**企业微信**

**API文档**

腾讯科技（深圳）有限公司

| 文档版本号 | 适用于产品版本号 | 修改人 | 审核人 | 文档修订时间 | 文档审核时间 | 修改点 |
| --- | --- | --- | --- | --- | --- | --- |
| V1 |  |  |  |  |  | 新建 |
| V1.1 |  | V\_xqinchen | sanhuang | 2017.11.30 | 2017.11.30 | 更新文档模板 |
| V1.2 | V1.2.2 | V\_xqinchen | sanhuang | 2018.2.1 | 2018.2.1 | 1.新增2.2.8用户踢下线；  2.新增第十章至第十二章。 |
| V1.3 | V1.3 | V\_xqinchen | sanhuang | 2018.4.13 | 2018.4.13 | 1、合并应用管理与应用管理助手  2、补充权限说明、版本信息  3、删除自定义菜单章节 |

目录

1 开发前必读 1

1.1 概述 1

1.1.1 术语介绍 1

1.1.2 开始 1

2 通讯录同步 4

2.1 概述 4

2.2 成员管理 5

2.2.1 创建成员 5

2.2.2 读取成员 8

2.2.3 更新成员 11

2.2.4 删除成员 14

2.2.5 批量删除成员 15

2.2.6 获取部门成员 16

2.2.7 获取部门成员详情 18

2.2.8 用户踢下线 21

2.3 部门管理 23

2.3.1 创建部门 23

2.3.2 更新部门 25

2.3.3 删除部门 26

2.3.4 获取部门列表 27

2.4 标签管理 29

2.4.1 创建标签 29

2.4.2 更新标签名字 30

2.4.3 删除标签 32

2.4.4 获取标签成员 33

2.4.5 增加标签成员 34

2.4.6 删除标签成员 36

2.4.7 获取标签列表 38

3 应用管理与应用管理助手 39

3.1 概述 39

3.2 应用管理 42

3.2.1 创建应用 42

3.2.2 获取应用 45

3.2.3 设置应用 48

3.2.4 删除应用 51

3.2.5 启动应用 53

3.2.6 停用应用 54

3.2.7 重置应用secret 55

3.2.8 开启应用回调 56

3.2.9 更新应用回调 58

3.2.10 停用应用回调 59

3.2.11 创建应用菜单 60

3.2.12 获取应用菜单 67

3.2.13 删除应用菜单 68

3.2.14 获取应用列表 69

3.3 分组管理 73

3.3.1 创建应用分组 73

3.3.2 更新应用分组 75

3.3.3 删除应用分组 76

3.3.4 增加分组应用 77

3.3.5 删除分组应用 78

3.3.6 获取指定分组的应用列表 80

3.3.7 获取分组列表 82

3.4 设置工作台显示方式 84

4 消息推送 86

4.1 主动发送消息 86

4.1.1 接口定义 86

4.1.2 消息类型 87

4.1.2.1 文本消息 87

4.1.2.2 图片消息 88

4.1.2.3 语音消息 89

4.1.2.4 视频消息 90

4.1.2.5 文件消息 92

4.1.2.6 文本卡片消息 93

4.1.2.7 图文消息 95

4.2 接收消息服务器配置 98

4.2.1 关于接收消息 98

4.2.2 开启接收消息 98

4.2.3 使用接收消息 100

4.2.4 消息的加解密处理 102

4.3 接收事件推送 105

4.3.1 接收事件推送 105

4.3.2 事件类型 105

4.3.2.1 成员关注/取消关注事件 105

4.3.2.2 进入应用 106

4.3.2.3 上报地理位置 107

4.3.2.4 异步任务完成事件推送 109

4.3.2.5 通讯录变更事件 110

4.3.2.6 菜单事件 112

4.4 接收普通消息 123

4.4.1 接收普通消息 123

4.4.2 消息类型 123

4.4.2.1 文本消息 123

4.4.2.2 图片消息 124

4.4.2.3 语音消息 125

4.4.2.4 视频消息 126

4.4.2.5 位置消息 127

4.4.2.6 链接消息 128

4.5 被动回复消息 130

4.5.1 被动响应 130

4.5.2 消息类型 130

1.1.1.1 文本消息 130

4.5.2.1 图片消息 131

4.5.2.2 语音消息 132

4.5.2.3 视频消息 133

4.5.2.4 图文消息 134

5 素材管理 136

5.1 上传临时素材文件 136

5.2 获取临时素材文件 138

6 网页开发 139

6.1 JS-SDK 139

6.1.1 概述 139

6.1.2 使用说明 140

5.1.2.1 步骤一：引入JS文件 140

5.1.2.2 步骤二：通过config接口注入权限验证配置 141

5.1.2.3 步骤三：通过ready接口处理成功验证 142

5.1.2.4 步骤四：通过error接口处理失败验证 142

5.1.2.5 接口调用说明 142

6.1.3 基础接口 143

6.1.4 通讯录和会话接口 144

5.1.4.1 创建会话接口 145

6.1.5 分享接口 145

6.1.6 图像与文件接口 146

6.1.7 音频接口 148

6.1.8 设备信息 149

6.1.9 地理位置 150

6.1.10 界面操作 150

6.1.11 企业微信扫一扫 152

6.1.12 附录1-JS-SDK使用权限签名算法 152

6.1.13 附录2-所有JS接口列表 156

6.1.14 附录3-所有菜单项列表 157

6.1.15 附录4-常见错误及解决方法 158

6.2 网页授权 161

6.2.1 概述 161

6.2.2 关于网页授权的可信域名 161

6.2.3 关于UserID机制 162

6.2.4 接入流程说明 162

6.2.5 关键步骤 163

5.2.5.1 获取code 163

5.2.5.2 根据code获取成员信息 165

5.2.5.3 使用user\_ticket获取成员详情 167

7 企业微信登录 169

7.1 移动应用 169

7.1.1 IOS应用 169

7.1.2 Android应用 176

7.1.3 资源下载 184

7.2 网页应用 185

7.2.1 概叙 185

7.2.2 开启网页接入 186

7.2.3 使用说明 187

7.2.4 F.A.Q 191

8 SDK 192

8.1 SDK 192

