**目录**

[READ ME 11](#_Toc517706373)

[路径规则: 11](#_Toc517706374)

[**示例数据都为PHP格式, 不是JS** 11](#_Toc517706375)

[关于前端获取PHP变量: 11](#_Toc517706376)

[前端检测用户是否登陆|注册? 12](#_Toc517706377)

[返回数据格式 12](#_Toc517706378)

[社群Group 12](#_Toc517706379)

[获取社群列表 12](#_Toc517706380)

[获取社群列表 无page 12](#_Toc517706381)

[获取社群列表带page 13](#_Toc517706382)

[获取社群用户带page 13](#_Toc517706383)

[编辑社群 13](#_Toc517706384)

[操作社群 13](#_Toc517706385)

[创建社群 13](#_Toc517706386)

[加入社群 14](#_Toc517706387)

[退出社群/将他人踢出社群 14](#_Toc517706388)

[将多人批量踢出社群 14](#_Toc517706389)

[获取社群详情 14](#_Toc517706390)

[获取member没参加的社群/其他社群list 15](#_Toc517706391)

[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 15](#_Toc517706392)

[社群申请group\_apply 16](#_Toc517706393)

[获取申请列表page 16](#_Toc517706394)

[不带page 16](#_Toc517706395)

[[邀请加入|申请加入,写入一条]|[同意加入|拒绝加入] 16](#_Toc517706396)

[奖金池/擂台Pool 16](#_Toc517706397)

[获取擂台列表 16](#_Toc517706398)

[获取擂台列表 无page 17](#_Toc517706399)

[获取擂台列表带page 17](#_Toc517706400)

[获取擂台用户带page 17](#_Toc517706401)

[编辑擂台 17](#_Toc517706402)

[创建擂台 18](#_Toc517706403)

[打卡Punch 18](#_Toc517706404)

[获取打卡列表 18](#_Toc517706405)

[获取打卡列表 无page 18](#_Toc517706406)

[获取打卡列表带page 18](#_Toc517706407)

[获取社群的打卡记录带page 19](#_Toc517706408)

[创建打卡 19](#_Toc517706409)

[打卡评论 20](#_Toc517706410)

[获取评论列表 20](#_Toc517706411)

[获取评论列表byPage 20](#_Toc517706412)

[点赞 20](#_Toc517706413)

[评论 20](#_Toc517706414)

[打赏 21](#_Toc517706415)

[会员荣誉 21](#_Toc517706416)

[创建会员荣誉 21](#_Toc517706417)

[编辑会员荣誉 21](#_Toc517706418)

[会员荣誉列表 22](#_Toc517706419)

[会员荣誉列表（页码） 22](#_Toc517706420)

[获取会员个人荣誉与球队荣誉集合列表 22](#_Toc517706421)

[比赛技术数据 23](#_Toc517706422)

[球员赛季数据 23](#_Toc517706423)

[球队录入比赛球员技术数据 23](#_Toc517706424)

[球队荣誉 24](#_Toc517706425)

[创建球队荣誉 24](#_Toc517706426)

[编辑球队荣誉 24](#_Toc517706427)

[球队荣誉列表（页码） 24](#_Toc517706428)

[球队荣誉列表 25](#_Toc517706429)

[球队最近荣誉列表 25](#_Toc517706430)

[删除球队荣誉 25](#_Toc517706431)

[获取球队成员的荣誉列表（页码） 26](#_Toc517706432)

[获取球队成员的荣誉列表 26](#_Toc517706433)

[球队比（约）赛 27](#_Toc517706434)

[创建球队比赛 27](#_Toc517706435)

[编辑球队比赛 27](#_Toc517706436)

[直接录入球队比赛 28](#_Toc517706437)

[保存球队友谊赛信息+比赛战绩数据（不使用了） 28](#_Toc517706438)

[发送比赛认领信息给对方球队 29](#_Toc517706439)

[比赛列表（有页码） 29](#_Toc517706440)

[比赛列表 29](#_Toc517706441)

[比赛列表（所有数据） 30](#_Toc517706442)

[比赛列表包含比赛战绩、出赛球员（有页码） 30](#_Toc517706443)

[球队认领比赛列表（分页） 31](#_Toc517706444)

[球队认领比赛列表 31](#_Toc517706445)

[球队认领比赛数据提交 31](#_Toc517706446)

[比赛管理操作 32](#_Toc517706447)

