=>（3层数据结构）

url地址：keeper/referee/matchapply/apply\_id/{match\_referee\_apply表id}

## 裁判申请|受邀比赛列表

api: api/referee/matchapplylist

参数：apply\_type（1申请|2邀请）、referee\_id（裁判id）、page（第x页）

成功 code=>200, msg=>, data=>（2层数据结构）

## 获取裁判申请|邀请比赛详情

api: api/referee/getmatchrefereeapply

参数：match\_referee\_apply表字段

成功 code=>200, msg=>, data=>；data[match]是关联的比赛信息

## 裁判回复比赛邀请

api： api/referee/ replymatchinvit

post：apply\_id（match\_referee\_apply表id）、status（2同意|3拒绝）、reply（回复内容）

成功 code=>200, msg=>

## 裁判列表与比赛邀请|申请关系（分页）

api： api/referee/getrefereepagewithmatchapply

翻页传参: {:url('api/match/ getrefereeoagewithmatchapply)}?page=页数

参数：keyword（裁判名称）、match\_id（比赛id，比传）、match\_referee\_apply表字段

成功code => 200，msg=>'', data=>（3层数据结构）,data[match\_apply]是裁判-比赛申请|邀请数据

## 裁判列表与比赛邀请|申请关系

api： api/referee/getrefereelistwithmatchapply

参数：keyword（裁判名称）、match\_id（比赛id，比传）、match\_referee\_apply表字段、page（第x页）

成功code => 200，msg=>'', data=>（2层数据结构）,data[match\_apply]是裁判-比赛申请|邀请数据

## 裁判最近比赛列表（裁判最近订单）

api: api/referee/refereelastmatch

参数：match\_apple表字段，page（第几页）

成功code=>200，msg=>''，data=>；data[match]是关联的比赛信息，data[record]是关联的比赛成绩信息