8.1.1 SDK介绍 192

9 第三方认证 193

10 短信邮件网关对接 195

10.1 短信邮件服务器配置 195

10.1.1 开启短信邮件对接 195

10.1.2 使用网关接口 197

10.2 短信邮件消息对接 198

10.2.1 短信消息对接 198

10.2.2 邮件消息对接 199

11 附录 201

11.1 加解密方案的详细说明 201

11.2 加解密库下载与返回码 203

11.2.1 加解密库的返回码 203

11.2.2 加解密库下载及示例 204

c++库 204

python库 204

php库 205

java库 205

c#库 206

11.3 接口代码参考示例 207

11.3.1 开启接收消息模式 207

11.3.2 通讯录接口 208

11.3.3 应用管理 210

11.3.4 消息服务 210

11.3.5 网页开发 213

11.3.6 代码示例下载 217

11.4 全局错误码 217

11.5 联系我们 232

# 开发前必读

## 概述

欢迎使用企业微信

企业微信提供了通讯录同步、应用管理、消息推送、OAuth2用户身份识别、JS-SDK等API，管理员可以使用这些API，为企业接入更多个性化 的办公应用。

### 术语介绍

1. corpid：每个企业都拥有唯一的corpid，获取此信息可在管理后台“我的企业”－“企业信息”下查看（需要有管理员权限）
2. secret：secret是企业应用里面用于保障数据安全的“钥匙”，每一个应用都有一个独立的访问密钥，为了保证数据的安全，secret务必不能泄漏。通讯录接口的密钥在“管理工具”-“通讯录同步助手”里面查看，企业自定义的应用的密钥可以在企业应用的详情里面手动生成
3. access\_token：access\_token是企业后台去企业微信的后台获取信息时的重要票据，所有接口在通信时都需要携带此信息用于验证接口的访问权限

### 开始

第一步：创建企业应用

登录企业管理后台进入“企业应用”页面，通讯录管理是企业微信默认集成的应用，可以直接开启，如果企业需要开发自定义的应用，可点击“添加应用”完成应用的添加和配置，详细步骤请参见应用概述。

第二步：开启接收消息模式

开启接收消息模式并不是必须步骤，但是如果在你的企业应用中需要用到如下功能时需提前开启接收消息模式

* 获取企业成员的地理位置信息
* 动态调整企业应用的信息
* 获取企业成员点击事件类型的应用菜单行为
* 获取企业成员通过应用给企业后台发送的消息

关于如何开启接收消息模式，请阅读**接收消息模式设置**章节。

第三步：获取access\_token

**请求方式：**GET  
**请求URL：**http://wwlocal.qq.com/cgi-bin/gettoken?corpid=id&corpsecret=secrect

**参数说明：**

| **参数** | **必须** | **说明** |
| --- | --- | --- |
| corpid | 是 | 企业ID |
| corpsecret | 是 | 应用的凭证密钥 |

**权限说明：**

每个应用有独立的secret，所以每个应用的access\_token应该分开来获取

**返回结果：**

1. {
2. "errcode":0，
3. "errmsg":""，
4. "access\_token": "accesstoken000001",
5. "expires\_in": 7200
6. }

| **参数** | **说明** |
| --- | --- |
| access\_token | 获取到的凭证,最长为512字节 |
| expires\_in | 凭证的有效时间（秒） |

**出错返回示例：**

1. {
2. "errcode":40091,
3. "errmsg":"provider\_secret is invalid"
4. }

**版本信息：**  
v1.0以上版本有效

第四步：对access\_token进行缓存处理

为了保障服务器的稳定性，企业微信对access\_token每天的调用有频次限制，目前为2000次/天/应用，每个access\_token的有效期为7200秒（2小时），所以为了防止因为频率调用次数超出限制而影响功能正常使用的问题，建议开发者将中间生成的access\_token 进行缓存，过期以后再重新获取。同时由于企业微信每个应用的access\_token是彼此独立的，所以进行缓存时需要区分应用来进行存储。

第五步：开发应用逻辑

到这里，准备工作已经就绪，你可以继续阅读下面的接口文档了解更多关于企业微信提供的开放能力，并将这些能力与你的应用进行结合，开始吧！

# 通讯录同步

## 概述

通讯录同步相关接口，可以对部门和成员进行查询、添加、修改、删除操作。

开始开发：

1、进入企业微信管理后台，在“管理工具” — “通讯录同步助手”开启“API接口同步”  
  
2、开启后，可设置通讯录API的权限：读取或者编辑通讯录  
  
3、使用通讯录同步助手的secret进行开发。点击查看 示例代码

只有管理员才可以看到“管理工具” - “通讯录同步助手”这个工具，其他分级管理员无操作权限。当企业开启API接口编辑通讯录后，可接收成员变更个人信息的事件。

## 成员管理

### 创建成员

**请求方式：**POST  
**请求地址：**http://wwlocal.qq.com/cgi-bin/user/create?access\_token=ACCESS\_TOKEN

**请求包体：**

1. {
2. "userid": "zhangsan",
3. "name": "张三",
4. "english\_name": "jackzhang"
5. "mobile": "15913215421",
6. "department": [1, 2],
7. "order":[10,40],
8. "position": "产品经理",
9. "gender": "1",
10. "email": "zhangsan@gzdev.com",
11. "isleader": 1,
12. "enable":1,
13. "avatar\_mediaid": "2-G6nrLmr5EC3MNb\_-zL1dDdzkd0p7cNliYu9V5w7o8K0",
14. "telephone": "020-123456"，
15. "extattr": {"attrs":[{"name":"爱好","value":"旅游"},{"name":"卡号","value":"1234567234"}]}
16. }