[球队成员报名参加比赛 32](#_Toc517706448)

[提交约赛应战信息 32](#_Toc517706449)

[回复约赛应战 33](#_Toc517706450)

[球队参加比赛申请列表（页码） 33](#_Toc517706451)

[球队参加比赛申请列表 34](#_Toc517706452)

[球队战绩列表（有页码） 34](#_Toc517706453)

[球队战绩列表 35](#_Toc517706454)

[球队战绩列表（所有数据） 35](#_Toc517706455)

[球队认领完成比赛列表（分页） 36](#_Toc517706456)

[球队认领完成比赛列表 36](#_Toc517706457)

[历史对手球队列表（有页码） 36](#_Toc517706458)

[历史对手球队列表 37](#_Toc517706459)

[历史对手球队列表（所有数据） 37](#_Toc517706460)

[比赛战绩-会员关联列表（页码） 38](#_Toc517706461)

[比赛战绩-会员关联列表 38](#_Toc517706462)

[比赛战绩-会员关联列表（所有数据） 39](#_Toc517706463)

[分页获取lat,lng最近数据 39](#_Toc517706464)

[球员参赛列表（页码） 39](#_Toc517706465)

[球员参赛列表 40](#_Toc517706466)

[已登记技术数据的比赛列表 40](#_Toc517706467)

[已登记技术数据的比赛列表（页码） 40](#_Toc517706468)

[印象标签 41](#_Toc517706469)

[印象标签列表（所有数据无分页） 41](#_Toc517706470)

[发表印象标签评论（一个标签只能评论一次） 41](#_Toc517706471)

[获取印象评论记录列表 42](#_Toc517706472)

[球队 42](#_Toc517706473)

[创建球队 42](#_Toc517706474)

[编辑球队 43](#_Toc517706475)

[球队列表（有页码） 43](#_Toc517706476)

[球队列表（无页码） 44](#_Toc517706477)

[搜索球队列表（有页码） 44](#_Toc517706478)

[搜索球队列表 44](#_Toc517706479)

[我的球队列表 45](#_Toc517706480)

[球队成员列表 45](#_Toc517706481)

[移除球队成员（只有领队可操作） 45](#_Toc517706482)

[球队成员申请离队（成员自己操作） 46](#_Toc517706483)

[申请加入球队 46](#_Toc517706484)

[回复申请加入球队结果 46](#_Toc517706485)

[添加平台会员（学生）进入球队 47](#_Toc517706486)

[平台会员回复加入球队邀请 47](#_Toc517706487)

[添加非平台会员进入球队 48](#_Toc517706488)

[离队成员回到球队 48](#_Toc517706489)

[创建球队活动 48](#_Toc517706490)

[修改球队活动 49](#_Toc517706491)

[直接创建并录入活动 49](#_Toc517706492)

[球队活动列表（有页码） 49](#_Toc517706493)

[球队活动列表 50](#_Toc517706494)

[球队最新活动记录 50](#_Toc517706495)

[训练营球队最新活动记录 51](#_Toc517706496)

[球队热门活动记录 51](#_Toc517706497)

[报名参加球队活动 52](#_Toc517706498)

[球队活动-会员关联列表（有页码） 52](#_Toc517706499)

[球队活动-会员关联列表 53](#_Toc517706500)

[球队活动-会员关联列表（无分页） 53](#_Toc517706501)

[球队活动、比赛评论列表（有页码） 53](#_Toc517706502)

[球队活动、比赛评论列表 54](#_Toc517706503)

[发布球队活动、比赛评论 54](#_Toc517706504)

[删除球队评论 55](#_Toc517706505)

[球队活动、比赛点赞/取消点赞 55](#_Toc517706506)

[获取球队活动、比赛点赞状态 56](#_Toc517706507)

[最新比赛记录 56](#_Toc517706508)

[训练营球队最新比赛记录 57](#_Toc517706509)

[热门比赛记录 57](#_Toc517706510)

[联赛 League 58](#_Toc517706511)

[注册联赛组织 58](#_Toc517706512)

[编辑联赛组织 58](#_Toc517706513)

[创建联赛 58](#_Toc517706514)

[编辑联赛 59](#_Toc517706515)

[平台展示联赛列表 59](#_Toc517706516)

[平台展示联赛列表（页码） 59](#_Toc517706517)

[联赛球队列表 60](#_Toc517706518)

[联赛球队列表（页码） 60](#_Toc517706519)

