# 微信开放平台文档

socialinzen微信开放平台: 首页验证用户是否登录,若登录过,并且本地缓存了accesstoken验证凭证,则直接跳过登录,进入第三方授权判断,如发现用户授权过,则跳过授权阶段,直接进入后台控制台。使用微信rest api形式进行请求,必须附带url query。格式为xxx?access\_token=xxxxxxx accesstoken是该平台验证用户的唯一凭证,无此凭证除用户登录和网页授权外都不能使用。

## 平台接口

## 用户行为接口:

#### 1.用户登录接口

接口路径	/apis/user/login
请求方法	post Content-Type:application/x-www-form-urlencoded
请求参数	json: {username:用户名,password:密码}
回复结果	json&cookies json:{errcode:错误编码,msg:错误信息} cookies: SocialinzenToken———用户accessToken SocialinzenUser————用户id
附	SocialinzenToken是用户登录的唯一凭据,也是访问数据时校验权限必须携带的,使 用url query 方式传递

#### 2.用户注册接口

接口路径	/apis/user/register
请求方法	post Content-Type:application/x-www-form-urlencoded
请求参数	json: {username:用户名,password:密码,email:用户邮箱}
回复结果	json:{errcode:错误编码,msg:错误信息}
附	注册账号成功后会发送一封邮件到用户注册邮箱,验证邮件后方可登录

#### 3.用户登出

接口路径	/apis/user/logout
请求方法	post Content-Type:application/x-www-form-urlencoded
请求参数	无
回复结果	json:{errcode:错误编码,msg:错误信息}
附	登出成功会摧毁accesstoken和缓存cookies凭证,重定向到主页。

## 微信回复管理接口:

## 1.增加回复规则:

接口路径	/apis/wxreplymanage/addReply
请求方法	post Content-Type:application/x-www-form-urlencoded
请求参数	json: { type:回复类型(text,image,video,music,multitext,voice), reply:回复信息结构体(根据不同类型格式不同,见附) request:信息请求关键词 }
回复结果	json:{errcode:错误编码,msg:错误信息}
附	回复信息格式: text: {type: 'text',content: 回复信息主体} image:{type: 'image',content: {mediald: 媒体id(使用素材接口上传)} vedio:{type: 'vedio',content: {title:标题,description: 描述,mediald: 素材id}} music:{title: 音乐标题,description: 描述(可选),musicUrl:音乐文件url,thumbMediald: 缩略图媒体id} multitext:{title:标题,description:描述,picurl:图片url(通过文件上传接口获得),url:点击后跳转url} voice:{type: 'voice', content: {mediald:素材id}} 请求时附带access_token(url query方式)

### 2.删除回复规则

接口路径	/apis/wxreplymanage/deleteReply
请求方法	post Content-Type:application/x-www-form-urlencoded
请求参数	json:{request:请求关键词}
回复结果	json:{errcode:错误编码,msg:错误信息}
附寸	请求时附带access_token(url query方式)

## 3.列出回复规则

接口路径	/apis/wxreplymanage/listReply
请求方法	post Content-Type:application/x-www-form-urlencoded
请求参数	无

```
回复结果
                array of json 回复规则数组
                e.g:
                [{
                type:回复类型(text,image,video,music,multitext,voice),
                reply:回复信息结构体(根据不同类型格式不同,见附)
                request:信息请求关键词
                type:回复类型(text,image,video,music,multitext,voice),
                reply:回复信息结构体(根据不同类型格式不同,见附)
                request:信息请求关键词
                }, {
                type:回复类型(text,image,video,music,multitext,voice),
                reply:回复信息结构体(根据不同类型格式不同,见附)
                request:信息请求关键词
附
                请求时附带access_token(url query方式)
```

## REST API 微信接口:

#### 微信网页授权:

1成后两贝技权:		
接口路径	/wxplatform/webauth	
请求方法	get	
请求参数	appid:公众号appid scope:获取授权方式(见附)	
回复结果	json:{openid:用户唯一标识id 用户详细信息(scope为snsapi_userinfo时)如: "nickname": NICKNAME, "sex":"1", "province":"PROVINCE" "city":"CITY", "country":"COUNTRY", "headimgurl": "http://wx.qlogo.cn/mmopen/ g3MonUZtNHkdmzicIlibx6iaFqAc56vxLSUfpb6n5WKSYVY0ChQKkiaJSgQ1 dZuTOgvLLrhJbERQQ 4eMsv84eavHiaiceqxibJxCfHe/46", "privilege":[ "PRIVILEGE1" "PRIVILEGE2" ], "unionid": "o6_bmasdasdsad6_2sgVt7hMZOPfL" }	
附	以snsapi_base为scope发起的网页授权,是用来获取进入页面的用户的openid的,并且是静默授权并自动跳转到回调页的。用户感知的就是直接进入了回调页(往往是业务页面)以snsapi_userinfo为scope发起的网页授权,是用来获取用户的基本信息的。但这种授权需要用户手动同意,并且由于用户同意过,所以无须关注,就可在授权后获取该用户的基本信息。	