**参数说明：**

| **参数** | **必须** | **说明** |
| --- | --- | --- |
| access\_token | 是 | 调用接口凭证 |
| userid | 是 | 成员UserID。对应管理端的帐号，企业内必须唯一。不区分大小写，长度为1~64个字节 |
| name | 是 | 成员名称。长度为1~64个字节 |
| english\_name | 否 | 英文名。长度为1-64个字节。第三方暂不支持 |
| mobile | 是 | 手机号码。企业内必须唯一 |
| department | 是 | 成员所属部门id列表,不超过20个 |
| order | 否 | 部门内的排序值，默认为0。数量必须和department一致，数值越大排序越前面。第三方暂不支持 |
| position | 否 | 职位信息。长度为0~64个字节 |
| gender | 否 | 性别。1表示男性，2表示女性 |
| email | 否 | 邮箱。长度为0~64个字节。企业内必须唯一 |
| telephone | 否 | 座机。长度0-64个字节。第三方暂不支持 |
| isleader | 否 | 上级字段，标识是否为上级。第三方暂不支持 |
| avatar\_mediaid | 否 | 成员头像的mediaid，通过多媒体接口上传图片获得的mediaid |
| enable | 否 | 启用/禁用成员。1表示启用成员，0表示禁用成员 |
| extattr | 否 | 自定义字段。自定义字段需要先在WEB管理端“我的企业” — “通讯录管理”添加，否则忽略未知属性的赋值 |

**权限说明：**

系统应用须拥有指定部门的管理权限。

注意，每个部门下的节点不能超过3万个。

**返回结果：**

1. {
2. "errcode": 0,
3. "errmsg": "created",
4. "init\_pwd": "Td789456"
5. }

**参数说明：**

| **参数** | **说明** |
| --- | --- |
| errcode | 返回码 |
| errmsg | 对返回码的文本描述内容 |
| init\_pwd | 创建成员的初始登录密码 |

**版本信息：**  
v1.0以上版本有效

### 读取成员

在通讯录同步助手中此接口可以读取企业通讯录的所有成员信息，而企业自定义的应用可以读取该应用设置的可见范围内的成员信息。

**请求方式：**GET  
**请求地址：**http://wwlocal.qq.com/cgi-bin/user/get?access\_token=ACCESS\_TOKEN&userid=USERID

**参数说明：**

| **参数** | **必须** | **说明** |
| --- | --- | --- |
| access\_token | 是 | 调用接口凭证 |
| userid | 是 | 成员UserID。对应管理端的帐号，企业内必须唯一。不区  分大小写，长度为1~64个字节 |

**权限说明：**

系统应用须拥有指定部门的管理权限。

**返回结果：**

1. {
2. "errcode": 0,
3. "errmsg": "ok",
4. "userid": "zhangsan",
5. "name": "李四",
6. "department": [1, 2],
7. "position": "后台工程师",
8. "mobile": "15913215421",
9. "gender": "1",
10. "email": "zhangsan@gzdev.com",
11. "isleader": 1,
12. "avatar": "http://wx.qlogo.cn/mmopen/ajNVdqHZLLA3WJ6DSZUfiakYe37PKnQhBIeOQBO4czqrnZDS79FH5Wm5m4X69TBicnHFlhiafvDwklOpZeXYQQ2icg/0",
13. "telephone": "020-123456",
14. "english\_name": "jackzhang",
15. "extattr": {"attrs":[{"name":"爱好","value":"旅游"},{"name":"卡号","value":"1234567234"}]}，
16. "enable": 1,
17. "status": 1
18. }

**参数说明：**

| **参数** | **说明** |
| --- | --- |
| errcode | 返回码 |
| errmsg | 对返回码的文本描述内容 |
| userid | 成员UserID。对应管理端的帐号 |
| name | 成员名称 |
| mobile | 手机号码，第三方不可获取 |
| department | 成员所属部门id列表 |
| position | 职位信息 |
| gender | 性别。0表示未定义，1表示男性，2表示女性 |
| email | 邮箱，第三方不可获取 |
| isleader | 上级字段，标识是否为上级。第三方暂不支持 |
| avatar | 头像url。注：如果要获取小图将url最后的”/0”改成”/100”即  可 |
| telephone | 座机。第三方暂不支持 |
| english\_name | 英文名。第三方暂不支持 |
| extattr | 扩展属性，第三方不可获取 |
| enable | 启用/禁用成员，第三方不可获取。1表示启用成员，0表示禁用成员 |
| status | 激活状态: 1=已激活，2=已禁用，4=未激活 |

**版本信息：**  
v1.0以上版本有效

### 更新成员

**请求方式：**POST  
**请求地址：**

**http://wwlocal.qq.com/cgi-bin/user/update?access\_token=ACCESS\_TOKEN**

**请求包体：**

1. {
2. "userid": "zhangsan",
3. "name": "李四",
4. "department": [1],
5. "order"：[10],
6. "position": "后台工程师",
7. "mobile": "15913215421",
8. "gender": "1",
9. "email": "zhangsan@gzdev.com",
10. "isleader": 0,
11. "enable": 1,
12. "avatar\_mediaid": "2-G6nrLmr5EC3MNb\_-zL1dDdzkd0p7cNliYu9V5w7o8K0",
13. "telephone": "020-123456",
14. "english\_name": "jackzhang",
15. "extattr": {"attrs":[{"name":"爱好","value":"旅游"},{"name":"卡号","value":"1234567234"}]}
16. }

**参数说明：**

| **参数** | **必须** | **说明** |
| --- | --- | --- |
| access\_token | 是 | 调用接口凭证 |
| userid | 是 | 成员UserID。对应管理端的帐号，企业内必须唯一。不区分大小写，长度为1~64个字节 |
| name | 否 | 成员名称。长度为1~64个字节 |
| english\_name | 否 | 英文名。长度为1-64个字节。第三方暂不支持 |
| mobile | 否 | 手机号码。企业内必须唯一。若成员已激活企业微信，则需成员自行修改 |
| department | 否 | 成员所属部门id列表,不超过20个 |
| order | 否 | 部门内的排序值，默认为0。数量必须和department一致，数值越大排序越前面。第三方暂不支持 |
| position | 否 | 职位信息。长度为0~64个字节 |
| gender | 否 | 性别。1表示男性，2表示女性 |
| email | 否 | 邮箱。长度为0~64个字节。企业内必须唯一 |
| telephone | 否 | 座机。长度0-64个字节。第三方暂不支持 |
| isleader | 否 | 上级字段，标识是否为上级。第三方暂不支持 |
| avatar\_mediaid | 否 | 成员头像的mediaid，通过多媒体接口上传图片获得的mediaid |
| enable | 否 | 启用/禁用成员。1表示启用成员，0表示禁用成员 |
| extattr | 否 | 扩展属性。扩展属性需要在WEB管理端创建后才生效，否则忽略未知属性的赋值 |

