

盈创 IM 服务端接口规范

说明

1、为防止非法请求和恶意请求，服务端采用 OAuth2 授权协议。

2、接口返回值说明：

接口调用成功：此时接口返回 JSON 对象中 resultCode 值为 1，resultMsg 为接口提示消息，data 为数据节点，例如评论、赞、商务圈消息等。

如：

```
{
  "resultCode": "1",
  "resultMsg": "提示信息",
  "data": {}
}
```

接口调用失败：此时接口返回 JSON 对象中 resultCode 值为非 1 值，resultMsg 为错误消息。

如：

```
{
  "resultCode": "0或大于1000000",
  "resultMsg": "错误提示"
}
```

1.BasicController ---基础接口

~获取应用配置 /config

① 请求参数：

参数名	参数类型	参数说明	是否必填
access_token	String	访问令牌	Y

② 返回值属性说明：

XMPPDomain #XMPP 主机 IP
XMPPHost #XMPP 主机 IP
androidAppUrl #安卓更新地址
androidExplain #安卓更新说明
androidVersion #安卓版本号
apiUrl #酷聊接口地址

audioLen #语音时长

displayRedPacket #是否开放 IOS 红包

downloadAvatarUrl #头像下载地址

downloadUrl #资源下载地址

freeswitch #视频会议主机 IP

helpUrl #帮助 url

id #id

iosAppUrl #安卓更新地址

iosExplain #安卓更新说明

iosVersion #安卓版本号

liveUrl #直播地址

meetingHost #视频会议主机 IP

uploadUrl #文件上传服务地址

shareUrl #分享 url

videoLen # 视频时长

③ 返回数据示例

```

currentTime: 1504063509
data:
  XMPPDomain: "www.shiku.co"
  XMPPHost: "www.shiku.co"
  androidAppUrl: ""
  androidExplain: ""
  androidVersion: 343
  apiUrl: "http://imapi.youjob.co/"
  audioLen: "20"
  displayRedPacket: 1
  distance: 20
  downloadAvatarUrl: "http://file.shiku.co/"
  downloadUrl: "http://file.shiku.co/"
  freeswitch: "120.24.211.24"
  helpUrl: ""
  id: 10000
  iosAppUrl: ""
  iosExplain: ""
  iosVersion: 0
  liveUrl: "rtmp://v1.one-tv.com:1935/live/"
  meetingHost: "120.24.211.24"
  shareUrl: ""
  softUrl: ""
  uploadUrl: "http://upload.shiku.co/"
  videoLen: "20"
  XMPPDomain: "www.shiku.co"
  XMPPHost: "www.shiku.co"
  resultCode: 1

```

~获取服务器当前时间 /getCurrentTime

① 请求参数：

参数名	参数类型	参数说明	是否必填
access_token	String	访问令牌	Y

② 返回值属性说明:

data #服务器当前时间毫秒数

③ 返回数据示例

```
currentTime: 1504066912
data: 1504066912
resultCode: 1
```

~发送短信验证码 /basic/randcode/sendSms

① 请求参数:

参数名	参数类型	默认值	参数说明	必填
access_token	String		访问令牌	Y
telephone	String		手机号码	Y
areaCode	String	86	区号	N
language	String	zh	语言 zh: 中文 eg: 英文	N
isRegister	int	1	是否注册	N

② 返回值属性说明:

code #验证码

③ 返回数据示例

```
currentTime: 1504067346
data:
  code: "321334"
resultCode: 1
```

~config 配置设置 /config/set

注: /config/set 接口有两个其中 GET 是页面获取配置, POST 接口是设置 config 配置, 参数如下:

① 请求参数:

参数名	参数类型	参数说明	是否必填
access_token	String	访问令牌	Y
config	Config	Config 实体	Y

② 返回值属性说明: 返回值为 jsp 页面

~验证手机号 /verify/telephone

① 请求参数:

参数名	参数类型	默认值	参数说明	是否必填
areaCode	String	"86"	区号	N
telephone	String		电话号码	Y

② 返回值属性说明:

resultCode : 1 成功 0 失败

③ 返回数据示例

```
currentTime: 1504427088
resultCode: 1
resultMsg: "手机号未注册"
```

```
currentTime: 1504427114
resultCode: 0
resultMsg: "手机号已注册"
```

2.UserController --- 用户相关接口

~用户注册接口 /user/register

请求参数:

参数名	参数类型	参数说明	是否必填
telephone	String	11 位手机号码	Y
password	String	密码, MD5 加密	Y
nickname	String	昵称	Y
userType	int	用户类型 0=普通; 2=公众号	N
description	String	签名	N
sex	int	性别	N
birthday	long	生日	N
countryId	int	国家 Id	N
provinceId	int	省份 Id	N
cityId	int	城市 Id	N
areaId	int	区县 Id	N
apiVersion	String	客户端使用的接口版本号	N
model	String	设备型号	N
osVersion	String	设备操作系统版本号	N
serial	String	设备序列号	N

latitude	double	维度	N
longitude	double	经度	N
location	String	位置文本	N

请求成功:

```
{
  "data": {
    "access_token": "e6de7766f337415693e8869ec830b45b",
    "expires_in": 604800,
    "nickname": "测试昵称123456",
    "userId": 100005
  },
  "resultCode": 1
}
```

~用户登录接口 /user/login

请求参数:

参数名	参数类型	参数说明	是否必填
telephone	String	手机号码, 使用 MD5 加密	Y
password	String	密码, 使用 MD5 加密	Y
apiVersion	String	版本	N
model	String	设备型号	N
osVersion	String	设备操作系统版本号	N
serial	String	设备序列号	N
latitude	double	纬度	N
longitude	double	经度	N
location	String	位置文本	N

请求成功:

```
{
  "data": {
    "access_token": "2e707085469844aebd36a98136f90520",
    "companyId": 1,
    "expires_in": 604800,
    "login": {
      "isFirstLogin": 0,
      "latitude": 0,
      "loginTime": 1413188143,
      "longitude": 0
    },
    "nickname": "罗融春",
    "userId": 100001
  },
  "resultCode": 1
}
```

~自动登录 /user/login/auto

请求参数：

参数名	参数类型	参数说明	是否必填
access_token	String	访问令牌	Y
userId	int	用户 Id	Y
serial	String	设备序列号	Y
appId	String		N

说明：

```
{
  "data": {
    "serialStatus": 3, // 1=没有设备号、2=设备号一致、3=设备号不一致
    "tokenExists": 0 // 1=令牌存在、0=令牌不存在
  },
  "resultCode": 1
}
```

~用户注销接口 /user/logout

请求参数：

参数名	参数类型	参数说明	是否必填
access_token	String	访问令牌	Y
areaCode	String	区号（默认“86”）	N
telephone	String	手机号码，使用 MD5 加密	Y

注：无返回数据

~上传用户公钥密钥 /user/uploadKey

请求参数：

参数名	参数类型	参数说明	是否必填
publicKey	String	用户公钥	Y
privateKey	String	用户私钥 用明文密码加密	Y

~用户资料更新 /user/update

请求参数：

参数名	参数类型	参数说明	是否必填
access_token	String	访问令牌	Y

userType	int	用户类型 1=普通; 2=公众号	Y
nickname	String	昵称	Y
description	String	个性签名	N
sex	int	性别	N
birthday	long	生日	N
countryId	int	国家 Id	N
provinceId	int	省份 Id	N
cityId	int	城市 Id	N
areaid	int	区县 Id	N

注: 返回值为 User

用户绑定微信 openid /user/bind/wxcode

参数名	参数类型	参数说明	必填
access_token	string	访问令牌	Y
code	string	微信 code	Y

余额微信提现 /transfer/wx/pay

参数名	参数类型	参数说明	必填
access_token	string	访问令牌	Y
amount	string	提现金额 0.3=30 单位为分 最少提现 0.5 会扣除 千分之六 手续费	Y
secret	string	提现密钥 md5(apiKey+openid+userid) + md5(token+amount+time)	
time	long	请求时间 服务器会检验 允许 5 分钟时差	Y

~百度云 channelId 上传 /user/channelId/set

参数名	参数类型	参数说明	必填
access_token	string	访问令牌	Y
device	string	iOS=3; 安卓=4	Y
channelId	string	百度云推送 onBind 函数返回的 channelId	Y

注: 无返回数据

~用户资料获取 /user/get

请求参数:

参数名	参数类型	参数说明	是否必填
access_token	String	访问令牌	Y
userId	int	用户 Id, 查自己则不传	Y

请求成功:

```
{
  "data": {
    "areald": 0,
    "attCount": 0,
    "attList": [],
    "birthday": 1411130015,
    "cityId": 110100,
    "company": {
      "areald": 0,
      "browseCount": 0,
      "cityId": 0,
      "countryId": 0,
      "createTime": 0,
      "description": "我公司数年以来全体成员秉着产品上精益求精、开发上不断创新的精神，现已成功开发了视佳VOD点歌系统、无线PDA
      餐饮管理系统等。软件以领先的技术，强大的功能、精美的界面、贴心的服务从同行产品中脱颖而出，用户已遍及北京、上海、广州、深圳、天津、
      重庆、昆明、武汉等国内大中小城市和港、澳、台地区，并拓展至东南亚、加拿大、日韩、欧美国家，用户总数已达到数千家，已迅速发展成为国际
      知名的软件开发商。产品不仅受到行",
      "fansCount": 0,
      "id": 1,
      "industryId": 0,
      "isAuth": 0,
      "name": "深圳市技雄信息技术有限公司",
      "natureId": 0,
      "provinceId": 0,
      "scale": 0,
      "sponsorMoney": 0,
      "website": "http://www.voddvd.com"
    },
    "companyId": 1,
    "countryId": 1,
    "createTime": 1411158775,
    "description": "淡淡的忧伤",
    "fansCount": 0,
    "fansList": [],
    "friends": {
      "companyId": 0,
      "createTime": 0,
      "modifyTime": 0,
```



```

        "toUserId": 0,
        "userId": 0
    },
    "friendsCount": 0,
    "friendsList": [],
    "isAuth": 0,
    "level": 0,
    "money": 0,
    "moneyTotal": 0,
    "nickname": "罗融春",
    "provinceId": 110000,
    "sex": 0,
    "status": 1,
    "telephone": "18908983800",
    "userId": 100001,
    "userType": 0,
    "vip": 0
},
"resultCode": 1
}

```

~根据通讯号获取用户资料 /user/getByAccount

请求参数:

参数名	参数类型	参数说明	是否必填
access_token	String	访问令牌	Y
account	String	用户通讯号	Y

~用户搜索 /user/query

请求参数:

参数名	参数类型	参数说明	是否必填
access_token	String	访问令牌	Y
userId	int	每页数据的最后一个 userId	N
pageSize	int	每页数据条数, 默认 10	N
sex	int	性别	N
minAge	int	最小年龄	N
maxAge	int	最大年龄	N
nickname	String	昵称	N

说明: userId 为每页数据按 userId 降序排序的最后一个 used, 也就是最小的 userId

请求成功:

```

{
    "data": [

```

```
{
  "age": 24,
  "birthday": "2014-05-30 17:34:23",
  "headImg": "http://himg.bdimg.com/sys/portrait/item/4ee8e99ba8e6b6a6e5b08fe99d92db30.jpg",
  "headThumImg": "http://himg.bdimg.com/sys/portrait/item/4ee8e99ba8e6b6a6e5b08fe99d92db30.jpg",
  "nickname": "\\u4e2d\\u6587",
  "sex": 1,
  "userId": 11,
  "username": "75511dbaa8368b88bd2954e21ba5c5a325a3758c"
},
{
  "age": 24,
  "birthday": "2014-05-30 17:34:23",
  "headImg": "http://himg.bdimg.com/sys/portrait/item/4ee8e99ba8e6b6a6e5b08fe99d92db30.jpg",
  "headThumImg": "http://himg.bdimg.com/sys/portrait/item/4ee8e99ba8e6b6a6e5b08fe99d92db30.jpg",
  "nickname": "\\u4e2d\\u6587",
  "sex": 1,
  "userId": 10,
  "username": "1567722b5143af661f4d86600f7f6d9231070068"
}
],
"resultCode": 1,
"resultMsg": "用户查询成功"
}
```

~修改密码 /user/password/update

请求参数:

参数名	参数类型	参数说明	是否必填
access_token	String	访问令牌	Y
oldPassword	String	旧密码	Y
newPassword	String	新密码	Y

注: 无返回数据

~重置密码 /user/password/reset

请求参数:

参数名	参数类型	参数说明	是否必填
access_token	String	访问令牌	Y
areaCode	String	区号	N
randcode	String	验证码	Y
telephone	String	手机号码	Y
newPassword	String	新密码	Y

注: 无返回数据

~用户隐私设置修改 /user/settings/update

请求参数:

参数名	参数类型	参数说明	是否必填
access_token	String	访问令牌	Y
allowAtt	int	允许关注 (0=不允许; 1=允许)	N
allowGreet	int	允许打招呼 (0=不允许; 1=允许)	N
friendsVerify	int	加好友验证状态 (0=不验证; 1=需验证)	N
chatRecordTimeOut	double	个人设置 聊天记录 保存时间 1 天 =1 1 小时=0.04	N
chatSyncTimeLen	double	个人设置 单聊聊天记录 同步时长 1 天 =1 1 小时=0.04	N
groupChatSyncTimeLen	double	个人设置 群聊聊天记录 同步时长 1 天 =1 1 小时=0.04	N

注: 无返回数据

~用户隐私设置查询 /user/settings

请求参数:

参数名	参数类型	参数说明	是否必填
access_token	String	访问令牌	Y
userId	int	用户 Id, 查自己则不传	Y

注: 返回 User.Settings 属性说明:

allowAtt=1; #允许关注 (0=不允许; 1=允许)

allowGreet=1; #允许打招呼 (0=不允许; 1=允许)

friendsVerify=0; #加好友验证状态 (0=不验证; 1=需验证)

chatRecordTimeOut double 个人设置 聊天记录 保存时间 1 天 =1 1 小时=0.04

chatSyncTimeLen double 个人设置 单聊聊天记录 同步时长 1 天 =1 1 小时=0.04

groupChatSyncTimeLen double

个人设置 群聊聊天记录 同步时长 1 天 =1 1 小时=0.04

~用户余额充值 /user/Recharge

请求参数:

参数名	参数类型	参数说明	是否必填
access_token	String	访问令牌	Y
money	double	用户 Id, 查自己则不传	Y
type	int	支付方式 1: 支付宝支付 2: 微信支付	Y

注: 返回值说明: balance :充值后的余额

~获取用户余额 /user/getUserMoeny

请求参数:

参数名	参数类型	参数说明	是否必填
access_token	String	访问令牌	Y

注: 返回值说明: balance :充值后的余额

~获取在线/离线用户数量 /user/getUserStatusCount

请求参数:

参数名	参数类型	参数说明	是否必填
access_token	String	访问令牌	Y
pageIndex	int	当前页码数	N
pageSize	int	每页数据条数	N
sign	String	标志	N
startData	String	开始日期	N
endData	String	截止日期	N
type	int	状态值 0: 离线 1: 在线 默认 0	N

注: 返回值为 List<UserStatusCount>

~获取用户在线状态 /user/getOnLine

请求参数:

参数名	参数类型	参数说明	是否必填
access_token	String	访问令牌	Y
userId	int	用户 Id, 查自己则不传	Y

注: 返回值说明: onlinestate 0: 不在线 1: 在线

~举报其它用户或群组 /user/report

请求参数:

参数名	参数类型	参数说明	是否必填
access_token	String	访问令牌	Y
toUserId	int	要举报的用户 Id 举报用户时传	N
roomId	String	群组 ID 举报群组时传	N
reason	int	举报原因 ID	Y

注: 无返回值

~添加到收藏 /emoji/add

请求参数:

参数名	参数类型	参数说明	是否必填
access_toke	String	访问令牌	Y

n			
String	url		Y
String	roomId		Y
String	msgId		Y
String	type		Y

注：无返回值

~收藏的表情列表 /emoji/list

请求参数：

参数名	参数类型	参数说明	是否必填
access_token	String	访问令牌	Y
int	userId	用户 id	Y
int	pageIndex	当前页码	Y
int	pageSize	每页数量，默认 10 条	N

~收藏列表 /collection/list

请求参数：

参数名	参数类型	参数说明	是否必填
access_token	String	访问令牌	Y
int	userId	用户 id	Y
int	type	收藏类型 1.图片 2.视频 3.文件 4.语音 5.文本 6.表情	Y
int	pageIndex	当前页码	Y
int	pageSize	每页数量，默认 10 条	N

~取消收藏 /emoji/delete

请求参数：

参数名	参数类型	参数说明	是否必填
access_token	String	访问令牌	Y
String	emojiId	收藏 id	Y

~添加消息录制 /course/add

请求参数：

参数名	参数类型	参数说明	是否必填
access_token	String	访问令牌	Y
int	userId	用户 id	Y
String	messageIds	消息 id 列表	Y

long	createTime	创建时间	Y
String	courseName	课程名称	Y
String	roomJid	群组 Jid	Y

3.FriendsController ---好友相关

Friends 好友实体 属性说明：

```
Integer blacklist=0; # 是否拉黑（1=是；0=否）
int companyId; # 所属公司 Id
long createTime; # 建立关系时间
Objectid id; # 关系 Id
long modifyTime; # 修改时间
long lastTalkTime; # 最后沟通时间
long msgNum; # 未读消息数量
String remarkName; # 备注
Integer status; # 状态（1=关注；2=好友；0=陌生人）
String toNickname; # 好友昵称
int toUserId; # 好友 Id
int userId; # 用户 Id
//聊天记录过期时间 -1 为永久 数值 为天数
private double chatRecordTimeOut=-1;
```

~加关注 /friends/attention/add

① 请求参数：

参数名	参数类型	参数说明	是否必填
access_token	String	访问令牌	Y
toUserId	int	被关注用户 Id	Y

② 注：返回值 resultCode 说明

- 1=关注成功，已关注目标用户
- 2=关注成功，已互为好友
- 3=关注失败，已关注目标用户
- 4=关注失败，已互为好友
- 1100801=关注失败，目标用户拒绝关注
- 1100802=关注失败，目标用户将你拉黑