## 微信网页jssdk签名获取

接口路径	/apis/wxplatform/jsticket
请求方法	post Content-Type:application/x-www-form-urlencoded
请求参数	{appid:使用jssdk的公众号appid,url:使用jssdk的页面url}
回复结果	json:{     "appid": "wxd222ec9daccce0b9",     "nonceStr": "noStrAASOc1#@@=",hash salt     "timestamp": 1482214703,时间戳     "signature": "133f1315da046d12b59ad36a3b4cd82d54f5c0ca"签名 }

## 微信二维码生成:

接口路径	/apis/wxplatform/qrcode
请求方法	post Content-Type:application/x-www-form-urlencoded
请求参数	{expire(二维码过期时间,可选,发送即生成临时二维码),senseld:场景id(临时二维码为小与32位的正整数,永久二维码可选1-10000的正整数或者64位以下字符串))}
回复结果	json:{qrcodeUrl:二维码图片链接}
附	请求时附带access_token(url query方式)

## 微信关注者管理:

## 更新关注者备注:

接口路径	/apis/wxplatform/updateRemark
请求方法	post Content-Type:application/x-www-form-urlencoded
请求参数	json:{openid:关注者的openid,remark:备注信息}
回复结果	json:{errcode:错误编码,msg:错误信息}
附寸	请求时附带access_token(url query方式)

#### 获取分组列表

接口路径	/apis/wxplatform/getGroups
请求方法	post Content-Type:application/x-www-form-urlencoded
请求参数	无

```
json:{
回复结果
                     "groups": [
                       "id": 0,
                       "name": "未分组",
                       "count": 5
                      },
                       "id": 1,
                       "name": "黑名单",
                       "count": 1
                      "id": 2,
                       "name": "星标组",
                       "count": 0
                      },
                       "id": 100,
                      "name": "测试分组",
                       "count": 0
                     }
                    ]
                   请求时附带access_token(url query方式)
附
```

#### 获取当前关注者分组id

接口路径	/apis/wxplatform/getWhichGroup
请求方法	post Content-Type:application/x-www-form-urlencoded
请求参数	{openid:关注者openid}
回复结果	json:{   "groupid": 0(若为0 则代表该用户为未分组用户,非0整数则为分组id,id为1组为黑名单组) }
附	请求时附带access_token(url query方式)

#### 新建分组

接口路径	/apis/wxplatform/createGroup
请求方法	post Content-Type:application/x-www-form-urlencoded
请求参数	{groupName:分组名称}
回复结果	json:{errcode:错误编码,msg:错误信息}
附	请求时附带access_token(url query方式)

#### 更新分组信息

接口路径	/apis/wxplatform/updateGroup
请求方法	post Content-Type:application/x-www-form-urlencoded
请求参数	{groupName:分组名称}

回复结果	json:{errcode:错误编码,msg:错误信息}
附	请求时附带access_token(url query方式)
将用移动至分组	

接口路径	/apis/wxplatform/moveUserToGroup
请求方法	post Content-Type:application/x-www-form-urlencoded
请求参数	{groupId:分组Id,openid:关注者openid}
回复结果	json:{errcode:错误编码,msg:错误信息}
附	请求时附带access_token(url query方式)

## 获取当前分组用户

接口路径	/apis/wxplatform/moveUserToGroup
请求方法	post Content-Type:application/x-www-form-urlencoded
请求参数	{groupId:分组Id}
回复结果	json Array:[ {     "appid": "5850a5b9d74dfbb25dad8907",     "city": "吉林",     "country": "中国",     "groupid": "0",     "language": "zh_CN",     "nickname": "Bryan
附	请求时附带access_token(url query方式)

## 素材管理

## 上传素材

接口路径	/apis/materials/uploadMaterial
请求方法	post Content-Type:multipart/form-data

请求参数	{type:素材种类 (image-图片素材2m限制 / .png .jpg .jpeg voice-语音素材2m限制 / .mp3 .wma .wav .amr thumb-缩略图素材64kb / .jpg) , file:素材文件 }
回复结果	json:{errcode:错误编码,msg:错误信息}
附	请求时附带access_token(url query方式)
上传视频素材	
接口路径	/apis/materials/uploadVideoMaterial
请求方法	post Content-Type:multipart/form-data
请求参数	{ file:素材文件 mp4格式, tile:视频标题 desc:视频描述 }
回复结果	json:{errcode:错误编码,msg:错误信息}
附	请求时附带access_token(url query方式)
删除素材	
接口路径	/apis/materials/removeMaterial
请求方法	post Content-Type:multipart/form-data
请求参数	{ mediald:素材id }
回复结果	json:{errcode:错误编码,msg:错误信息}
附	请求时附带access_token(url query方式)
获取素材使用情况	
接口路径	/apis/materials/getMaterialCount
请求方法	post Content-Type:multipart/form-data
请求参数	无
回复结果	json:{   "voice_count": 0,   "video_count": 3,   "image_count": 3,   "news_count": 0 }
附	请求时附带access_token(url query方式)

## 通过类型获取素材列表

接口路径	/apis/materials/getMaterialByType
请求方法	post Content-Type:multipart/form-data

## 上传logo

接口路径	/apis/materials/uploadLogo
请求方法	post Content-Type:multipart/form-data
请求参数	{logolFile:logo文件}
回复结果	json:{errcode:错误编码,msg:错误信息}
附	请求时附带access_token(url query方式)