**权限说明：**

系统应用须拥有指定部门、成员的管理权限。

注意，每个部门下的节点不能超过3万个。

**返回结果：**

1. {
2. "errcode": 0,
3. "errmsg": "updated"
4. }

**版本信息：**  
v1.0以上版本有效

### 删除成员

**请求方式：**GET  
**请求地址：**http://wwlocal.qq.com/cgi-bin/user/delete?access\_token=ACCESS\_TOKEN&userid=USERID

**参数说明：**

| **参数** | **必须** | **说明** |
| --- | --- | --- |
| access\_token | 是 | 调用接口凭证 |
| userid | 是 | 成员UserID。对应管理端的帐号 |

**权限说明：**

系统应用须拥有指定成员的管理权限。

**返回结果：**

1. {
2. "errcode": 0,
3. "errmsg": "deleted"
4. }

**版本信息：**  
v1.0以上版本有效

### 批量删除成员

**请求方式：**POST  
**请求地址：**http://wwlocal.qq.com/cgi-bin/user/batchdelete?access\_token=ACCESS\_TOKEN

**请求包体：**

1. {
2. "useridlist": ["zhangsan", "lisi"]
3. }

**参数说明：**

| **参数** | **必须** | **说明** |
| --- | --- | --- |
| access\_token | 是 | 调用接口凭证 |
| useridlist | 是 | 成员UserID列表。对应管理端的帐号 |

**权限说明：**

系统应用须拥有指定成员的管理权限。

**返回结果：**

1. {
2. "errcode": 0,
3. "errmsg": "deleted"
4. }

**版本信息：**  
v1.0以上版本有效

### 获取部门成员

**请求方式：**GET  
**请求地址：**http://wwlocal.qq.com/cgi-bin/user/simplelist?access\_token=ACCESS\_TOKEN&department\_id=DEPARTMENT\_ID&fetch\_child=FETCH\_CHILD

**参数说明：**

| **参数** | **必须** | **说明** |
| --- | --- | --- |
| access\_token | 是 | 调用接口凭证 |
| department\_id | 是 | 获取的部门id |
| fetch\_child | 否 | 1/0：是否递归获取子部门下面的成员 |

**权限说明：**

系统应用须拥有指定部门的查看权限。

**返回结果：**

1. {
2. "errcode": 0,
3. "errmsg": "ok",
4. "userlist": [
5. {
6. "userid": "zhangsan",
7. "name": "李四",
8. "department": [1, 2]
9. }
10. ]
11. }

**参数说明：**

| **参数** | **说明** |
| --- | --- |
| errcode | 返回码 |
| errmsg | 对返回码的文本描述内容 |
| userlist | 成员列表 |
| userid | 成员UserID。对应管理端的帐号 |
| name | 成员名称 |
| department | 成员所属部门 |

**版本信息：**  
v1.0以上版本有效

### 获取部门成员详情

**请求方式：**GET  
**请求地址：**http://wwlocal.qq.com/cgi-bin/user/list?access\_token=ACCESS\_TOKEN&department\_id=DEPARTMENT\_ID&fetch\_child=FETCH\_CHILD

**参数说明：**

| **参数** | **必须** | **说明** |
| --- | --- | --- |
| access\_token | 是 | 调用接口凭证 |
| department\_id | 是 | 获取的部门id |
| fetch\_child | 否 | 1/0：是否递归获取子部门下面的成员 |

**权限说明：**

系统应用须拥有指定部门的查看权限。

**返回结果：**

1. {
2. "errcode": 0,
3. "errmsg": "ok",
4. "userlist": [
5. {
6. "userid": "zhangsan",
7. "name": "李四",
8. "department": [1, 2],
9. "position": "后台工程师",
10. "mobile": "15913215421",
11. "gender": "1",
12. "email": "zhangsan@gzdev.com",
13. "isleader": 0,
14. "avatar": "http://wx.qlogo.cn/mmopen/ajNVdqHZLLA3WJ6DSZUfiakYe37PKnQhBIeOQBO4czqrnZDS79FH5Wm5m4X69TBicnHFlhiafvDwklOpZeXYQQ2icg/0",
15. "telephone": "020-123456",
16. "english\_name": "jackzhang",
17. "enable": 1,
18. "status": 1,
19. "extattr": {"attrs":[{"name":"爱好","value":"旅游"},{"name":"卡号","value":"1234567234"}]}
20. }
21. ]
22. }

**参数说明 :**

| **参数** | **说明** |
| --- | --- |
| errcode | 返回码 |
| errmsg | 对返回码的文本描述内容 |
| userlist | 成员列表 |
| userid | 成员UserID。对应管理端的帐号 |
| name | 成员名称 |
| mobile | 手机号码，第三方不可获取 |
| department | 成员所属部门id列表 |
| order | 部门内的排序值。第三方暂不支持 |
| position | 职位信息 |
| gender | 性别。0表示未定义，1表示男性，2表示女性 |
| email | 邮箱，第三方不可获取 |
| isleader | 标示是否为上级。第三方暂不支持 |
| avatar | 头像url。注：如果要获取小图将url最后的”/0”改成”/100”即可 |
| telephone | 座机。第三方暂不支持 |
| english\_name | 英文名。第三方暂不支持 |
| enable | 启用/禁用成员，第三方不可获取。1表示启用成员，0表示禁用成员 |
| status | 激活状态: 1=已激活，2=已禁用，4=未激活 |
| extattr | 扩展属性，第三方不可获取 |

**版本信息：**  
v1.0以上版本有效

### 用户踢下线

**请求方式：**POST  
**请求地址：**https://saastest2.gov.weixin.qq.com/cgi-bin/user/offline?access\_token=ACCESS\_TOKEN

**请求包体：**

1. {
2. "userid\_list": ["sanzhang", "sili"]
3. }

**参数说明：**

| **参数** | **必须** | **说明** |
| --- | --- | --- |
| access\_token | 是 | 调用接口凭证 |
| userid\_list | 是 | 成员UserID。对应管理端的帐号，单位内必须唯一。不区分大小写，长度为1~64个字节 |

**权限说明：**

该接口只有通讯录同步助手可调用

**返回结果：**

1. {
2. "errcode": 0,
3. "errmsg": "ok",
4. "result":[
5. {
6. "userid": "username",
7. "errcode": 0,
8. "errmsg": "ok"
9. }
10. ]
11. }

