**目录**

[READ ME 8](#_Toc515033191)

[路径规则: 8](#_Toc515033192)

[**示例数据都为PHP格式, 不是JS** 9](#_Toc515033193)

[关于前端获取PHP变量: 9](#_Toc515033194)

[前端检测用户是否登陆|注册? 9](#_Toc515033195)

[返回数据格式 9](#_Toc515033196)

[社群Group 9](#_Toc515033197)

[获取社群列表 9](#_Toc515033198)

[获取社群列表 无page 10](#_Toc515033199)

[获取社群列表带page 10](#_Toc515033200)

[获取社群用户带page 10](#_Toc515033201)

[编辑社群 10](#_Toc515033202)

[操作社群 10](#_Toc515033203)

[创建社群 11](#_Toc515033204)

[加入社群 11](#_Toc515033205)

[退出社群/将他人踢出社群 11](#_Toc515033206)

[获取社群详情 11](#_Toc515033207)

[获取member没参加的社群/其他社群list 12](#_Toc515033208)

[奖金池/擂台Pool 12](#_Toc515033209)

[获取擂台列表 12](#_Toc515033210)

[获取擂台列表 无page 12](#_Toc515033211)

[获取擂台列表带page 12](#_Toc515033212)

[获取擂台用户带page 12](#_Toc515033213)

[编辑擂台 13](#_Toc515033214)

[创建擂台 13](#_Toc515033215)

[打卡Punch 13](#_Toc515033216)

[获取打卡列表 13](#_Toc515033217)

[获取打卡列表 无page 14](#_Toc515033218)

[获取打卡列表带page 14](#_Toc515033219)

[获取社群的打卡记录带page 14](#_Toc515033220)

[创建打卡 14](#_Toc515033221)

[会员荣誉 15](#_Toc515033222)

[创建会员荣誉 15](#_Toc515033223)

[编辑会员荣誉 15](#_Toc515033224)

[会员荣誉列表 15](#_Toc515033225)

[会员荣誉列表（页码） 16](#_Toc515033226)

[获取会员个人荣誉与球队荣誉集合列表 16](#_Toc515033227)

[比赛技术数据 16](#_Toc515033228)

[球员赛季数据 16](#_Toc515033229)

[球队录入比赛球员技术数据 17](#_Toc515033230)

[球队荣誉 17](#_Toc515033231)

[创建球队荣誉 17](#_Toc515033232)

[编辑球队荣誉 18](#_Toc515033233)

[球队荣誉列表（页码） 18](#_Toc515033234)

[球队荣誉列表 18](#_Toc515033235)

[球队最近荣誉列表 19](#_Toc515033236)

[删除球队荣誉 19](#_Toc515033237)

[获取球队成员的荣誉列表（页码） 19](#_Toc515033238)

[获取球队成员的荣誉列表 20](#_Toc515033239)

[球队比（约）赛 20](#_Toc515033240)

[创建球队比赛 20](#_Toc515033241)

[编辑球队比赛 21](#_Toc515033242)

[直接录入球队比赛 21](#_Toc515033243)

[保存球队友谊赛信息+比赛战绩数据（不使用了） 22](#_Toc515033244)

[发送比赛认领信息给对方球队 22](#_Toc515033245)

[比赛列表（有页码） 22](#_Toc515033246)

[比赛列表 23](#_Toc515033247)

[比赛列表（所有数据） 23](#_Toc515033248)

[比赛列表包含比赛战绩、出赛球员（有页码） 24](#_Toc515033249)

[球队认领比赛列表（分页） 24](#_Toc515033250)

[球队认领比赛列表 25](#_Toc515033251)

[球队认领比赛数据提交 25](#_Toc515033252)

[比赛管理操作 25](#_Toc515033253)

[球队成员报名参加比赛 26](#_Toc515033254)

[提交约赛应战信息 26](#_Toc515033255)

[回复约赛应战 26](#_Toc515033256)

[球队参加比赛申请列表（页码） 27](#_Toc515033257)

[球队参加比赛申请列表 27](#_Toc515033258)

[球队战绩列表（有页码） 28](#_Toc515033259)

[球队战绩列表 28](#_Toc515033260)

[球队战绩列表（所有数据） 29](#_Toc515033261)

[球队认领完成比赛列表（分页） 29](#_Toc515033262)

[球队认领完成比赛列表 30](#_Toc515033263)

[历史对手球队列表（有页码） 30](#_Toc515033264)

[历史对手球队列表 30](#_Toc515033265)

[历史对手球队列表（所有数据） 31](#_Toc515033266)

[比赛战绩-会员关联列表（页码） 31](#_Toc515033267)

[比赛战绩-会员关联列表 32](#_Toc515033268)

[比赛战绩-会员关联列表（所有数据） 32](#_Toc515033269)