~批量添加关注 /friends/attention/batchAdd

① 请求参数:

参数名	参数类型	参数说明	是否必填
access_token	String	访问令牌	Y
toUserIds	String	拼接的被关注用户 Id (用逗号分隔)	Y

② 注: 返回值 resultCode 说明 (自动过滤非正常关系的好友关系 (拉黑), 只会添加没有好友关系的好友)

1=关注成功, 已关注目标用户

~拉黑 /friends/blacklist/add

请求参数:

参数名	参数类型	参数说明	是否必填
access_token	String	访问令牌	Y
toUserId	int	目标用户 Id	Y

注: 无返回数据

~添加好友 /friends/add

请求参数:

参数名	参数类型	参数说明	是否必填
access_token	String	访问令牌	Y
toUserId	int	目标用户 Id	Y

注: 无返回数据

~修改好友属性 /friends/update

请求参数:

参数名	参数类型	参数说明	是否必填
access_token	String	访问令牌	Y
toUserId	int	目标用户 Id	Y
chatRecordTimeOut	double	好友聊天记录删除时间 1 天=1.0	N

注: 无返回数据

~取消拉黑 /friends/blacklist/delete

请求参数:

参数名	参数类型	参数说明	是否必填
-----	------	------	------

access_token	String	访问令牌	Y
toUserId	int	目标用户 Id	Y

注：无返回数据

~取消关注 /friends/attention/delete

请求参数：

参数名	参数类型	参数说明	是否必填
access_token	String	访问令牌	Y
toUserId	int	被关注用户 Id	Y

注：无返回数据

~删除好友 /friends/delete

请求参数：

参数名	参数类型	参数说明	是否必填
access_token	String	访问令牌	Y
toUserId	int	目标用户 Id	Y

注：无返回数据

~好友备注 /friends/remark

请求参数：

参数名	参数类型	参数说明	是否必填
access_token	String	访问令牌	Y
toUserId	int	目标用户 Id	Y
remarkName	String	备注名	Y

注：无返回数据

~获取黑名单用户列表 /friends/blacklist

请求参数：

参数名	参数类型	参数说明	是否必填
access_token	String	访问令牌	Y

注：返回 List<Friends>

~获取粉丝列表 /friends/fans/list

请求参数：

参数名	参数类型	参数说明	是否必填
access_token	String	访问令牌	Y

userId	int	用户 Id	N
pageIndex	int	当前页码数	N
pageSize	int	每页数据条数	N

注：返回 List<Fans>

~获取关注列表 /friends/attention/list

请求参数：

参数名	参数类型	参数说明	是否必填
access_token	String	访问令牌	Y
userId	int	用户 Id	Y
status	int	状态值，默认值 0 （1=关注；2=好友；0=陌生人）	N

注：返回 List<Friends>

~获取好友详情 /friends/get

请求参数：

参数名	参数类型	参数说明	是否必填
access_token	String	访问令牌	Y
userId	int	用户 Id	Y
toUserId	int	目标用户 Id	Y

注：返回 Friends

~获取好友列表 /friends/list

请求参数：

参数名	参数类型	参数说明	是否必填
access_token	String	访问令牌	Y
userId	int	用户 Id	Y
keyword	String	关键词（用于搜索）	N

注：返回 List<Friends>

~查找好友 /friends/page

请求参数：

参数名	参数类型	参数说明	是否必填
access_token	String	访问令牌	Y
userId	int	用户 Id	Y
keyword	int	关键词	N

注：返回 List<Friends>

~获取好友的 userId 和单向关注的 userId 或黑名单的 userId /friends/friendsAndAttention

请求参数:

参数名	参数类型	参数说明	是否必填
access_token	String	访问令牌	Y
userId	int	用户 ID	Y
type	String	类型("friendList":好友和单向关注 "blackList":黑名单)	Y

注: 返回 List<Integer> : userId 集合

~获取新的朋友列表 /friends/newFriend/list

请求参数:

参数名	参数类型	默认值	参数说明	是否必填
access_token	String		访问令牌	Y
userId	int		用户 Id	Y
pageIndex	int	0	当前页码数	N
pageSize	int	10	每页数据条数	N

注: 返回 List<Friends>

~添加系统好友 /friends/addAllFriendsSys

请求参数:

参数名	参数类型	参数说明	是否必填
access_token	String	访问令牌	Y
toUserId	int	目标用户 Id, 默认值 10000	N

注: 无返回值

~好友消息免打扰, 阅后即焚, 聊天置顶 /friends/update/OfflineNoPushMsg

请求参数:

参数名	参数类型	参数说明	是否必填
access_token	String	访问令牌	Y
offlineNoPushMsg	int	是否开启消息免打扰, 阅后即焚, 聊天置顶, 0: 关闭, 1: 开启	Y
toUserId	int	目标用户 Id	Y
type	int	type = 0 消息免打扰, type = 1 阅后即焚, type = 2 聊天置顶	Y

请求返回示例:

```
{
  "currentTime": 1559788026732,
  "data": {
    "blacklist": 0,
    "chatRecordTimeOut": -1.0,
    "createTime": 1559119017,
    "fromAddType": 4,
    "isBeenBlack": 0,
    "isOpenSnapchat": 1,
    "isOpenTopChat": 0,
    "lastTalkTime": 0,
    "modifyTime": 1559300873,
    "msgNum": 0,
    "offlineNoPushMsg": 1,
    "openTopChatTime": 0,
    "status": 2,
    "toNickname": "测试帐号",
    "toUserId": 10000014,
    "toUserType": 0,
    "userId": 10000016
  }
}
```

```
"resultCode":1}
```

4.RoomController ---群组相关接口

群组实体 Room 属性说明

```
ObjectId id; # 房间编号
String jid; #群的 id,即 tigase 里的群组 Id
String name; #房间名称
String desc; #房间描述
String subject; 房间主题
Integer category; #房间分类
List<String> tags; #房间标签
String call; #语音通话标识符
Notice notice; #房间公告
List<Notice> notices; #公告列表
Integer userSize; #当前成员数
Integer maxSize = 1000; #最大成员数
Member member; #自己
List<Member> members; #成员列表
Integer countryId; #国家 Id
Integer provinceId; #省份 Id
Integer cityId; #城市 Id
Integer areaId; #地区 Id

Double longitude; #经度
Double latitude; #纬度
Integer userId; #创建者 Id
String nickname; #创建者昵称
Long createTime; #创建时间
Integer modifier; #修改人
Long modifyTime; #修改时间
Integer s; #状态
private Integer isLook=0;//是否可见 0 为可见 1 为不可见

//群主设置 群内消息是否发送已读 回执 显示数量
private int showRead=0;

private Integer isNeedVerify=0; // 加群是否需要通过验证 0: 不要 1: 要

private int showMember=1;//显示群成员给 普通用户 1 显示 0 不显示

private int allowSendCard=1;//允许发送名片 好友 1 允许 0 不允许
```

private int allowHostUpdate=1;//是否允许群主修改 群属性

private int allowInviteFriend=1;//允许普通成员邀请好友 默认 允许

private int allowUploadFile=1;//允许群成员上传群共享文件

private int allowConference=1;//允许成员 召开会议

private int allowSpeakCourse=1;//允许群成员 开启 讲课

// 大于当前时间时禁止发言

private long talkTime;

~创建群组 /room/add

/room/add?countryId=1&jid=7526ff27bce54d95bea9efd6be83862b&areaId=440307&access_token=62830e9d3b20457e94f
ce624a43a6087&longitude=114.066307&provinceId=440000&desc=yui&latitude=22.609084&name=yui&showRead=0&c
ityId=440300&

请求参数:

参数名	参数类型	参数说明	是否必填
access_token	String	访问令牌	Y
room	Room	群组实体	Y
text	String	要加入群组的用户 userId 集合 (json 字符串)	Y

注: 返回 Room

~删除群组 /room/delete

请求参数:

参数名	参数类型	参数说明	是否必填
access_token	String	访问令牌	Y
roomId	String	房间 Id	Y

注: 无返回数据

~更新群组 /room/update

请求参数:

参数名	参数类型	参数说明	是否必填
access_token	String	访问令牌	Y
roomId	String	房间 Id	Y
roomName	String	群名名称	N
notice	String	群组公告内容	N
desc	String	群组说明	N
showRead	Int	显示群消息已读人数 1 显示 0 关闭	N

isNeedVerify	int	加群是否需要验证 1 开启验证 0 关闭验证	N
isLook	int	群组是否公开 1 不公开 0 公开群组	N
allowSendCard	int	允许成员私聊 1 允许私聊 0 不允许私聊	N
chatRecordTimeOut	long	群组聊天记录删除时间 秒 值为到期时间	N
allowInviteFriend	int	允许成员 邀请好友 1 允许 0 禁止	N
allowUploadFile	int	允许成员上传 群共享文件 1 允许 0 禁止	N
allowConference	int	允许成员召开会议 1 允许 0 禁止	N
allowSpeakCourse	int	允许成员 开启讲课 1 允许 0 禁止	N
talkTime	long	全员禁言 long 到期时间	N