**参数说明：**

| **参数** | **说明** |
| --- | --- |
| errcode | 返回码 |
| errmsg | 对返回码的文本描述内容 |
| userid | 用户id |

**版本信息：**  
v1.2以上版本有效

## 部门管理

### 创建部门

**请求方式：**POST  
**请求地址：**http://wwlocal.qq.com/cgi-bin/department/create?access\_token=ACCESS\_TOKEN

**请求包体：**

1. {
2. "name": "广州研发中心",
3. "parentid": 1,
4. "order": 1,
5. "id": 2
6. }

**参数说明：**

| **参数** | **必须** | **说明** |
| --- | --- | --- |
| access\_token | 是 | 调用接口凭证 |
| name | 是 | 部门名称。长度限制为1~64个字节，字符不能包括\:*?”<>｜* |
| parentid | 是 | 父部门id |
| order | 否 | 在父部门中的次序值。order值大的排序靠前。 |
| id | 否 | 部门id，整型。指定时必须大于1，否则自动生成 |

**权限说明：**

系统应用须拥有父部门的管理权限。

注意，部门的最大层级为15层；部门总数不能超过3万个；每个部门下的节点不能超过3万个。

**返回结果：**

1. {
3. "errcode": 0,
4. "errmsg": "created",
5. "id": 2
6. }

**参数说明：**

| **参数** | **说明** |
| --- | --- |
| errcode | 返回码 |
| errmsg | 对返回码的文本描述内容 |
| id | 创建的部门id |

**版本信息：**  
v1.0以上版本有效

### 更新部门

**请求方式：**POST  
**请求地址：**http://wwlocal.qq.com/cgi-bin/department/update?access\_token=ACCESS\_TOKEN

**请求包体**（如果非必须的字段未指定，则不更新该字段）：

1. {
2. "id": 2,
3. "name": "广州研发中心",
4. "parentid": 1,
5. "order": 1
6. }

**参数说明：**

| **参数** | **必须** | **说明** |
| --- | --- | --- |
| access\_token | 是 | 调用接口凭证 |
| id | 是 | 部门id |
| name | 否 | 部门名称。长度限制为1~64个字节，字符不能包括\:*?”<>｜* |
| parentid | 否 | 父部门id |
| order | 否 | 在父部门中的次序值。order值大的排序靠前。 |

**权限说明 ：**

系统应用须拥有指定部门的管理权限。

注意，部门的最大层级为15层；部门总数不能超过3万个；每个部门下的节点不能超过3万个。

**返回结果：**

1. {
2. "errcode": 0,
3. "errmsg": "updated"
4. }

**版本信息：**  
v1.0以上版本有效

### 删除部门

**请求方式：**GET  
**请求地址：**http://wwlocal.qq.com/cgi-bin/department/delete?access\_token=ACCESS\_TOKEN&id=ID

**参数说明 ：**

| **参数** | **必须** | **说明** |
| --- | --- | --- |
| access\_token | 是 | 调用接口凭证 |
| id | 否 | 部门id。（注：不能删除根部门；不能删除含有子部门、成员的部门） |

**权限说明：**

系统应用须拥有指定部门的管理权限。

**返回结果：**

1. {
2. "errcode": 0,
3. "errmsg": "deleted"
4. }

**版本信息：**  
v1.0以上版本有效

### 获取部门列表

**请求方式：**GET  
**请求地址：**http://wwlocal.qq.com/cgi-bin/department/list?access\_token=ACCESS\_TOKEN&id=ID

**参数说明 ：**

| **参数** | **必须** | **说明** |
| --- | --- | --- |
| access\_token | 是 | 调用接口凭证 |
| id | 否 | 部门id。获取指定部门及其下的子部门。 如果不填，默认获取全量组织架构 |

**权限说明：**

系统应用须拥有指定部门的查看权限。

**返回结果：**

1. Asd {
2. "errcode": 0,
3. "errmsg": "ok",
4. "department": [
5. {
6. "id": 2,
7. "name": "广州研发中心",
8. "parentid": 1,
9. "order": 10
10. },
11. {
12. "id": 3
13. "name": "邮箱产品部",
14. "parentid": 2,
15. "order": 40
16. }
17. ]
18. }

**参数说明：**

| **参数** | **说明** |
| --- | --- |
| errcode | 返回码 |
| errmsg | 对返回码的文本描述内容 |
| department | 部门列表数据。 |
| id | 创建的部门id |
| name | 部门名称 |
| parentid | 父亲部门id。根部门为1 |
| order | 在父部门中的次序值。order值大的排序靠前。 |

**版本信息：**  
v1.0以上版本有效

## 标签管理

### 创建标签

**请求方式：**POST

\_\_请求地址：http://wwlocal.qq.com/cgi-bin/tag/create?access\_token=ACCESS\_TOKEN

**请求包体：**

1. {
2. "tagname": "UI",
3. "tagid": 12
4. }

**参数说明：**

| **参数** | **必须** | **说明** |
| --- | --- | --- |
| access\_token | 是 | 调用接口凭证 |
| tagname | 是 | 标签名称，长度限制为32个字（汉字或英文字母），标签名不可与其他标签重名。 |
| tagid | 否 | 标签id，非负整型，指定此参数时新增的标签会生成对应的标签id，不指定时则以目前最大的id自增。 |

**权限说明：**

创建的标签属于该应用，只有该应用才可以增删成员。

注意，标签总数不能超过3000个。

**返回结果：**

1. {
2. "errcode": 0,
3. "errmsg": "created"
4. "tagid": 12
5. }

**版本信息：**  
v1.0以上版本有效

### 更新标签名字

**请求方式：**POST  
**请求地址：**http://wwlocal.qq.com/cgi-bin/tag/update?access\_token=ACCESS\_TOKEN

**请求包体：**

1. {
2. "tagid": 12,
3. "tagname": "UI design"
4. }

**参数说明：**

| **参数** | **必须** | **说明** |
| --- | --- | --- |
| access\_token | 是 | 调用接口凭证 |
| tagid | 是 | 标签ID |
| tagname | 是 | 标签名称，长度限制为32个字（汉字或英文字母），标签不可与其他标签重名。 |

