

微信服务接口文档

V1.2

目录

微信服务接口文档	1
1.文档修订记录	3
2.文档说明	4
3.接口说明	4
3.1 消息发送	4
3.1.1 接口	4
3.1.2 消息发送组装实体	5
3.1.3 消息发送 Content 组装 Demo	6
3.2 永久二维码	8
3.2.1 含义及作用	8
3.2.2 接口类型	8
3.2.3 调用方法	8
3.3 临时二维码 http 接口	8
3.3.1 含义及作用	8
3.3.2 接口类型	8
3.3.3 接口	8
3.3 扫描二维码消息总线接口	
3.3.1 接口	9
3.3.2 消息体	10
3.4 微信二维码服务接口	10
3.4.1 接口	10
3.5 微信传图接口	12
3.5.1 接口类型	12
3.5.2 接口	12
3.6 微信绑定服务接口	14
3.6.1 接口	14
3.7 微信信息查询服务接口	
3.7.1 接口	17
3.8 微信红包接口	19
3.8.1 规则	19
3.8.2 接口	20
3.8.3 调用 Demo	21
3.9 微信 JS 接口	23
3.10 微信授权接口	24
3.10.1 含义及作用	24
3.10.2 接口	25
4.相关配置	27
4.1Maven 依赖	27
4.2scf.config 配置	27
5.资源分配	28

1.文档修订记录

版本号	修订人	修订内容	备注
V1.0	刘晓龙	创建文档	
V1.1	董鹏军	天天微信传图、微信	
		绑定、微信二维码、	
		微信信息查询、二维	
		码扫描通知等接口	
V1.1	刘晓龙	修复配置文件、更新	
		接口	
V1.1	刘晓龙	红包接口	
V1.1	李腾	微信 JS 接口	
V1.2	刘晓龙	增加微信授权接口	

2.文档说明

微信服务(SNS)是与58微信服务号交互唯一入口。功能包括向已关注者发送信息、获取微信号二维码、记录用户绑定关系、管理微信配置等。

3.接口说明

3.1 消息发送

发送消息使用的是 scf 服务接口, maven 依赖和 scf 配置依据本文档的相关配置里的内容。

3.1.1 接口

```
public interface IWXSendService {
    IWXSendService PROXY = ProxyFactory.create(IWXSendService.class,
"tcp://sns/WXSendServiceImpl");

    /**
    * 发送微信信息
    *
        * @param systemId 发送方的系统ID,为空直接返回异常,需要提前申请
        * @param userId 接收信息的58用户ID 为空直接返回异常
        * @param content 消息内容 组装方法详见: {@link

com.bj58.vip.sns.entity.WXMessage}
        * @throws Exception
        */
        @OperationContract
        public void send(String systemId, Long userId, String content)

throws Exception;
```

```
/**
   * 发送微信信息
   * @param systemId 发送方的系统ID,为空直接返回异常,需要提前申请
   * @param opened 接收信息的微信id,可以通过TUMC用58id查询
   * @param content 消息内容 组装方法详见: {@link
com.bj58.vip.sns.entity.WXMessage}
   * @throws Exception
   */
   @OperationContract
   public void sendToOpenid(String systemId, String openid, String
content) throws Exception;
}
3.1.2 消息发送组装实体
public class WXMessage {
   /**
   * 组装 纯文本消息
   * @return
   */
   public static WeixinTextMsg textMsg() {
      return new WeixinTextMsg();
   }
   /**
   * 组装 图文消息
   * @return
   public static WeixinGraphicMsg graphicMsg() {
      return new WeixinGraphicMsg();
   }
   /**
   * 代金券
   * @return
   public static DaiJinQuan daiJinQuan(){
     return new DaiJinQuan();
   }
```

3.1.3 消息发送 Content 组装 Demo

● 纯文本

代码:

```
public void testSendTxt() throws Exception{
    //文本消息仅支持超级链接(<a>)和换行(\n)
    String content = WXMessage.textMsg().text("你好\n<a
href=\"http://www.58.com\">点击</a>").finish();
}
```

发送后效果:



● 多图文

代码:

```
public void testSendGraph() throws Exception{
    //支持多图文,每个图文一个articles
    String content = WXMessage.graphicMsg()
        .articles("第一个图文", "第一个图文描述",
"http://img.58cdn.com.cn/ui6/index/logo.gif",
"http://img.58cdn.com.cn/ui6/index/logo.gif")
        .articles("第二个图文", "第二个图文描述",
"http://img.58cdn.com.cn/ui6/index/logo.gif",
"http://img.58cdn.com.cn/ui6/index/logo.gif")
        .finish();
}
```