注：无返回数据

~群组转让 /room/transfer

请求参数：

参数名	参数类型	参数说明	是否必填
access_token	String	访问令牌	Y
roomId	String	群组 Id	Y
toUserId	int	新群主用户 ID 必须是群内成员	Y

注：返回 Room

~获取群组详情 /room/get

请求参数：

参数名	参数类型	参数说明	是否必填
access_token	String	访问令牌	Y
roomId	String	群组 Id	Y

注：返回 Room

~获取群组列表（按创建时间排序） /room/list

请求参数：

参数名	参数类型	参数说明	是否必填
access_token	String	访问令牌	Y
pageIndex	int	当前页码数	N
pageSize	int	每页数据条数	N
roomName	String	群名称（用于搜索）	N

注：返回 List<Room>

~添加群成员 /room/member/update

请求参数：

参数名	参数类型	参数说明	是否必填
-----	------	------	------

access_token	String	访问令牌	Y
roomId	String	房间 Id	Y
member	Member	成员对象（我自己）	N
text	String	要添加的成员用户 Id 集合（json 字符串）	N

注：无返回数据

~删除群成员 /room/member/delete

请求参数：

参数名	参数类型	参数说明	是否必填
access_token	String	访问令牌	Y
roomId	String	群组 Id	Y
userId	int	要删除的成员 Id	Y

注：无返回数据

~获取成员详情 /room/member/get

请求参数：

参数名	参数类型	参数说明	是否必填
access_token	String	访问令牌	Y
roomId	String	群组 Id	Y
userId	int	目标成员 userId	Y

注：返回 Room.Member

~获取成员列表 /room/member/list

请求参数：

参数名	参数类型	参数说明	是否必填
access_token	String	访问令牌	Y
roomId	String	群组 Id	Y
keyword	String	关键词（用于搜索）	N

注：返回 List<Room.Member>

~获取群组公告列表 /room/notice/list

请求参数：

参数名	参数类型	参数说明	是否必填
access_token	String	访问令牌	Y
roomId	String	群组 Id	Y

注：返回 List<Room.Notice>

~删除群组公告 /room/notice/delete

参数名	参数类型	参数说明	是否必填
access_token	String	访问令牌	Y
roomId	String	群组 Id	Y
noticeId	String	公告 ID	Y

~获取语音通话标识符 /room/get/call

请求参数:

参数名	参数类型	参数说明	是否必填
access_token	String	访问令牌	Y
roomId	String	群组 Id	Y

注: 返回数据说明: call 语音通话标识符

~加入群组 /room/join

请求参数:

参数名	参数类型	参数说明	是否必填
access_token	String	访问令牌	Y
roomId	String	房间 Id	Y
type	int	1=自己的房间; 2=加入的房间 (默认为 2)	N

注: 无返回数据

~离开群组 /room/leave

请求参数:

参数名	参数类型	参数说明	是否必填
access_token	String	访问令牌	Y
roomId	String	群组 Id	Y

注: 无返回数据

~用户历史房间列表 /room/list/his

请求参数:

参数名	参数类型	参数说明	是否必填
access_token	String	访问令牌	Y
type	int	0=所有; 1=自己的房间; 2=加入的房间 (默认为 0)	N
pageIndex	int	当前页码数	N
pageSize	int	每页数据条数	N

注：返回 List<Room>

~设置/取消 管理员 /room/set/admin

请求参数：

参数名	参数类型	参数说明	是否必填
access_token	String	访问令牌	Y
roomId	String	群组 id	Y
touserId	int	目标用户 id	Y
type	int	角色值 1=创建者、2=管理员、3=成员	Y

注：返回 List<Room>

~添加群共享 /room/add/share

请求参数：

参数名	参数类型	参数说明	是否必填
access_token	String	访问令牌	Y
roomId	String	群组 id	Y
type	int	类型	Y
size	long	文件大小	Y
userId	int	上传者 userId	Y
url	String	文件 url	Y
name	String	文件名称	Y

注：返回 Share 实体

~查询群共享 /room/share/find

请求参数：

参数名	参数类型	参数说明	是否必填
access_token	String	访问令牌	Y
roomId	String	群组 id	Y
time	long	时间（用于搜索）	N
pageIndex	int	当前页码数	N
pageSize	int	每页数据条数	N
userId	int	用户 id	N

注：返回 List<Share>

~获取群共享详情 /room/share/get

请求参数：

参数名	参数类型	参数说明	是否必填
access_token	String	访问令牌	Y
roomId	String	房间 Id	Y

shareId	String	共享文件 Id	Y
---------	--------	---------	---

注：返回 Share

~删除群共享 /room/share/delete

请求参数：

参数名	参数类型	参数说明	是否必填
access_token	String	访问令牌	Y
roomId	String	群组 Id	Y
shareId	int	共享文件 Id	Y
userId	int	用户 Id	Y

注：无返回数据

~群消息免打扰，置顶聊天 /room/member/setOfflineNoPushMsg

请求参数：

参数名	参数类型	参数说明	是否必填
access_token	String	访问令牌	Y
roomId	String	群组 Id	Y
offlineNoPushMsg	int	是否开启消息免打扰、聊天置顶 0:关闭，1: 开启	Y
type	int	0 :消息免打扰，1: 聊天置顶,	Y
userId	int	用户 Id	N

注：无返回数据

~群复制 /room/copyRoom

请求参数：

参数名	参数类型	参数说明	是否必填
access_token	String	访问令牌	Y
roomId	String	群组 Id	Y

请求返回示例：

```
{
  "currentTime": 1559789602637,
  "data": {
    "allowConference": 1,
    "allowHostUpdate": 1,
    "allowInviteFriend": 1,
    "allowSendCard": 1,
    "allowSpeakCourse": 0,
    "allowUploadFile": 1,
    "areaId": 440307,
    "call": "300136",
    "category": 0,
    "chatRecordTimeOut": -1.0,
    "cityId": 440300,
    "countryId": 1,
    "createTime": 1559789601,
    "desc": "hfd",
    "id": "5cf88021d85c8f108c6be2eb",
    "isAttritionNotice": 1,
    "isLook": 1,
    "isNeedVerify": 0,
    "jid": "f5e194e3258f41e08fbc2b28863d715b",
    "latitude": 22.610123,
    "longitude": 114.065744,
    "maxUserSize": 1000,
    "members": [],
    "modifyTime": 1559789601,
    "name": "xhdj",
    "nickname": "测试帐号",
    "notice": {
      "modifyTime": 0,
      "time": 0
    },
    "notices": [],
    "provinceId": 440000,
    "s": 1,
    "showMember": 1,
    "showRead": 0,
    "subject": "",
    "tags": [],
    "talkTime": 0,
    "userId": 10000014,
    "userSize": 0,
    "videoMeetingNo": "350136"
  },
  "resultCode": 1
}
```

5.NearbyController ---附近接口

~附近的用户 /nearby/user

请求参数:

参数名	参数类型	参数说明	是否必填
access_token	String	访问令牌	Y
longitude	double	经度	Y
latitude	double	纬度	Y
sex	int	性别	N
minAge	int	最小年龄	N
maxAge	int	最大年龄	N
active	long	最后出现时间（传时间对应的秒数，例如 30 分钟内则传 1800）	N
birthday	long	生日（毫秒数）	N
pageIndex	int	当前页码，默认 0	N
pageSize	int	每页数据页数，默认 10	N
nickname	String	用户昵称（用于搜索）	N
userId	Int	用户 ID 搜索	N

注：返回 List<User>

~最新的用户 /nearby/newUser

请求参数:

参数名	参数类型	参数说明	是否必填
access_token	String	访问令牌	Y
pageIndex	int	当前页码数	N
pageSize	int	每页数据条数	N

注：返回值为 List<User>

6.CompanyController --组织架构功能

~ 创建公司 /org/company/create

① 请求示例:

http://127.0.0.1:8092/org/company/create?companyName=MYCompany&createUserId=10007134&access_token=d4f44f6f4e564ee4b0438dd0e26e7ca9&

② 请求参数:

参数名	参数类型	参数说明	必填
-----	------	------	----

access_token	String	访问令牌	Y
companyName	String	公司名称	Y
createUserId	int	创建者 userId	Y

③ 返回值为 Company 实体 ,属性说明:

companyName #公司名
 createTime #创建时间
 createUserId #创建者 userId
 deleteTime #删除时间 默认值 0
 deleteUserId # 删除者 userId
 departments #该公司部门数据 (里面的数据为部门实体)
~~empNum~~ #~~该公司员工数量~~
 id #公司 id
 noticeContent #公告内容
 noticeTime #公告时间
 rootDpartId #根部门 Id
 type #类型值