**权限说明：**

调用者必须是指定标签的创建者。

**返回结果：\_\_**

1. {
2. "errcode": 0,
3. "errmsg": "updated"
4. }

**版本信息：**  
v1.0以上版本有效

### 删除标签

**请求方式：**GET  
**请求地址：**http://wwlocal.qq.com/cgi-bin/tag/delete?access\_token=ACCESS\_TOKEN&tagid=TAGID

**参数说明：**

| **参数** | **必须** | **说明** |
| --- | --- | --- |
| access\_token | 是 | 调用接口凭证 |
| tagid | 是 | 标签ID |

**权限说明：**

调用者必须是指定标签的创建者。

**返回结果：**

1. {
2. "errcode": 0,
3. "errmsg": "deleted"
4. }

**版本信息：**  
v1.0以上版本有效

### 获取标签成员

**请求方式：**GET  
**请求地址：**http://wwlocal.qq.com/cgi-bin/tag/get?access\_token=ACCESS\_TOKEN&tagid=TAGID

**参数说明：**

| **参数** | **必须** | **说明** |
| --- | --- | --- |
| access\_token | 是 | 调用接口凭证 |
| tagid | 是 | 标签ID |

**权限说明：**

无限制，但返回列表仅包含应用可见范围的成员；第三方仅可获取自己创建的标签

**返回结果：**

1. {
2. "errcode": 0,
3. "errmsg": "ok",
4. "userlist": [
5. {
6. "userid": "zhangsan",
7. "name": "李四"
8. }
9. ],
10. "partylist": [2]
11. }

**版本信息：**  
v1.0以上版本有效

### 增加标签成员

**请求方式：**POST  
**请求地址：**http://wwlocal.qq.com/cgi-bin/tag/addtagusers?access\_token=ACCESS\_TOKEN

**请求包体：**

1. {
2. "tagid": 12,
3. "userlist":[ "user1","user2"],
4. "partylist": [4]
5. }

**参数说明：**

| **参数** | **必须** | **说明** |
| --- | --- | --- |
| access\_token | 是 | 调用接口凭证 |
| tagid | 是 | 标签ID |
| userlist | 否 | 企业成员ID列表，注意：userlist、partylist不能同时为空，单次请求长度不超过1000 |
| partylist | 否 | 企业部门ID列表，注意：userlist、partylist不能同时为空，单次请求长度不超过100 |

**权限说明：**

调用者必须是指定标签的创建者；成员属于应用的可见范围。

注意，每个标签下的节点不能超过3万个。

**返回结果：**

a)正确时返回

1. {
2. "errcode": 0,
3. "errmsg": "ok"
4. }

b)若部分userid、partylist非法，则返回

1. {
2. "errcode": 0,
3. "errmsg": "错误消息",
4. "invalidlist"："usr1|usr2|usr",
5. "invalidparty"：[2,4]
6. }

其中错误消息视具体出错情况而定，分别为：  
invalid userlist and partylist faild  
invalid userlist faild  
invalid partylist faild

c)当包含userid、partylist全部非法时返回

1. {
2. "errcode": 40070,
3. "errmsg": "all list invalid "
4. }

**版本信息：**  
v1.0以上版本有效

### 删除标签成员

**请求方式：**POST  
**请求地址：**http://wwlocal.qq.com/cgi-bin/tag/deltagusers?access\_token=ACCESS\_TOKEN

**请求包体：**

1. {
2. "tagid": 12,
3. "userlist":[ "user1","user2"],
4. "partylist":[2,4]
5. }

**参数说明：**

| **参数** | **必须** | **说明** |
| --- | --- | --- |
| access\_token | 是 | 调用接口凭证 |
| tagid | 是 | 标签ID |
| userlist | 否 | 企业成员ID列表，注意：userlist、partylist不能同时为空 |
| partylist | 否 | 企业部门ID列表，注意：userlist、partylist不能同时为空 |

**权限说明：**

调用者必须是指定标签的创建者；成员属于应用的可见范围。

**返回结果：**

a)正确时返回

1. {
2. "errcode": 0,
3. "errmsg": "deleted"
4. }

b)若部分userid、partylist非法，则返回

1. {
2. "errcode": 0,
3. "errmsg": "错误消息",
4. "invalidlist"："usr1|usr2|usr",
5. "invalidparty": [2,4]
6. }

其中错误消息视具体出错情况而定，分别为：  
invalid userlist and partylist faild  
invalid userlist faild  
invalid partylist faild

c)当包含的userid、partylist全部非法时返回

1. {
2. "errcode": 40031,
3. "errmsg": "all list invalid"
4. }

**版本信息：**  
v1.0以上版本有效

### 获取标签列表

**请求方式：**GET  
**请求地址：**http://wwlocal.qq.com/cgi-bin/tag/list?access\_token=ACCESS\_TOKEN

**参数说明：**

| **参数** | **必须** | **说明** |
| --- | --- | --- |
| access\_token | 是 | 调用接口凭证 |

**权限说明：**

企业可获取所有标签列表；第三方仅可获取自己创建的标签。

**返回结果：**

1. {
2. "errcode": 0,
3. "errmsg": "ok",
4. "taglist":[
5. {"tagid":1,"tagname":"a"},
6. {"tagid":2,"tagname":"b"}
7. ]
8. }

**版本信息：**  
v1.0以上版本有效

# 应用管理与应用管理助手

## 概述

企业微信的应用具备以下能力：

**收发消息**

以普通会话的形式存在，可推送消息、接收消息，也可以设置自定义菜单。

* 通过“消息推送”API，推送应用消息给企业成员
* 开启“接收消息”，员工进入应用、发送消息、操作菜单等动作会以事件的方式转发给企业的应用后台
* 自定义7种个性化菜单