发送后效果:



● 模板消息

代码:

```
public void testSendTemplate() throws Exception{
    //不同模板对应字段不同,详见各个模板的代码注释
    String content = WXMessage.daiJinQuan().title("这里是标题").coupon("58").expDate("2099年12月31日").remark("这里是备注").finish();
    }
发送后效果:
```

获得代金券通知
12月31日
这里是标题
代金券金额:58
有效期:2099年12月31日
这里是备注

3.2 永久二维码

3.2.1 含义及作用

微信永久二维码生成后将一直生效,可用于线下推广或固定入口。 按照现有规定,永久二维码有数量限制,仅10万个。

3.2.2 接口类型

永久二维码仅支持 scf 调用。

3.2.3 调用方法

永久二维码申请,需要和微信产品@于雪楠沟通后,方可分配。具体调用方法,详见后面章节:微信二维码服务接口。

3.3 临时二维码 http 接口

3.3.1 含义及作用

临时二维码生成后 30 分钟即失效,用户扫描可以根据业务将当前场景的信息传递到服务器,可用于用户定向推送信息。

3.3.2 接口类型

http 请求, 使用 get 请求

3.3.3 接口

请求 url: http://weixin.58.com/sns/qrcode/{source}?scenestr=***

请求支持 jsonp

参数: source 区分调用方 1~31, scenestr 场景值

Demo: http://weixin.58.com/sns/grcode/1?scenestr=12345678

结果: ({"imgurl":["https://mp.weixin.qq.com/cgi-bin/showqr code?ticket=gQFW8ToAAAAAAAAAAAAAAXxodHRw0i8vd2VpeGluLnFxLmNvbS9 xL0VuVXhNRXZtQVNSNDVKaVZNRnZZAAIEucmjVAMECAcAAA%3D%3D"]}) 结果中的 imgurl 就是生成的二维码的 url

注意: 二维码有效期为 30 分钟

3.3 扫描二维码消息总线接口

在用户扫描二维码后,我们会以消息总线,方便业务端根据各自 场景进行业务处理。

3.3.1 接口

esb 主题: 18301

esb 测试地址:

tcp://esb1. service. test. 58dns. org:30000, esb2. service. tes
t. 58dns. org:30000?clientID=1

esb 生产地址:

tcp://esb1.service.58dns.org:30000, esb2.service.58dns.or

3.3.2 消息体

消息体:

{openid":"oxQkTt1U8GVbJMWv3r0SYbztD0hg", "scancode":"0", "source":0, "srcvalue":"0", "trigger":"subscribe"}

消息体为为标准 json 串,由于订阅后会收到所有的 esb 消息,所以业务方需要筛选自己需要的消息

openid 为关注用户的微信 id

scancode 为通过二维码扫描到的值,临时二维码的值使计算得来的,这里我们将 source 和 srcvalue 依据分解出来了

source 为业务方标示 接收到 esb 消息后需要业务方筛选一次

srcvalue 为业务方生产二维码的原始值 , 即传入的 scenestr 值

trigger 为产生消息的来源 subscribe 代表新关注, scan 是已 关注用户再次扫描二维码,接收到 esb 消息后需要业务方筛选一次

3.4 微信二维码服务接口

3.4.1 接口

public interface QrcodeImgService {

```
QrcodeImgService proxy =
ProxyFactory.create(QrcodeImgService.class,
         "tcp://sns/QrcodeImgServiceImpl");
   /**
   * 生成临时二维码图片 默认有效期20分钟
   * @param source 区分调用方 1~31
   * @param scene_id 场景id 1~99999999
   * @return 图片的url
   */
   @OperationContract
   public String qrCodeTemp(int source, int scene_id);
   /**
   * 生成临时二维码图片 * 默认有效期20分钟
   * @param source 区分调用方 1~31
   * @param scene_Str 场景值
   * @return 图片的url
   */
   @OperationContract
   public String qrCodeTemp(int source, String scene Str);
   /**
   * 生成临时二维码图片 *
   * @param source 区分调用方 1~31
* @param scene_id 场景id 1~99999999
* @param seconds 有效期,最大1800毫秒
   * @return 图片的url
   */
   @OperationContract
   public String qrCodeTemp(int source, int scene_id, int
seconds);
   /**
   * 生成永久二维码图片
   * @param scene id 1~100000
   * @return 图片的url
   */
   @OperationContract
   public String qrCodeForever(long scene_id);
   /**
   * 通过扫描值,得到原始值 *
   * @param code 十进制扫描值
```

```
* @return
*/
@OperationContract
public String getSceneStr(long code);
}
```