④ 返回数据示例:

```

currentTime: 1504074866
data:
  companyName: "MY Company"
  createTime: 1504074865
  createUserId: 10007134
  deleteTime: 0
  deleteUserId: 0
  departments:
    0: Object
    1:
      companyId: "59a65c716a1f6c087c59e338"
      createTime: 1504074865
      createUserId: 10007134
      departName: "人事部"
      empNum: 1
      employees:
        0:
          companyId: "59a65c716a1f6c087c59e338"
          departmentId: "59a65c716a1f6c087c59e33a"
          id: "59a65c726a1f6c087c59e33c"
          nickname: "风度"
          position: "员工"
          role: 3
          userId: 10007134
          id: "59a65c716a1f6c087c59e33a"
          parentId: "59a65c716a1f6c087c59e339"
          type: 0
        2: Object
      empNum: 1
      id: "59a65c716a1f6c087c59e338"
      noticeContent: ""
      noticeTime: 0
      rootDpartId:
        0: "59a65c716a1f6c087c59e339"
        type: 0
      resultCode: 1
  
```

~ 根据 userId 查找其所属的公司 /org/company/getByUserId

① 请求参数:

参数名	参数类型	参数说明	必填
access_token	String	访问令牌	Y
userId	int	用户 Id	Y

② 返回值属性说明: 同上

③ 返回数据示例

```

currentTime: 1504427983
data:
  0:
    companyName: "深圳视酷科技有限公司"
    createTime: 1495888355
    createUserId: 10008121
    deleteTime: 0
    deleteUserId: 0
    departments: [7]
    empNum: 20
    id: "592971e3f914cb1b25976169"
    noticeContent: "欢迎使用，如有不足之处，可以反馈给我们，我们会及时修改，谢谢合作！"
    noticeTime: 1503883369
    rootDpartId:
      0: "592971e3f914cb1b2597616a"
      type: 5
    resultCode: 1|

```

~ 设置公司管理员 /org/company/setManager

① 请求参数:

参数名	参数类型	参数说明	必填
access_token	String	访问令牌	Y
companyId	String	公司 Id	Y
managerId	String	要设置为管理员的 userId 集合 (json 字符串)	Y

② 无返回值

~ 管理员列表 /org/company/managerList

① 请求参数:

参数名	参数类型	参数说明	必填
access_token	String	访问令牌	Y
companyId	String	公司 Id	Y

② 返回值属性说明:

companyId #公司 Id
departmentId #部门 Id
id #员工记录 Id
position #职位
role #角色值 3: 创建者 2: 管理员 1: 普通员工
userId #用户 Id

③ 返回数据示例

```
currentTime: 1504430065
data:
  0:
    companyId: "595318967760c93a9a58d9a2"
    departmentId: "595319107760c93a9a58d9aa"
    id: "595319207760c93a9a58d9ad"
    position: "员工"
    role: 2
    userId: 10009183
  1:
    companyId: "595318967760c93a9a58d9a2"
    departmentId: "595318967760c93a9a58d9a4"
    id: "59abc7c37760c962da5f021e"
    position: "员工"
    role: 2
    userId: 10007134
resultCode: 1
```

~ 修改公司名称、公告 /org/company/modify

① 请求参数:

参数名	参数类型	参数说明	必填
access_token	String	访问令牌	Y
companyId	String	公司 Id	Y
companyName	String	要修改的公司名称	N
noticeContent	String	要修改的公告内容	N

② 返回值属性说明:

companyName #公司名
createTime #创建时间
createUserId #创建者 userId
deleteTime #删除时间 默认值 0
deleteUserId # 删除者 userId
departments #该公司部门数据 (里面的数据为部门实体)
~~empNum~~ #该公司员工数量
id #公司 id

noticeContent #公告内容
noticeTime #公告时间
rootDpartId #根部门 Id
type #类型值

③ 返回数据示例

```
currentTime: 1504430708
data:
  companyName: "长麟集团"
  createTime: 1498618006
  createUserId: 10009188
  deleteTime: 0
  deleteUserId: 0
  empNum: 8
  id: "595318967760c93a9a58d9a2"
  noticeContent: "奖金已发，请速领取"
  noticeTime: 1504430708
  rootDpartId:
    0: "595318967760c93a9a58d9a3"
  type: 0
resultCode: 1
```

~ 删除公司 /org/company/delete (即：记录删除者 id,并将公司信息隐藏)

① 请求参数：

参数名	参数类型	参数说明	必填
access_token	String	访问令牌	Y
companyId	String	公司 Id	Y
userId	int	删除者 userId	Y

② 无返回值

~ 创建部门 /org/department/create

① 请求参数：

参数名	参数类型	参数说明	必填
access_token	String	访问令牌	Y
companyId	String	公司 Id	Y
parentId	String	父 Id (上一级部门的 Id, 根部门的父 id 为公司 Id)	Y
departName	String	部门名称	Y
createUserId	int	创建者 userId	Y

② 返回值属性说明:

companyId #公司 Id
createTime #创建时间
createUserId #创建者 userId
departName #部门名称
empNum #员工数量
id #该部门的 id
parentId #父节点 Id
type #类型值

③ 返回数据示例

```
currentTime: 1504432303
data:
  companyId: "595318967760c93a9a58d9a2"
  createTime: 1504432303
  createUserId: 10009188
  departName: "研发部"
  empNum: 0
  id: "59abd0af7760c962da5f022c"
  parentId: "595318967760c93a9a58d9a3"
  type: 0
resultCode: 1
```

~ 修改部门名称 /org/department/modify

① 请求参数:

参数名	参数类型	参数说明	必填
access_token	String	访问令牌	Y
departmentId	String	部门 Id	Y
dpartmentName	String	部门名称	Y

② 返回值属性说明:

companyId #公司 Id
createTime #创建时间
createUserId #创建者 userId
departName #部门名称
empNum #员工数量
id #该部门的 id
parentId #父节点 Id
type #类型值

③ 返回数据示例

```
currentTime: 1504440149
data:
  companyId: "595318967760c93a9a58d9a2"
  createTime: 1504439848
  createUserId: 10009188
  departName: "研发部1"
  empNum: 1
  id: "59abee287760c962da5f024a"
  parentId: "595318967760c93a9a58d9a3"
  type: 0
resultCode: 1|
```

~ 删除部门 /org/department/delete

① 请求参数:

参数名	参数类型	参数说明	必填
access_token	String	访问令牌	Y
departmentId	String	要删除的部门 Id	Y

② 无返回值

~ 添加员工 /org/employee/add

① 请求参数:

参数名	参数类型	参数说明	必填
access_token	String	访问令牌	Y
userId	String	要添加的用户 userId 集合 (json 字符串)	Y
companyId	String	公司 Id	Y
departmentId	String	部门 Id	Y
role	int	角色值 默认值:1 ,1:普通员工 2: 管理员 3: 创建者	Y

② 返回值属性说明:

- companyId #公司 Id
- departmentId #部门名称
- id #该员工的 id
- nickname #用户昵称
- position #职位名称
- role #角色值

userId #用户 Id

③ 返回数据示例

```
currentTime: 1504441405
data:
  0:
    companyId: "595318967760c93a9a58d9a2"
    departmentId: "59abee287760c962da5f024a"
    id: "59abee6f7760c962da5f024b"
    nickname: "acup20"
    position: "员工"
    role: 0
    userId: 10008121
  1:
    companyId: "595318967760c93a9a58d9a2"
    departmentId: "59abee287760c962da5f024a"
    id: "59abf43d7760c962da5f0252"
    nickname: "ghy"
    position: "员工"
    role: 0
    userId: 10008129
resultCode: 1
```

~ 删除员工 /org/employee/delete

① 请求参数:

参数名	参数类型	参数说明	必填
access_token	String	访问令牌	Y
userIds	String	要删除的用户 userId 集合 (json 字符串)	Y
departmentId	String	部门 id	Y

② 无返回值

~ 更改员工部门 /org/employee/modifyDpart

① 请求参数:

参数名	参数类型	参数说明	必填
access_token	String	访问令牌	Y
userId	int	要更改的员工的 userId	Y
companyId	String	公司 Id	Y
newDepartmentId	String	要更改的部门 Id	Y

② 返回值属性说明:

companyId #公司 Id
departmentId #部门名称
id #该员工的 id
nickname #用户昵称
position #职位名称

role #角色值
userId #用户 Id

③ 返回数据示例

```
currentTime: 1504443193
data:
  companyId: "595318967760c93a9a58d9a2"
  departmentId: "59abfab37760c962da5f025c"
  id: "59abee6f7760c962da5f024b"
  nickname: "acup20"
  position: "员工"
  role: 0
  userId: 10008121
resultCode: 1
```

~ 部门员工列表 /org/department/empList

① 请求参数:

参数名	参数类型	参数说明	必填
access_token	String	访问令牌	Y
departmentId	String	部门 Id	Y

② 返回值属性说明:

companyId #公司 Id
departmentId #部门名称
id #该员工的 id
nickname #用户昵称
position #职位名称
role #角色值
userId #用户 Id

③ 返回数据示例

```

    currentTime:      1504443675
  data:
    0:
      companyId:      "595318967760c93a9a58d9a2"
      departmentId:   "59abee287760c962da5f024a"
      id:              "59abf43d7760c962da5f0252"
      nickname:        "ghy"
      position:        "员工"
      role:            0
      userId:          10008129
    1:
      companyId:      "595318967760c93a9a58d9a2"
      departmentId:   "59abee287760c962da5f024a"
      id:              "59abfd127760c962da5f0260"
      nickname:        "TEST02"
      position:        "员工"
      role:            0
      userId:          10009180
    2:                Object
    3:                Object
  resultCode:         1|