**应用主页**

应用主页是一个H5网页，配置以后就会出现在工作台，点击以后直接进入H5网页。

* Oauth2用户身份识别。
* JS-SDK调用原生客户端Native的能力。

**授权登录**

企业已有的Web网页、移动APP，可以使用企业微信的帐号登录。

* 扫一扫授权登录Web网页
* 嵌入登录SDK，一键登录移动APP

**开始开发**

应用的创建目前存在两种方式，第一种是进入管理后台，手动在单位应用里边创建应用（如下图）。

1、管理员进入企业微信“管理后台” — “企业应用”创建应用  
  
2、设置应用logo、名称、描述、可见范围等。

该种方式创建的应用可以在管理后台获取应用的secret，进而通过接口获取到接口调用凭证access\_token，该凭证在针对应用管理上，能有效调用”/cgi-bin/agent/get”、”/cgi-bin/agent/set”, “/cgi-bin/agent/list”，”/cgi-bin/menu/create”, “/cgi-bin/menu/get”, “/cgi-bin/menu/delete” 5个接口。对其他应用管理接口无操作能力。

第二种：通过“应用管理助手” 自助创建应用的方式，操作如下。  
1、进入蓝水源科技管理后台，在“管理工具” — “应用管理助手”开启“应用管理助手”  
  
2、开启应用管理助手之后，可通过API管理应用、应用分组、排序、工作台显示方式。  


应用管理助手的凭证拥有应用管理所有接口的调用权限。

开发阶段“应用可见范围”可先选择小范围可见，待开发完成后再开放给单位员工。应用管理助手在1.2及以后版本中生效。

## 应用管理

### 创建应用

**请求方式：**POST  
**请求地址：**https://saastest2.gov.weixin.qq.com/cgi-bin/agent/create?access\_token=ACCESS\_TOKEN

**请求示例：**

1. {
2. "report\_location\_flag": 0,
3. "logo\_mediaid": "xxxxx",
4. "name": "NAME",
5. "description": "DESC",
6. "redirect\_domain": "xxxxxx",
7. "isreportenter":0,
8. "home\_url":"http://www.qq.com",
9. "allow\_userinfos": {
10. "user": [
11. {
12. "userid": "userid1"
13. }
14. ]
15. },
16. "allow\_partys": {
17. "partyid": [
18. 1,34
19. ]
20. },
21. "allow\_tags": {
22. "tagid": [
23. 33
24. ]
25. }
26. }

**参数说明：**

| **参数** | **必须** | **说明** |
| --- | --- | --- |
| access\_token | 是 | 调用接口凭证 |
| name | 是 | 单位应用名称，1~32个字符 |
| logo\_mediaid | 是 | 单位应用头像的mediaid，通过多媒体接口上传图片获得mediaid，上传后会自动裁剪成方形和圆形两个头像 |
| description | 否 | 单位应用详情，4~120个字符 |
| report\_location\_flag | 否 | 单位应用是否打开地理位置上报 0：不上报；1：进入会话上报； |
| redirect\_domain | 否 | 单位应用可信域名 |
| isreportenter | 否 | 是否上报用户进入应用事件。0：不接收；1：接收。 |
| home\_url | 否 | 应用主页url。url必须以http或者https开头。 |
| isforbidden | 否 | 应用是否关闭，0：开启，1关闭 |
| allow\_userinfos | 否 | 应用可见范围（用户） |
| allow\_partys | 否 | 应用可见范围（部门） |
| alllow\_tags | 否 | 应用可见范围（标签） |

**权限说明：**

仅有应用管理助手的凭证具有操作权限

**返回结果 ：**

1. {
2. "errcode":0,
3. "errmsg":"ok",
4. "agentid": 1000022,
5. "secret": "d2XKjqLvrZFyp43MQgV487wwGeAOK72Gcc-W3q0NJRM"
6. }

**参数说明：**

| **参数** | **说明** |
| --- | --- |
| agentid | 被创建应用的id |
| secret | 被创建应用的secret |

**版本信息：**  
v1.2以上版本有效

### 获取应用

**请求方式：**GET  
**请求地址：**https://saastest2.gov.weixin.qq.com/cgi-bin/agent/get?access\_token=ACCESS\_TOKEN&agentid=AGENTID

**参数说明 ：**

| **参数** | **必须** | **说明** |
| --- | --- | --- |
| access\_token | 是 | 调用接口凭证 |
| agentid | 是 | 需要获取的应用id |

**权限说明：**  
非应用管理助手凭证可获取凭证对应的应用信息，应用管理助手凭证可获取任何非系统应用信息

**返回结果：**

1. {
2. "errcode":0,
3. "errmsg":"ok" ,
4. "agentid":"1" ,
5. "name":"NAME" ,
6. "logo\_mediaid":"xxxxxxxx" ,
7. "description":"desc" ,
8. "allow\_userinfos":{
9. "user":[
10. {
11. "userid":"id1",
12. },
13. {
14. "userid":"id2",
15. },
16. {
17. "userid":"id3",
18. }
19. ]
20. },
21. "allow\_partys":{
22. "partyid": [1, 2]
23. },
24. "allow\_tags": {
25. "tagid": [33, 32]
26. },
27. "close":0 ,
28. "redirect\_domain":"www.qq.com",
29. "report\_location\_flag":0,
30. "isreportenter":0,
31. "home\_url":"http://www.qq.com",
32. "callback\_open": 1,
33. "groupid": 0,
34. }

**参数说明：**

| **参数** | **说明** |
| --- | --- |
| agentid | 单位应用id |
| name | 单位应用名称 |
| logo\_mediaid | 单位应用的头像 |
| description | 单位应用详情 |
| allow\_userinfos | 单位应用可见范围（人员），其中包括userid |
| allow\_partys | 单位应用可见范围（部门） |
| allow\_tags | 单位应用可见范围（标签） |
| close | 单位应用是否被禁用 |
| redirect\_domain | 单位应用可信域名 |
| report\_location\_flag | 单位应用是否打开地理位置上报 0：不上报；1：进入会话上报； |
| isreportenter | 是否上报用户进入应用事件。0：不接收；1：接收 |
| home\_url | 应用主页url |
| callback\_open | 应用回调是否开启 |
| groupid | 应用所在分组的id，0为默认分组 |

**版本信息：**  
v1.0以上版本有效。 其中groupid字段在1.2及以上版本返回

### 设置应用

**请求方式：**POST  
**请求地址：**https://saastest2.gov.weixin.qq.com/cgi-bin/agent/set?access\_token=ACCESS\_TOKEN

**请求示例：**

1. {
2. "agentid": 5,
3. "report\_location\_flag": 0,
4. "logo\_mediaid": "xxxxx",
5. "name": "NAME",
6. "description": "DESC",
7. "redirect\_domain": "xxxxxx",
8. "isreportenter":0,
9. "home\_url":"http://www.qq.com",
10. "allow\_userinfos": {
11. "user": [
12. {
13. "userid": "userid1"
14. }
15. ]
16. },
17. "allow\_partys": {
18. "partyid": [
19. 1,34
20. ]
21. },
22. "allow\_tags": {
23. "tagid": [
24. 33
25. ]
26. },
27. "order": 10
28. }