3.5 微信传图接口

微信传图通过微信用户扫描二维码后,上传图片完成。

3.5.1 接口类型

http 请求, 使用 get 方式或者 post 方式

3.5.2 接口

1. 二维码调用

weixin.58.com/sns/upimg/show?callback=*&uptype=*返回数据 callback({url:http://****, scenestr:****})
Url 为待扫描二维码

Scenestr 为本次调用标示,后续接口调用需要传递此参数 二维码 30 分钟失效,超时前台要重新调用接口获取

2. 收取微信用户上传的图片

weixin. 58. com/sns/upimg/list?scenestr=***&callback=***
成功上传后返回数据 callback({url:[***,***,***]})
超时返回数据 call({code:1,msg:servicetimeout}) 超时时间: 30

此接口前台不需要控制超时,后台会自动断开连接,前台接收超时 code=1 后重连即可

3. 上传达到图片上线后提示

weixin. 58. com/sns/upimg/alert?scenestr=***&callback=***
返回消息 callback({code:0:msg:ok})

用户收到的消息:图片上传数量已达到上限,如需继续上传,请在电脑上删除之前上传的图片

以下收取图片的代码样例可作为参考:

```
var queryWeixinUploadImages = function() {
        $. ajax('http://weixin.58.com/sns/upimg/list?scenestr='+securityCode, {
                 type: 'get',
dataType: 'jsonp',
                 jsonp: 'callback',
                 success: function(data) {
                         var json = eval(data);
                         //console.log('QUERY SUCCESS');
                         //console.log(json);
                         if (json.url && json.url.length > 0) {
                                  cachedImages = json.url;
                                  shiftCacheImages();
                         queryWeixinUploadImages();
                 },
                 error: function(e) {
                         //console.log('>>>>QUERY ERROR');
                         //console.log(e);
                         queryWeixinUploadImages();
                 }
        });
}
```

```
3.6 微信绑定服务接口
3.6.1 接口
3.6.1.1 查询绑定关系
@ServiceContract
public interface IThirdUserService {
  IThirdUserService Proxy =
ProxyFactory.create(IThirdUserService.class,
"tcp://sns/ThirdUserServiceImpl");
  /**
   * 根据userId查询
   * @param type 1、微信; 2、手Q
   * @param userId
   * @return 如果用户没有绑定则返回null
   * @throws Exception
   */
  @OperationContract
  public ThirdUser getByUserId(Integer type, Long
userId) throws Exception;
```