```

~ 公司员工列表 /org/company/employees

① 请求参数:

参数名	参数类型	参数说明	必填
access_token	String	访问令牌	Y
companyId	String	公司 Id	Y

② 返回值属性说明:

同上

③ 返回数据示例

```

currentTime:      1504443675
data:
  0:
    companyId:      "595318967760c93a9a58d9a2"
    departmentId:    "59abee287760c962da5f024a"
    id:              "59abf43d7760c962da5f0252"
    nickname:        "ghy"
    position:        "员工"
    role:            0
    userId:          10008129
  1:
    companyId:      "595318967760c93a9a58d9a2"
    departmentId:    "59abee287760c962da5f024a"
    id:              "59abfd127760c962da5f0260"
    nickname:        "TEST02"
    position:        "员工"
    role:            0
    userId:          10009180
  2:                Object
  3:                Object
resultCode:        1

```

~ 更改员工角色 /org/employee/modifyRole

① 请求参数:

参数名	参数类型	参数说明	必填
access_token	String	访问令牌	Y
userId	int	用户 Id	Y
companyId	String	公司 Id	Y
role	int	角色值	Y

② 返回值属性说明:

companyId #公司 Id
 departmentId #部门名称
 id #该员工的 id
 nickname #用户昵称
 position #职位名称
 role #角色值

userId #用户 Id

③ 返回数据示例

```

currentTime:      1504443193
data:
  companyId:       "595318967760c93a9a58d9a2"
  departmentId:    "59abfab37760c962da5f025c"
  id:              "59abee6f7760c962da5f024b"
  nickname:        "acup20"
  position:        "员工"
  role:            0
  userId:          10008121
resultCode:        1

```

~ 更改员工职位(头衔) /org/employee/modifyPosition

① 请求参数:

参数名	参数类型	参数说明	必填
access_token	String	访问令牌	Y
userId	int	用户 Id	Y
companyId	String	公司 Id	Y
position	String	职位名称	Y

② 返回值属性说明: 返回值为员工实体

说明同上

③ 返回数据示例

```

currentTime:      1504443193
data:
  companyId:       "595318967760c93a9a58d9a2"
  departmentId:    "59abfab37760c962da5f025c"
  id:              "59abee6f7760c962da5f024b"
  nickname:        "acup20"
  position:        "员工"
  role:            0
  userId:          10008121
resultCode:        1

```

~ 公司列表 /org/company/list

① 请求参数:

参数名	参数类型	默认值	参数说明	必填
access_token	String		访问令牌	Y

pageIndex	int	0	当前页码数	N
pageSize	int	30	每页数据条数	N

② 返回值属性说明:

companyName #公司名
 createTime #创建时间
 createUserId #创建者 userId
 deleteTime #删除时间 默认值 0
 deleteUserId # 删除者 userId
 departments #该公司部门数据（里面的数据为部门实体）
~~empNum~~ #~~该公司员工数量~~
 id #公司 id
 noticeContent #公告内容
 noticeTime #公告时间
 rootDpartId #根部门 Id
 type #类型值

③ 返回数据示例

```

currentTime:      1504445818
data:
  0:
    companyName:   "SHIKU"
    createTime:    1496199016
    createUserId:  10007789
    deleteTime:   0
    deleteUserId: 0
    empNum:        1
    id:            "592e2f68f914cb1b25976627"
    noticeContent: ""
    noticeTime:    1496199016
    rootDpartId:
      0:           "592e2f68f914cb1b25976628"
      type:        0
  1: Object
  2: Object
  3: Object
  4: Object
resultCode:       1
  
```

~ 部门列表 /org/department/list

① 请求参数:

参数名	参数类型	参数说明	必填
access_token	String	访问令牌	Y

② 返回值属性说明:

companyld #公司 ld

createTime #创建时间
createUserId #创建者 userId
departName #部门名称
employees #该部门的员工数据（封装员工实体）
empNum #员工数量
id #该部门的 id
parentId #父节点 Id
type #类型值

③ 返回数据示例

```
currentTime: 1504446011
data:
  0: Object
  1:
    companyId: "595318967760c93a9a58d9a2"
    createTime: 1498618006
    createUserId: 10009188
    departName: "人事部"
    empNum: 4
    employees: [4]
    id: "595318967760c93a9a58d9a4"
    parentId: "595318967760c93a9a58d9a3"
    type: 0
  2: Object
  3: Object
  4: Object
  5: Object
  6: Object
  7: Object
  8: Object
resultCode: 1
```

~ 获取公司详情 /org/company/get

① 请求参数:

参数名	参数类型	参数说明	必填
access_token	String	访问令牌	Y
companyId	String	公司 Id	Y

② 返回值为公司 CompanyVO 实体

~ 获取员工详情 /org/employee/get

① 请求参数:

参数名	参数类型	参数说明	必填
access_token	String	访问令牌	Y
employeeId	String	员工 Id	Y

② 返回值为员工 Employee 实体

~ 获取部门详情 /org/department/get

① 请求参数:

参数名	参数类型	参数说明	必填
access_token	String	访问令牌	Y
departmentId	String	部门 Id	Y

② 返回值为员工 DepartmentVO 实体

~ 员工退出公司 /org/company/quit

① 请求参数:

参数名	参数类型	参数说明	必填
access_token	String	访问令牌	Y
companyId	String	公司 Id	Y
userId	int	用户 Id	Y

② 无返回值

~ 获取公司中某个员工角色值 /org/employee/role

① 请求参数:

参数名	参数类型	参数说明	必填
access_token	String	访问令牌	Y
companyId	String	公司 Id	Y
userId	int	员工的 userId	Y

② 返回值属性说明:

3: 创建者 2: 管理员 1: 普通员工

③ 返回数据示例

```
currentTime: 1504447773
data: 3
resultCode: 1
```

7.TigaseController ---tigase 支持接口

消息类型说明:

- type = 1, //文本
- type = 2, //图片
- type = 3, //语音
- type=4, //位置
- type=5, //动画
- type=6, //视频
- type=7, //音频
- type=8, //名片
- type=9, //文件
- type=10, //提醒

获取首页的最近一条的聊天记录列表/tigase/getLastChatList

参数名	参数类型	参数说明	是否必填
access_token	String	访问令牌	Y
startTime	long	开始时间 即账号下线时间（单位：秒）	N
endTime	long	结束时间（单位：秒）	N

~单聊聊天记录 /tigase/shiku_msgs

请求参数:

参数名	参数类型	参数说明	是否必填
access_token	String	访问令牌	Y
receiver	int	接收者 userId	Y
startTime	long	开始时间（单位：秒）	N
endTime	long	结束时间（单位：秒）	N
pageIndex	int	页码（默认：0）	N

pageSize	int	页大小（默认：50）	N
----------	-----	------------	---

调用成功：

```
{
  "data": [
    {
      "_id": "5534d49e806f18d9acbab50c",
      "body": "消息记录 1",
      "direction": 0,
      "message": "",
      "receiver": 10000018,
      "receiver_jid": "101@127.0.0.1",
      "sender": 10000019,
      "sender_jid": "100@127.0.0.1",
      "ts": 1429525662702,
      "type": 1
    },
    {
      "_id": "5534d4d4806f18d9acbab50f",
      "body": "消息记录 1 回复",
      "direction": 1,
      "message": "",
      "receiver": 10000018,
      "receiver_jid": "101@127.0.0.1",
      "sender": 10000019,
      "sender_jid": "100@127.0.0.1",
      "ts": 1429525716531,
      "type": 1
    }
  ],
  "resultCode": 1,
  "resultMsg": ""
}
```

说明：body=消息文本内容；message=原始消息；
direction=1（receiver=发送方，sender=接收方）
direction=0（receiver=接收方；sender=发送方）

~群聊聊天记录 /tigase/shiku_muc_msgs

请求参数：

参数名	参数类型	参数说明	是否必填
access_token	String	访问令牌	Y
roomId	String	群组 Id	Y
startTime	long	开始时间（单位：秒）	N
endTime	long	结束时间（单位：秒）	N

pageIndex	int	当前页码数（默认：0）	N
pageSize	int	每页数据条数（默认：20）	N

调用成功：

```
{
  "data": [
    {
      "_id": "5534d556806f18d9acbab510",
      "body": "啦啦啦 1",
      "event_type": 1,
      "message": "",
      "nickname": "",
      "public_event": 0,
      "room_jid_id": "abc",
      "room_jid": "abc@muc.127.0.0.1",
      "sender_jid": "101@127.0.0.1",
      "sender": 101,
      "ts": 1429525846437
    },
    {
      "_id": "5534d559806f18d9acbab511",
      "body": "啦啦啦 2",
      "event_type": 1,
      "message": "",
      "nickname": "",
      "public_event": 0,
      "room_jid_id": "abc",
      "room_jid": "abc@muc.127.0.0.1",
      "sender_jid": "100@127.0.0.1",
      "sender": 100,
      "ts": 1429525849605
    }
  ],
  "resultCode": 1,
  "resultMsg": ""
}
```

