**目录**

[READ ME 6](#_Toc512602592)

[路径规则: 6](#_Toc512602593)

[**示例数据都为PHP格式, 不是JS** 7](#_Toc512602594)

[关于前端获取PHP变量: 7](#_Toc512602595)

[前端检测用户是否登陆|注册? 7](#_Toc512602596)

[返回数据格式 7](#_Toc512602597)

[比赛技术数据 8](#_Toc512602598)

[球员赛季数据 8](#_Toc512602599)

[球队录入比赛球员技术数据 8](#_Toc512602600)

[球队荣誉 9](#_Toc512602601)

[创建球队荣誉 9](#_Toc512602602)

[编辑球队荣誉 9](#_Toc512602603)

[球队荣誉列表（页码） 9](#_Toc512602604)

[球队荣誉列表 10](#_Toc512602605)

[球队最近荣誉列表 10](#_Toc512602606)

[删除球队荣誉 10](#_Toc512602607)

[获取球队成员的荣誉列表（页码） 11](#_Toc512602608)

[获取球队成员的荣誉列表 11](#_Toc512602609)

[球队比（约）赛 12](#_Toc512602610)

[创建球队比赛 12](#_Toc512602611)

[编辑球队比赛 12](#_Toc512602612)

[直接录入球队比赛 13](#_Toc512602613)

[保存球队友谊赛信息+比赛战绩数据（不使用了） 13](#_Toc512602614)

[发送比赛认领信息给对方球队 14](#_Toc512602615)

[比赛列表（有页码） 14](#_Toc512602616)

[比赛列表 14](#_Toc512602617)

[比赛列表（所有数据） 15](#_Toc512602618)

[比赛列表包含比赛战绩、出赛球员（有页码） 15](#_Toc512602619)

[球队认领比赛列表（分页） 16](#_Toc512602620)

[球队认领比赛列表 16](#_Toc512602621)

[球队认领比赛数据提交 16](#_Toc512602622)

[比赛管理操作 17](#_Toc512602623)

[球队成员报名参加比赛 17](#_Toc512602624)

[提交约赛应战信息 17](#_Toc512602625)

[回复约赛应战 18](#_Toc512602626)

[球队参加比赛申请列表（页码） 18](#_Toc512602627)

[球队参加比赛申请列表 19](#_Toc512602628)

[球队战绩列表（有页码） 19](#_Toc512602629)

[球队战绩列表 20](#_Toc512602630)

[球队战绩列表（所有数据） 20](#_Toc512602631)

[球队认领完成比赛列表（分页） 21](#_Toc512602632)

[球队认领完成比赛列表 21](#_Toc512602633)

[历史对手球队列表（有页码） 21](#_Toc512602634)

[历史对手球队列表 22](#_Toc512602635)

[历史对手球队列表（所有数据） 22](#_Toc512602636)

[比赛战绩-会员关联列表（页码） 23](#_Toc512602637)

[比赛战绩-会员关联列表 23](#_Toc512602638)

[比赛战绩-会员关联列表（所有数据） 24](#_Toc512602639)

[分页获取lat,lng最近数据 24](#_Toc512602640)

[球员参赛列表（页码） 24](#_Toc512602641)

[球员参赛列表 25](#_Toc512602642)

[已登记技术数据的比赛列表 25](#_Toc512602643)

[已登记技术数据的比赛列表（页码） 25](#_Toc512602644)

[印象标签 26](#_Toc512602645)

[印象标签列表（所有数据无分页） 26](#_Toc512602646)

[发表印象标签评论（一个标签只能评论一次） 26](#_Toc512602647)

[获取印象评论记录列表 27](#_Toc512602648)

[球队 27](#_Toc512602649)

[创建球队 27](#_Toc512602650)

[编辑球队 28](#_Toc512602651)

[球队列表（有页码） 28](#_Toc512602652)

[球队列表（无页码） 29](#_Toc512602653)

[搜索球队列表（有页码） 29](#_Toc512602654)

[搜索球队列表 29](#_Toc512602655)

[我的球队列表 30](#_Toc512602656)

[球队成员列表 30](#_Toc512602657)

[移除球队成员（只有领队可操作） 30](#_Toc512602658)

[球队成员申请离队（成员自己操作） 31](#_Toc512602659)

[申请加入球队 31](#_Toc512602660)

[回复申请加入球队结果 31](#_Toc512602661)

[添加平台会员（学生）进入球队 32](#_Toc512602662)

[平台会员回复加入球队邀请 32](#_Toc512602663)

[添加非平台会员进入球队 33](#_Toc512602664)

[离队成员回到球队 33](#_Toc512602665)

[创建球队活动 33](#_Toc512602666)

[修改球队活动 34](#_Toc512602667)

[直接创建并录入活动 34](#_Toc512602668)

[球队活动列表（有页码） 34](#_Toc512602669)

[球队活动列表 35](#_Toc512602670)

[球队最新活动记录 35](#_Toc512602671)

[训练营球队最新活动记录 36](#_Toc512602672)

[球队热门活动记录 36](#_Toc512602673)

[报名参加球队活动 37](#_Toc512602674)

[球队活动-会员关联列表（有页码） 37](#_Toc512602675)

[球队活动-会员关联列表 38](#_Toc512602676)

[球队活动-会员关联列表（无分页） 38](#_Toc512602677)

[球队活动、比赛评论列表（有页码） 38](#_Toc512602678)

[球队活动、比赛评论列表 39](#_Toc512602679)

[发布球队活动、比赛评论 39](#_Toc512602680)

[球队活动、比赛点赞/取消点赞 40](#_Toc512602681)

[获取球队活动、比赛点赞状态 40](#_Toc512602682)

[最新比赛记录 41](#_Toc512602683)

[训练营球队最新比赛记录 41](#_Toc512602684)

[热门比赛记录 42](#_Toc512602685)

[联赛 League 42](#_Toc512602686)

[注册联赛组织 42](#_Toc512602687)

[编辑联赛组织 43](#_Toc512602688)

[创建联赛 43](#_Toc512602689)

[编辑联赛 43](#_Toc512602690)

[裁判员Referee 44](#_Toc512602691)

[获取裁判员列表(有页码) 44](#_Toc512602692)

[获取裁判员列表 44](#_Toc512602693)

[注册裁判员信息 44](#_Toc512602694)

[编辑裁判员信息 45](#_Toc512602695)

[获取会员裁判员数据 45](#_Toc512602696)

[评论裁判 45](#_Toc512602697)

[获取裁判制裁列表（页码） 46](#_Toc512602698)

[获取裁判制裁列表 46](#_Toc512602699)

[邀请裁判加入比赛 46](#_Toc512602700)

[裁判申请|受邀比赛列表（分页） 47](#_Toc512602701)

[裁判申请|受邀比赛列表 47](#_Toc512602702)

[获取裁判申请|邀请比赛详情 47](#_Toc512602703)

[裁判回复比赛邀请 48](#_Toc512602704)

[裁判列表与比赛邀请|申请关系（分页） 48](#_Toc512602705)

[裁判列表与比赛邀请|申请关系 48](#_Toc512602706)

[裁判最近比赛列表（裁判最近订单） 49](#_Toc512602707)

# 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}]}

# 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，

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）

## 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）

## 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

response:

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

## 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=>''

# 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