}

3.6.1.2 绑定与解绑操作

```
/**
* 微信绑定相关业务
* @author liteng
@ServiceContract
public interface IBindService {
  IBindService PROXY =
ProxyFactory.create(IBindService.class,
"tcp://sns/BindServiceImpl");
  /**
   * Bind user and (weixin or shoug)
   * @param numberType shoug or weixin @link
com.bj58.vip.sns.enums.NumberType
   * @param uid 58账号id
   * @param opened 微信id标识符
   * @param password 58微信账号密码,这个请联系相关人员获取
   * @return
   * 0 非首次绑定
   * 1 两者都是首次绑定
   * 2 58用户首次绑定
   * 3 微信用户首次绑定
   * -1 openid已被绑定
   * -2 58id已被绑定
   * -3 tumc绑定失败
   * @throws Exception
   */
  @OperationContract
  public int bind(NumberType numberType, long uid,
String openid, String password) throws Exception;
```

```
/**
   * Bind user and (weixin or shoug), from QRCode scanning
   * @param numberType shoug or weixin @link
com.bj58.vip.sns.enums.NumberType
   * @param uid 58账号uid
   * @param openid 微信账号id
   * @param scanCode 扫描的二维码编号
   * @param password 密码 58微信账号密码,这个请联系相关人员获取
   * @return
   * 0 非首次绑定
   * 1 两者都是首次绑定
   * 2 58用户首次绑定
   * 3 微信用户首次绑定
   * -1 openid已被绑定
   * -2 58id已被绑定
   * -3 tumc绑定失败
   * @throws Exception
   */
  @OperationContract
  public int bind(NumberType numberType, long uid,
String openid, String scanCode, String password) throws
Exception;
   * Unbind user and (weixin or shoug) by uid
   * @param numberType shouq or weixin @link
com.bj58.vip.sns.enums.NumberType
   * @param uid
   * @param password 58微信账号密码,这个请联系相关人员获取
   * @return
   * @throws Exception
  @OperationContract
  public void unBindByUid(NumberType numberType, long
uid, String password) throws Exception;
```

```
/**
   * Unbind user and (weixin or shoug) by openid @link
com.bj58.vip.sns.enums.NumberType
   * @param numberType shouq or weixin
   * @param openid
   * @param password 58微信账号密码,这个请联系相关人员获取
   * @return
   * @throws Exception
  @OperationContract
  public void unBindByOpenId(NumberType numberType, S
tring openid, String password) throws Exception;
}
3.7 微信信息查询服务接口
3.7.1 接口
/**
* 微信信息相关业务
* @author liteng
*/
@ServiceContract
public interface IWeixinUserService {
   IWeixinUserService PROXY =
ProxyFactory.create(IWeixinUserService.class,
"tcp://sns/WeixinUserServiceImpl");
```

```
* 根据58用户id获取微信用户基本信息(58用户必须绑定微信,否则返回
null)
   * @param uid
   * @return
   * @throws Exception
  @OperationContract
  public WeixinUser getWeixinUserBy58Uid(long uid)
throws Exception;
  /**
   * 根据微信openid获取微信用户基本信息
   * @param openId
   * @return
   * @throws Exception
  @OperationContract
  public WeixinUser getWeixinUserByOpenId(String
openId) throws Exception;
/**
   * 根据58用户id获取微信用户基本信息(58用户必须绑定微信,否则返回
null)
   * @param uid
   * @param type shouq or weixin @link com.bj58.vip.sns.enum
                  s.NumberType
   * @return
   * @throws Exception
   */
  @OperationContract
```

```
public WeixinUser getWeixinUserBy58Uid(long uid,
NumberType type) throws Exception;
   * 根据微信openid获取微信用户基本信息
   * @param openId
   * @param type shouq or weixin @link com.bj58.vip.sns.enum
                  s.NumberType
   * @return
   * @throws Exception
  @OperationContract
   public WeixinUser getWeixinUserByOpenId(String
openId, NumberType type) throws Exception;
}
3.8 微信红包接口
3.8.1 规则
```

- 微信红包需要向产品@王东升@孙骥备案
- 微信红包需要提前向财务借款
- 微信内部策略:
 - ◆ 红包接收者可以不关注 58 微信号, 只需要获取到 0penid 即可
 - ◆ 同一用户每日最多可获取红包数: 2 个
 - ◆ 同一用户每月最多可获取红包数: 10 个

- ◆ 单个红包金额必须在 1²00 (包含 1 和 200) 元之间
- ◆ 红包金额最小单位是分
- ◆ 在 0:00-8:00 时间段内, 无法发送红包
- ◆ 系统每分钟发送红包数不能超过50个
- ◆ 每天发送红包总数不能超过 10000 个

3.8.2 接口

```
public interface IWXRedEnvelopeService {
  IWXRedEnvelopeService PROXY = ProxyFactory
        .create(IWXRedEnvelopeService.class,
"tcp://sns/WXRedEnvelopeServiceImpl");
  /**
   * 给58用户发送红包,58用户必须绑定微信
   * @param sysId
             业务系统ID
   * @param uid
             58用户ID
   * @param money
             红包金额 必须大于0
   * @param title
             红包标题
   * @param desc
             红包描述
   * @throws Exception
   */
  @OperationContract
  public void sendRedEnvelopeByUid(String sysId, long
```

```
uid, double money, String title, String desc) throws
Exception;
```

```
/**
   * 给微信用户发送红包
   * @param sysId
              业务系统ID
   * @param openId
              微信ID
   * @param money
              红包金额 必须大于0
   * @param title
              红包标题
   * @param desc
              红包描述
   * @throws Exception
  @OperationContract
  public void sendRedEnvelopeByOpenId(String sysId,
String openId, double money, String title, String desc)
        throws Exception;
   }
```