~离线推送 /tigase/notify

请求参数：

参数名	参数类型	参数说明	是否必填
access_token	String	访问令牌	Y
from	long	发送方 userId	Y
to	long	接收方 Id	Y
body	String	消息体	Y

ts	long	当前时间（默认：0）	N
----	------	------------	---

注：无返回值

~百度推送 /tigase/baiduPush

请求参数：

参数名	参数类型	参数说明	是否必填
access_token	String	访问令牌	Y
from	int	发送方 userId	Y
to	int	接收方 userId	Y
type	int	消息类型（默认值 1）	N

注：无返回值

~消息推送 /tigase/push

请求参数：

参数名	参数类型	参数说明	是否必填
access_token	String	访问令牌	Y
text	String	消息接收者的 userId 集合（json 字符串）	Y
body	String	消息体	Y

注：无返回值

~在线状态 /tigase/OnlineState

请求参数：

参数名	参数类型	参数说明	是否必填
access_token	String	访问令牌	Y
userId	int	用户 Id	Y
OnlineState	int	在线状态 0：离线 1：在线	Y

注：调用成功：

~消息加密 /tigase/encrypt

请求参数：

参数名	参数类型	参数说明	是否必填
access_token	String	访问令牌	Y
text	String	要加密的消息内容	Y
key	String	加密所用的 key	Y

注：返回值说明： text：加密后的内容

~消息解密 /tigase/decrypt

请求参数:

参数名	参数类型	参数说明	是否必填
access_token	String	访问令牌	Y
text	String	要解密的消息内容	Y
key	String	解密所用的 key	Y

注: 返回值说明: text: 解密后的内容

~获取消息接口 /tigase/getMessage

请求参数:

参数名	参数类型	参数说明	是否必填
access_token	String	访问令牌	Y
type	int	聊天类型 (默认 1) 1 单聊 2 群聊	N
messageId	String	消息 Id	Y

返回值属性说明:

_id #记录 Id
body #消息体
direction #类别值 0 :发出的消息 1: 接收到的消息
message #消息 xml 未解析前的格式 (原始消息)
receiver #接收者 Id
receiver_jid #接收者 jid
sender #发送者 id
sender_jid #发送者 jid
ts #消息发送/接收 时间
type 聊天类型: 1 单聊 2 群聊
contextType : 消息类型
messageId #消息 Id
context #消息内容

返回数据示例

```

currentTime:      1504583585
▼ data:
  _id:             "5971c7b05b927a720cf49383"
  ▼ body:
    "{\content\":"肥肥肥 v 发反反复复","\fromUserName\":"test06","\isEncrypt
    \":false,\isRead\:true,\isReadDel\:false,\messageId\:
    \"c09d3191496049748a32c3af01bf7662\",\timeSend\:1500628912.567,\type\:1}"
    direction:     0
  ▼ message:
    "<message xmlns=\jabber:client\" id=\c09d3191496049748a32c3af01bf7662\"
    to=\10009182@www.shiku.co\" from=\10009188@www.shiku.co/youjob\" type=\chat
    \"><request xmlns=\urn:xmpp:receipts\"/><body>{\isEncrypt\:false,\type\:1,
    \timeSend\:1500628912.567,\isRead\:false,\content\":"肥肥肥 v 发反反复复",
    \fromUserName\":"test06\",\messageId\:\"c09d3191496049748a32c3af01bf7662\",
    \isReadDel\:false}</body></message>"
  receiver:        10009182
  receiver_jid:    "10009182@www.shiku.co"
  sender:          10009188
  sender_jid:      "10009188@www.shiku.co"
  ts:              1500628912567
  type:            1
  contextType:     1
  messageId:       "c09d3191496049748a32c3af01bf7662"
  context:         "肥肥肥 v 发反反复复"
  resultCode:      1

```

~删除消息接口 /tigase/deleteMsg

请求参数:

参数名	参数类型	参数说明	是否必填
access_token	String	访问令牌	Y
type	int	聊天类型 1 单聊 2 群聊	Y
delete	int	delete 1: 删除属于自己的消息记录 2: 撤回 删除整条消息记录	Y
messageId	String	要删除的消息 Id	Y

注: 返回值说明: text: 加密后的内容

8. MpController ---公众号

~获取菜单列表 /public/menu/list

① 请求参数:

参数名	参数类型	参数说明	是否必填
access_token	String	访问令牌	Y
userId	int	用户 Id	Y

② 返回值属性说明:

id: 菜单 Id

name: 菜单名

parentId: 父菜单 Id
url: 菜单链接
menuList: 子菜单集合

③ 请求成功后返回值示例:

```
{
  "data": [
    {
      "id": 681385389583237120,
      "menuList": [
        {
          "id": 681391821095436288,
          "name": "江山如此多娇",
          "parentId": 681385389583237120,
          "url": "www.js.com",
          "userId": 10000104
        },
        {
          "id": 681391914045407232,
          "name": "江山如此多娇 2",
          "parentId": 681385389583237120,
          "url": "www.js2.com",
          "userId": 10000104
        }
      ],
      "name": "泥人 2003",
      "parentId": 0,
      "url": "http://weibo.com/ajaxlogin.php?framelogin=1&callback=parent.sinaSSOController.feedBackUrlCallBack",
      "userId": 10000104
    },
    {
      "id": 681391987286343680,
      "menuList": [
        {
          "id": 681392166903218176,
          "name": "第三个菜单子菜单 1",
          "parentId": 681391987286343680,
          "url": "1111111111.cn",
          "userId": 10000104
        }
      ],
      "name": "第三个菜单",
      "parentId": 0,
      "url": "www.baidu.com",
      "userId": 10000104
    }
  ]
}
```

```

    },
    {
        "id": 681392063853363200,
        "menuList": [
            {
                "id": 681392267222581248,
                "name": "充值活动",
                "parentId": 681392063853363200,
                "url": "cz.com",
                "userId": 10000104
            }
        ],
        "name": "四个菜单",
        "parentId": 0,
        "url": "123.com",
        "userId": 10000104
    }
],
"resultCode": 1
}

```

9.AdminMpController ---公众号后台页面相关

此模块返回值为 **jsp** 页面

~ 删除粉丝 /mp/fans/delete

① 请求参数：

参数名	参数类型	参数说明	是否必填
access_token	String	访问令牌	Y
toUserId	int	要删除的用户的 userId	Y

~ 公众号登录 /mp/login

注：/mp/login 接口有两个，其中 GET 接口是跳转到公众号登录界面，POST 为登录接口，参数如下：

① 请求参数：

参数名	参数类型	参数说明	是否必填
access_token	String	访问令牌	Y

username	String	用户名	Y
password	String	密码	Y

~ 公众号菜单操作 /mp/menu/{op}

① 请求参数：

参数名	参数类型	参数说明	是否必填
access_token	String	访问令牌	Y
op	String	选项 save：保存菜单 delete：删除菜单	Y
entity	Menu	菜单实体（封装菜单相关数据）	Y

~ 粉丝列表 /mp/fans

① 请求参数：

参数名	参数类型	默认值	参数说明	是否必填
access_token	String		访问令牌	Y
pageIndex	int	0	当前页码数	N
pageSize	int	10	每页数据条数	N

~ 公众号主页 /mp/home

① 请求参数：

参数名	参数类型	参数说明	是否必填
access_token	String	访问令牌	Y

~公众号页面菜单列表 /mp/menuList

① 请求参数：

参数名	参数类型	参数说明	是否必填
access_token	String	访问令牌	Y

~消息管理 /mp/msg

① 请求参数：

参数名	参数类型	默认值	参数说明	是否必填
access_token	String		访问令牌	Y
pageIndex	int	0	当前页码数	N
pageSize	int	15	每页数据条数	N

~ 消息列表 /mp/msg/list

① 请求参数：

参数名	参数类型	默认值	参数说明	是否必填
access_token	String		访问令牌	Y
toUserId	int		要查看消息的用户 UserId	Y
pageIndex	int	0	当前页码数	N
pageSize	int	15	每页数据条数	N

~ 消息回复 /mp/msg/reply

① 请求参数：

参数名	参数类型	参数说明	是否必填
access_token	String	访问令牌	Y
toUserId	int	回复消息者 userId	N

~ 群发单条图文 /mp/push

① 请求参数：

参数名	参数类型	参数说明	是否必填
access_token	String	访问令牌	Y

~ 群发多条图文 /mp/many

① 请求参数：

参数名	参数类型	参数说明	是否必填
access_token	String	访问令牌	Y

~群发消息 /mp/textToAll

① 请求参数：

参数名	参数类型	参数说明	是否必填
title	String	群发内容	Y

~ 消息发送 /mp/msg/send

① 请求参数：

参数名	参数类型	参数说明	是否必填
access_token	String	访问令牌	Y
toUserId	int	消息接受者的用户 userId	Y
body	String	消息体	Y