[联赛球队报名申请列表 60](#_Toc517706520)

[联赛球队列表（页码） 61](#_Toc517706521)

[球队报名申请联赛 61](#_Toc517706522)

[回复联赛球队报名申请 61](#_Toc517706523)

[通知报名联赛球队完善参赛球员名单 62](#_Toc517706524)

[联赛报名球员列表 62](#_Toc517706525)

[联赛报名球员列表（页码） 62](#_Toc517706526)

[创建联赛分组 63](#_Toc517706527)

[编辑联赛分组 63](#_Toc517706528)

[联赛分组列表 63](#_Toc517706529)

[联赛分组列表（页码） 64](#_Toc517706530)

[联赛分组包含组内球队列表 64](#_Toc517706531)

[联赛分组包含组内球队列表（页码） 64](#_Toc517706532)

[保存联赛球队分组数据 65](#_Toc517706533)

[获取联赛球队成员列表（无分页） 65](#_Toc517706534)

[获取联赛球队成员列表 65](#_Toc517706535)

[获取联赛球队成员列表（页码） 66](#_Toc517706536)

[球队提交参数成员名单 66](#_Toc517706537)

[设置联赛章程 66](#_Toc517706538)

[获取联赛组织人员列表 67](#_Toc517706539)

[获取联赛组织人员列表（页码） 67](#_Toc517706540)

[检查会员有无联赛组织人员信息 67](#_Toc517706541)

[邀请会员加入联赛组织 68](#_Toc517706542)

[会员回复联赛组织邀请 68](#_Toc517706543)

[删除联赛组织人员 68](#_Toc517706544)

[联赛工作人员列表 69](#_Toc517706545)

[联赛工作人员列表（页码） 69](#_Toc517706546)

[检查会员有无联赛工作人员信息 69](#_Toc517706547)

[邀请会员联赛工作人员 70](#_Toc517706548)

[会员回复联赛工作人员邀请 70](#_Toc517706549)

[会员申请联赛工作人员 70](#_Toc517706550)

[回复联赛工作人员申请 70](#_Toc517706551)

[删除联赛工作人员 71](#_Toc517706552)

[获取联赛工作人员（裁判）列表 71](#_Toc517706553)

[获取联赛工作人员（裁判）列表 页码 71](#_Toc517706554)

[创建赛程 72](#_Toc517706555)

[编辑赛程 72](#_Toc517706556)

[赛程列表（无分页） 72](#_Toc517706557)

[赛程列表 72](#_Toc517706558)

[赛程列表（页码） 73](#_Toc517706559)

[删除赛程 73](#_Toc517706560)

[获取根据阶段分组的赛程列表 73](#_Toc517706561)

[根据联赛分组（小组）生成预览赛程 74](#_Toc517706562)

[批量保存比赛赛程数据 74](#_Toc517706563)

[比赛阶段列表 74](#_Toc517706564)

[比赛阶段列表（页码） 75](#_Toc517706565)

[新建比赛阶段 75](#_Toc517706566)

[修改比赛阶段 75](#_Toc517706567)

[删除比赛阶段 76](#_Toc517706568)

[裁判员Referee 76](#_Toc517706569)

[获取裁判员列表(有页码) 76](#_Toc517706570)

[获取裁判员列表 76](#_Toc517706571)

[注册裁判员信息 77](#_Toc517706572)

[编辑裁判员信息 77](#_Toc517706573)

[获取会员裁判员数据 77](#_Toc517706574)

[评论裁判 78](#_Toc517706575)

[获取裁判制裁列表（页码） 78](#_Toc517706576)

[获取裁判制裁列表 78](#_Toc517706577)

[邀请裁判加入比赛 79](#_Toc517706578)

[裁判申请|受邀比赛列表（分页） 79](#_Toc517706579)

[裁判申请|受邀比赛列表 79](#_Toc517706580)

[获取裁判申请|邀请比赛详情 80](#_Toc517706581)

[裁判回复比赛邀请 80](#_Toc517706582)

[裁判列表与比赛邀请|申请关系（分页） 80](#_Toc517706583)

[裁判列表与比赛邀请|申请关系 81](#_Toc517706584)

[裁判最近比赛列表（裁判最近订单） 81](#_Toc517706585)

# 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;**

# 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=>'',

## 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=>

## 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（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（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=>''，

## 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=>[]

## 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=>''

# 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