3.8.3 调用 Demo

```
代码:

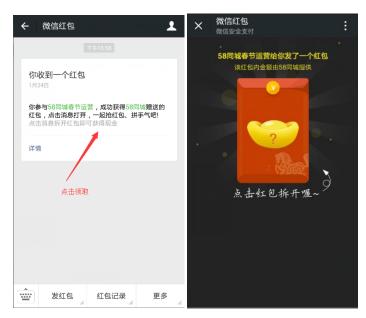
@Test

public void testSendRedEnvelope() throws Exception {
    // 接收者openid 必须是58微信公众号对应的openid 标志: 以oH开头
    String receiveOpenid = "oH***";
    // 红包标题(最长32个字节)
    String title = "58同城春节运营";
    // 红包具体描述(最长128个字节)
    String desc = "58同城祝您春节快乐!";
```

```
// 红包金额 >=1 & <= 200 最小单位是分
    double money = 1.11;
// 分配的系统id 这里仅仅是展示使用 不可以直接使用
"defalutSystemId"
    String systemId = "defalutSystemId";
// send</pre>
```

IWXRedEnvelopeService.PROXY.sendRedEnvelopeByOpenId(systemI
d, receiveOpenid, money, title, desc);
}

用户收到效果:





3.9 微信 JS 接口

3.9.1 准备工作

- 1、需要使用微信 JS 的页面必须位于 58.com 域名下, 否则将无法授权使用微信 JS。
- 2、引入 JS 文件

在需要调用 JS 接口的页面引入如下 JS 文件:

<script src=" http://j1.58cdn.com.cn/m58/m3/js/weixin/wxjs-0.0.3.js "></script>

3、调用微信 JS-API

由于微信对 JS-SDK 的授权是异步进行的,所以对 JS-API 的调用需要在授权成功后才可以进行调用。异步授权成功后,会执行 wxjs.ready 方法,所有对 JS-API 的调用都应该放在 wxjs.ready 方法中。

```
wxjs.ready(function(api){
```

//所有对微信 jsapi 的调用需要在此执行,使用回调函数的 api 参数进行调用 });

3.9.2 设置分享信息

同时设置所有分享信息(分享到朋友圈、分享给朋友、分享到 QQ、分享到腾讯微博):

```
api.share.set({
```

如果只需要设置某一种分享的信息,可以通过 api.share.set 函数指定分享类型参数,例如只设置分享到朋友圈的信息:

```
api.share.set("timeline", {
```

//标题、描述等信息

});

分享类型参数共有4种,分别为:

- 1. 分享到朋友圈: timeline
- 2. 分享给朋友: appMessage
- 3. 分享到 QQ: qq
- 4. 分享到腾讯微博: weibo

3.9.3 图像接口

3.9.3.1 图片预览

3.9.3.2 拍照或从手机相册中选图

3.9.4 音频接口

3.10 微信授权接口

3.10.1 含义及作用

微信授权接口用于第三方页面获取当前微信用户信息。通过单纯 获取 openid 或者同时获取昵称(用户需要手动点击授权)的方式, 进行后续业务处理

3.10.2 接口

● 引导用户点击链接:

http://testweixin.test.58v5.cn/wxoauth/request?path=http://bj.58.com&oauthid=123&scope=snsapi base

线下地址: http://testweixin.test.58v5.cn/wxoauth/request

线上地址: http://weixin. 58. com/wxoauth/request

其中:

path 为用户授权后的回调第三方链接地址,必须是 58 域名 oauthid 为分配的授权 ID。第三方业务线使用该接口时,需要先申请该 ID,同时会分配 RSA 私钥用户后续数据的解密

scope 为授权级别, snsapi_base 为基础授权, 只能获取到 openid 信息。snsapi_userinfo 为全部信息授权, 可以取到 openid 及微信头像等的信息。snsapi_base 对用户透明, snsapi_userinfo 需要用户操作(用户授权一次后 24 不需要重新授权)

● 用户授权后, 跳转至 path 的 url, 后面加 info=****, 数据需要用私钥解密

加密数据样例

b53545d28d79371f122eafb980ab42fc71e66a7cb6a836e03e4c142304f30516463b6cd864a7dd033 6c5f7b6e4c7eff939fb1b1dfdbed57c53fac75b8682ec0d8d3406fbae8f49b8d27116fd38249d70de57 17fe4bb7546d86c632e537c7ef87591379f4b1ba1ba0e691d683f8a3b76117e80afd5057e2871f4136 61c0510b4041a74f5efe3333821a6ef938a2ea8f6faf060d4dce924469ab4b9d357d8a3f757402284b7 64a33a56d176801737cf57147f4f9ea62385fe610dcf4f0a28eda85f3067488ad077283746dee9f1483 5f756ff6dc1b9eac8311ca2aeedf28a7d2661d1cbfeaad53219683fc0754c77c1534e7fa46e6a874760 ced72e2bfabc199665d3e72832585309103e567b33b1f2b6143c624e2b7c3967147c47ef02fd70796f ae33043c894bedb0cd8768e0d67d9746ed7fe409a96f62b35fa509e563e55f1f3eeaf898778fb7e706a