~单条图文推送 /mp/pushToAll

① 请求参数：

参数名	参数类型	参数说明	是否必填
access_token	String	访问令牌	Y
title	String	标题	Y
sub	String	内容	Y
img	String	图片地址	Y
url	String	url	Y

~多条图文推送 /mp/manyToAll

① 请求参数：

参数名	参数类型	参数说明	是否必填
access_token	String	访问令牌	Y
title	String[]	标题集合	Y
url	String[]	url 集合	Y
img	String[]	Img 地址集合	Y

10. AdminController ----IM 后台管理

~单聊聊天记录 /console/chat_logs_all

① 请求参数：

参数名	参数类型	默认值	参数说明	是否必填
access_token	String		访问令牌	Y
startTime	long	0	开始时间（毫秒数）	N
endTime	long	0	结束时间（毫秒数）	N
sender	int	0	发送者 Id（用于搜索）	N
receiver	int	0	接收者 Id（用于搜索）	N
pageIndex	int	0	当前页面	N
pageSize	int	10	每页数据条数	N

② 返回值为一个 jsp 页面

~后台删除消息记录 /console/chat_logs_all/del

① 请求参数：

参数名	参数类型	默认值	参数说明	是否必填
access_token	String		访问令牌	Y
startTime	long	0	开始时间（毫秒数）	N
endTime	long	0	结束时间（毫秒数）	N
sender	int	0	发送者 Id（用于搜索）	N
receiver	int	0	接收者 Id（用于搜索）	N
pageIndex	int	0	当前页面	N
pageSize	int	25	每页数据条数	N

~后台删除房间 /console/deleteRoom

① 请求参数：

参数名	参数类型	参数说明	是否必填
access_token	String	访问令牌	Y
roomId	String	群组 Id	Y
pageIndex	int	当前页码	Y

~后台页面删除房间成员 /console/deleteMember

① 请求参数：

参数名	参数类型	参数说明	是否必填
access_token	String	访问令牌	Y
roomId	String	群组 id	Y
userId	int	用户 id	Y
pageIndex	int	当前页码数	Y

~后台页面删除用户 /console/deleteUser

① 请求参数：

参数名	参数类型	参数说明	是否必填
access_token	String	访问令牌	Y
userId	int	用户 Id	Y
nickName	String	用户昵称（用于搜索）	N

~后台群组聊天记录 /console/groupchat_logs_all

① 请求参数：

参数名	参数类型	默认值	参数说明	是否必填
access_token	String		访问令牌	Y
startTime	long	0	开始时间（毫秒数）	N
endTime	long	0	结束时间（毫秒数）	N
room_jid_id	String	“ ”	群组 jid（tigase 里的群组 Id）用于搜索	N
pageIndex	int	0	当前页码	N
pageSize	int	25	每页数据条数	N

~后台删除群组聊天记录 /console/groupchat_logs_all/del

① 请求参数：

参数名	参数类型	默认值	参数说明	是否必填
access_token	String		访问令牌	Y
startTime	long	0	开始时间（毫秒数）	N
endTime	long	0	结束时间（毫秒数）	N

room_jid_id	String	“ ”	群组 jid (tigase 里的群组 Id) 用于搜索	N
pageIndex	int	0	当前页码	N
pageSize	int	25	每页数据条数	N

~后台登录 /console/login

注: /console/login 接口有两个 GET 和 POST 其中 GET 为跳转到登录界面 POST 为后台登录接口

① 请求参数:

参数名	参数类型	参数说明	是否必填
access_token	String	访问令牌	Y
telephone	String	电话号码	Y
password	String	密码	Y

~后台退出登录接口 /console/logout

① 请求参数:

参数名	参数类型	参数说明	是否必填
access_token	String	访问令牌	Y

~后台群发消息 /console/pushToAll

① 请求参数:

参数名	参数类型	参数说明	是否必填
access_token	String	访问令牌	Y
fromUserId	int	发送者 userId	Y
body	String	消息体	Y

~后台页面获取群组列表 /console/roomList

① 请求参数:

参数名	参数类型	默认值	参数说明	是否必填
access_token	String		访问令牌	Y
roomName	String	“ ”	群组名称 (用于搜索)	N
pageIndex	int	0	当前页码数	N
pageSize	int	50	每页数据条数	N

~单个群组聊天记录详情 /console/roomMsgDetail

① 请求参数：

参数名	参数类型	默认值	参数说明	是否必填
access_token	String		访问令牌	Y
pageIndex	int	0	当前页码数	N
pageSize	int	50	每页数据条数	N
room_jid_id	String	“ ”	群组 jid（tigase 里的群组 Id）	Y

~后台页面获取用户列表 /console/userList

① 请求参数：

参数名	参数类型	默认值	参数说明	是否必填
access_token	String		访问令牌	Y
pageIndex	int	0	当前页码数	N
pageSize	int	50	每页数据条数	N
onlinestate	int	0	用户在线状态：离线 0 在线 1	N
nickname	String	“ ”	用户昵称	N

~后台页面更新用户信息 /console/updateUser

注： /console/updateUser 接口有两个 GET 和 POST GET 是获取用户数据 参数：access_token userId，
POST 是用户数据更新接口，参数如下：

① 请求参数：

参数名	参数类型	参数说明	是否必填
access_token	String	访问令牌	Y
userId	int	用户 Id	Y
example	UserExample	UserExample 实体（封装需要更新的用户数据）	Y

~获取用户在线状态 /console/userStatus

注：此接口用于后台页面用户在线走势图

① 请求参数：

参数名	参数类型	参数说明	是否必填
access_token	String	访问令牌	Y

~后台创建群组 /console/addRoom

注：/console/addRoom 有两个接口，GET 为跳转到添加群主界面，POST 为创建群组接口，参数如下：

① 请求参数：

参数名	参数类型	参数说明	是否必填
access_token	String	访问令牌	Y
room	Room	群组实体（封装创建群组的数据）	Y
ids	String	需要加入群主的用户 id 集合(json 字符串)	N

~群组成员管理 /console/roomUserManager

① 请求参数：

参数名	参数类型	默认值	参数说明	是否必填
access_token	String		访问令牌	Y
pageIndex	int	0	当前页码数	N
pageSize	int	10	每页数据条数	N
id	String		群组 id	Y

~发送消息 /console/sendMessage

① 请求参数：

参数名	参数类型	参数说明	是否必填
access_token	String	访问令牌	Y
body	String	消息体	Y
from	int	发送者 UserId	Y
to	int	接收者 UserId	Y
count	int		Y

~后台添加用户 /console/useradd

① 请求参数：

参数名	参数类型	参数说明	是否必填
access_token	String	访问令牌	Y
telephone	String	电话号码	Y
nickname	String	昵称	Y
password	String	密码	Y

~初始化系统用户 /console/initSysUser

注：10000 号和 10005 号为系统用户，可在清除数据库后调用此方法初始化系统用户。

① 请求参数：

参数名	参数类型	默认值	参数说明	是否必填
access_token	String		访问令牌	Y
userId	int	0	用户 Id	N

11 AddressBookController 通讯录好友

参数名	参数类型	参数说明	必填
chatNum	int	当前会话的人数	
companyId	String	公司 Id	

12 文件 处理

复制文件 upload/copyFile

参数名	参数类型	参数说明	必填
paths	string	要复制的文件	Y
validTime	int	复制的文件 保存有效期 -1=永久 1= 一天 最小一天	Y

返回结果

```
{
  "data": {
    "url": "http://file.shiku.co/u/8295/10008295/201808/o/f83bd291738946ae90ea441e8e8170d0.jpg"
  },
  "resultCode": 1
}
```

13 AuthKeysController 用户授权相关接口

获取用户支付私钥 /authkeys/getPayPrivateKey

参数名	参数类型	参数说明	是否必填
access_token	String	访问令牌	Y

上传用户支付公私钥 /authkeys/uploadPayKey

参数名	参数类型	参数说明	是否必填
access_token	String	访问令牌	Y
publicKey	String	公钥	Y
privateKey	String	私钥	Y
mac	String	Mac 验签	Y

上传用户消息公私钥 /authkeys/uploadMsgKey

参数名	参数类型	参数说明	是否必填
access_token	String	访问令牌	Y
rsaPublicKey	String	rsa 公钥	Y
rsaPrivateKey	String	rsa 私钥	Y
dhPublicKey	String	dh 公钥	
dhPrivateKey	String	dh 私钥	
mac	String	Mac 验签	Y

上传用户 DH 消息公私钥 /authkeys/uploadDhMsgKey

参数名	参数类型	参数说明	是否必填
access_token	String	访问令牌	Y
publicKey	String	公钥	Y
privateKey	String	私钥	Y
mac	String	Mac 验签	Y

上传用户 RSA 消息公私钥 /authkeys/uploadRsaMsgKey

参数名	参数类型	参数说明	是否必填
access_token	String	访问令牌	Y
publicKey	String	公钥	Y
privateKey	String	私钥	Y
mac	String	Mac 验签	Y

获取用户 DH 公钥列表 /authkeys/getDHMsgKeyList

参数名	参数类型	参数说明	是否必填
access_token	String	访问令牌	Y
userId	int	好友用户 Id 不填 默认返回自己的	Y