[分页获取lat,lng最近数据 32](#_Toc515033270)

[球员参赛列表（页码） 33](#_Toc515033271)

[球员参赛列表 33](#_Toc515033272)

[已登记技术数据的比赛列表 34](#_Toc515033273)

[已登记技术数据的比赛列表（页码） 34](#_Toc515033274)

[印象标签 34](#_Toc515033275)

[印象标签列表（所有数据无分页） 34](#_Toc515033276)

[发表印象标签评论（一个标签只能评论一次） 35](#_Toc515033277)

[获取印象评论记录列表 36](#_Toc515033278)

[球队 36](#_Toc515033279)

[创建球队 36](#_Toc515033280)

[编辑球队 36](#_Toc515033281)

[球队列表（有页码） 37](#_Toc515033282)

[球队列表（无页码） 37](#_Toc515033283)

[搜索球队列表（有页码） 37](#_Toc515033284)

[搜索球队列表 38](#_Toc515033285)

[我的球队列表 38](#_Toc515033286)

[球队成员列表 38](#_Toc515033287)

[移除球队成员（只有领队可操作） 39](#_Toc515033288)

[球队成员申请离队（成员自己操作） 39](#_Toc515033289)

[申请加入球队 40](#_Toc515033290)

[回复申请加入球队结果 40](#_Toc515033291)

[添加平台会员（学生）进入球队 40](#_Toc515033292)

[平台会员回复加入球队邀请 41](#_Toc515033293)

[添加非平台会员进入球队 41](#_Toc515033294)

[离队成员回到球队 41](#_Toc515033295)

[创建球队活动 42](#_Toc515033296)

[修改球队活动 42](#_Toc515033297)

[直接创建并录入活动 43](#_Toc515033298)

[球队活动列表（有页码） 43](#_Toc515033299)

[球队活动列表 43](#_Toc515033300)

[球队最新活动记录 44](#_Toc515033301)

[训练营球队最新活动记录 44](#_Toc515033302)

[球队热门活动记录 45](#_Toc515033303)

[报名参加球队活动 45](#_Toc515033304)

[球队活动-会员关联列表（有页码） 46](#_Toc515033305)

[球队活动-会员关联列表 46](#_Toc515033306)

[球队活动-会员关联列表（无分页） 46](#_Toc515033307)

[球队活动、比赛评论列表（有页码） 47](#_Toc515033308)

[球队活动、比赛评论列表 47](#_Toc515033309)

[发布球队活动、比赛评论 48](#_Toc515033310)

[删除球队评论 48](#_Toc515033311)

[球队活动、比赛点赞/取消点赞 49](#_Toc515033312)

[获取球队活动、比赛点赞状态 49](#_Toc515033313)

[最新比赛记录 50](#_Toc515033314)

[训练营球队最新比赛记录 50](#_Toc515033315)

[热门比赛记录 51](#_Toc515033316)

[联赛 League 51](#_Toc515033317)

[注册联赛组织 51](#_Toc515033318)

[编辑联赛组织 51](#_Toc515033319)

[创建联赛 52](#_Toc515033320)

[编辑联赛 52](#_Toc515033321)

[平台展示联赛列表 52](#_Toc515033322)

[平台展示联赛列表（页码） 53](#_Toc515033323)

[联赛球队列表 53](#_Toc515033324)

[联赛球队列表（页码） 53](#_Toc515033325)

[联赛球队报名申请列表 54](#_Toc515033326)

[联赛球队列表（页码） 54](#_Toc515033327)

[球队报名申请联赛 54](#_Toc515033328)

[回复联赛球队报名申请 55](#_Toc515033329)

[裁判员Referee 55](#_Toc515033330)

[获取裁判员列表(有页码) 55](#_Toc515033331)

[获取裁判员列表 55](#_Toc515033332)

[注册裁判员信息 56](#_Toc515033333)

[编辑裁判员信息 56](#_Toc515033334)

[获取会员裁判员数据 56](#_Toc515033335)

[评论裁判 57](#_Toc515033336)

[获取裁判制裁列表（页码） 57](#_Toc515033337)

[获取裁判制裁列表 57](#_Toc515033338)

[邀请裁判加入比赛 58](#_Toc515033339)

[裁判申请|受邀比赛列表（分页） 58](#_Toc515033340)

[裁判申请|受邀比赛列表 58](#_Toc515033341)

[获取裁判申请|邀请比赛详情 59](#_Toc515033342)

[裁判回复比赛邀请 59](#_Toc515033343)

[裁判列表与比赛邀请|申请关系（分页） 59](#_Toc515033344)

[裁判列表与比赛邀请|申请关系 60](#_Toc515033345)

[裁判最近比赛列表（裁判最近订单） 60](#_Toc515033346)

# 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

# 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**

# 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=>

# 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

## 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

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=>'',

# 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