 $2340ef5656a8f3e911e9f23b44e2d418c5c24af842d1a5b7236223353d13ade8c78042b05cf3a6dced\\ 25ce36e3bf883296dbcf94e5a9c4b1637a23120fe8cd086528397093faf94188e97ae16778c0ee6c7b\\ 75a5a99fb8e9c12a0ab677c17a1f462588ad2443b497a00310e6f4150a67c8af85b29b4d0462feead9\\ 1d4edc1b0a578f8d7f4c94900b44814111732c6aff2392ac03f7e795fc50168ec6f121a3a31e1b43d2f\\ 80da64bfb2cb76302395ac3cb4394bc504735c6cb168f351436bb929d07008a0f4bae406215725aa7\\ e2d7da0facc6ecd7fcb9e99df12eccd86ea2baeb6a33d256716094953f3784650967f6d8002bc784faa\\ 0d2565da4d3201124dabdbbb191d9e0448c2117b2d31c9a95ff6a355bb354c498274b06fcc7d89a52\\ b19e4c260618ede913330f5225418d63ec6b8fd3e0b15c$

解密后

```
{"errcode":0,"errmsg":"ok","userinfo":{"city":" 海 淀 ","country":" 中 国 ","headimgurl":"http://wx.qlogo.cn/mmopen/ajNVdqHZLLCF0oibvxRFIto3vj6rw6lv9rwnrdc0wgH W9pzmeUfoxpx3XAKUHJ97zZISSMFLdAMaPzpuRPriaWoA/0","nickname":" 一 斤 三 毛 ","openid":"opCQLtzyokpkJ54_n1bypgTl3ezl","params":{},"privilege":[],"province":" 北 京 ","sex":1}}
```

解密需要依赖 jar 包:

com. bj58. biz. utility. RsaDecryptUtil 类的 decrypt 方法可以解密,需要调用 init 方法初始化私钥。

● 代码

```
static{
    RsaDecryptUtil.init("privatekey");
}

public static void main(String[] args) {
    String rtn = RsaDecryptUtil.decrypt("");
    System.out.println(rtn);
}
```

4.相关配置

4.1Maven 依赖

```
<dependency>
     <groupId>com.bj58.vip</groupId>
     <artifactId>com.bj58.vip.sns.contract</artifactId>
     <version>1.1.9-SNAPSHOT</version>
</dependency>
```

4.2scf.config 配置

```
<Service name="sns" id="5" maxThreadCount="1000">
    <Communication>
       <SocketPool protected="true" bufferSize="4096" minPoolSize="2"</p>
maxPoolSize="5" nagle="true" autoShrink="00:00:20" sendTimeout="00:00:02"
receiveTimeout="00:00:03" waitTimeout="00:00:02"/>
       <Protocol serialize="SCF" encoder="UTF-8"</pre>
compressType="UnCompress"/>
   </Communication>
    <Loadbalance>
        <Server deadTimeout="00:00:10">
            <!-- 测试环境 -->
            <!-- <add name="sns1" host="192.168.120.2" port="17810"
maxCurrentUser="1000"/> -->
            <add name="sns1" host="sns1.service.58dns.org" port="17810"
maxCurrentUser="1000"/>
           <add name="sns2" host="sns2.service.58dns.org" port="17810"
maxCurrentUser="1000"/>
            <add name="sns3" host="sns3.service.58dns.org" port="17810"
maxCurrentUser="1000"/>
            <add name="sns4" host="sns4.service.58dns.org" port="17810"
maxCurrentUser="1000"/>
           <add name="sns5" host="sns5.service.58dns.org" port="17810"
maxCurrentUser="1000"/>
            <add name="sns6" host="sns6.service.58dns.org" port="17810"
maxCurrentUser="1000"/>
            <add name="sns7" host="sns7.service.58dns.org" port="17810"
maxCurrentUser="1000"/>
            <add name="sns8" host="sns8.service.58dns.org" port="17810"
maxCurrentUser="1000"/>
```

5.资源分配

接口中包含 source、systemid 等用于标识调用微信平台的各个业务方,方便数据监控及后续统计分析。当新业务方接入微信平台,或原有业务方扩展业务需要新微信接口时,需要联系技术人员@张亚通(zhangyatong)或@刘晓龙(liuxiaolong)进行备案并分配。