**参数说明：**

| **参数** | **必须** | **说明** |
| --- | --- | --- |
| access\_token | 是 | 调用接口凭证 |
| agentid | 是 | 单位应用的id |
| report\_location\_flag | 否 | 单位应用是否打开地理位置上报 0：不上报；1：进入会话上报； |
| logo\_mediaid | 否 | 单位应用头像的mediaid，通过多媒体接口上传图片获得mediaid，上传后会自动裁剪成方形和圆形两个头像 |
| name | 否 | 单位应用名称 |
| description | 否 | 单位应用详情 |
| redirect\_domain | 否 | 单位应用可信域名 |
| isreportenter | 否 | 是否上报用户进入应用事件。0：不接收；1：接收。 |
| home\_url | 否 | 应用主页url。url必须以http或者https开头。 |
| allow\_userinfos | 否 | 应用可见范围（用户），仅当凭证是应用管理助手凭证有效 |
| allow\_partys | 否 | 应用可见范围（部门），仅当凭证是应用管理助手凭证有效 |
| alllow\_tags | 否 | 应用可见范围（标签），仅当凭证是应用管理助手凭证有效 |

**权限说明：**  
自建应用的凭证仅能设置对应应用的非可见范围字段。应用管理助手凭证可以设置非系统应用的所有属性字段，仅可设置系统应用的可见范围部分。

**返回结果 ：**

1. {
2. "errcode":0,
3. "errmsg":"ok"
4. }

### 删除应用

**请求方式：**POST  
**请求地址：**https://saastest2.gov.weixin.qq.com/cgi-bin/agent/delete?access\_token=ACCESS\_TOKEN

**请求示例：**

1. {
2. “agentid”: 5
3. }

**参数说明：**

| **参数** | **必须** | **说明** |
| --- | --- | --- |
| access\_token | 是 | 调用接口凭证 |
| agentid | 是 | 需要删除的应用的id |

**权限说明**  
仅应用管理助手凭证有效，系统应用不可删除。

**返回结果：**

1. {
2. "errcode": 0,
3. "errmsg": "ok",
4. "agentid": 5
5. }

**版本信息：**  
v1.2以上版本有效

### 启动应用

**请求方式：**GET  
**请求地址：**https://saastest2.gov.weixin.qq.com/cgi-bin/agent/open?access\_token=ACCESS\_TOKEN&agentid=AGENTID

**参数说明：**

| **参数** | **必须** | **说明** |
| --- | --- | --- |
| access\_token | 是 | 调用接口凭证 |
| agentid | 是 | 需要开启的应用的id |

**权限说明：**  
仅应用管理助手凭证有操作权限

**返回结果：**

1. {
2. "errcode": 0,
3. "errmsg": "ok",
4. "secret": "9hjZ-dlI3wKjDeuyVv\_5AjUGwE7TP0w8BB-tjsQIceA"
5. }

**参数说明：**

| **参数** | **说明** |
| --- | --- |
| secret | 开启应用的secret |

### 停用应用

**请求方式：**GET  
**请求地址：**https://saastest2.gov.weixin.qq.com/cgi-bin/agent/close?access\_token=ACCESS\_TOKEN&agentid=AGENATID

**参数说明：**

| **参数** | **必须** | **说明** |
| --- | --- | --- |
| access\_token | 是 | 调用接口凭证 |
| agentid | 是 | 需要关闭的应用的id |

**权限说明：**

仅应用管理助手凭证有操作权限

**返回结果：**

1. {
2. "errcode": 0,
3. "errmsg": "ok"
4. }

**版本信息：**  
v1.2以上版本有效

### 重置应用secret

**请求方式：**GET  
**请求地址：**https://saastest2.gov.weixin.qq.com/cgi-bin/agent/resetsecret?access\_token=ACCESS\_TOKEN&agentid=AGENTID

**参数说明 ：**

| **参数** | **必须** | **说明** |
| --- | --- | --- |
| access\_token | 是 | 调用接口凭证 |
| agentid | 是 | 需要重设secret的单位应用id |

**权限说明：**

仅应用管理助手凭证有操作权限,系统应用不可重置

**返回结果：**

**参数说明：**

1. {
2. "agentid": 1000021,
3. "secret": "965jVGDe0Sr4NHfA9eoEKrpaar1fPT\_D-HB1UKH-8ns",
4. "errcode": 0,
5. "errmsg": "ok"
6. }

| **参数** | **说明** |
| --- | --- |
| agentid | 该单位应用的id |
| secret | 该单位应用新的secret |

**版本信息：**  
v1.2以上版本有效

### 开启应用回调

**请求方式：**POST  
**请求地址：**https://saastest2.gov.weixin.qq.com/cgi-bin/agent/opencallback?access\_token=ACCESS\_TOKEN

**请求示例:**

1. {
2. "callback\_url": "http://xxx.xxx.xxx",
3. "token": "hJqcu3uJ9Tn2gXPmxx2w9kkCkCE2EPYo",
4. "aeskey": "6qkdMrq68nTKduznJYO1A37W2oEgpkMUvkttRToqhUt",
5. "agentid": 1000027
6. }

**参数说明：**

| **参数** | **必须** | **说明** |
| --- | --- | --- |
| access\_token | 是 | 调用接口凭证 |
| agentid | 是 | 需要开启回调的单位应用id |
| callback\_url | 是 | 单位应用的回调地址 |
| token | 是 | 可任意填写，用于生成签名（长度为3~32之间的英文或数字） |
| aeskey | 是 | 用于消息体的加密，是AES密钥的Base64编码（43位的英文或数字） |

**权限说明：**

不可以操作系统应用。 应用管理助手凭证具有操作自建应用的权限，自建应用凭证仅能操作凭证对应的应用

**返回结果：**

1. {
2. "errcode": 0,
3. "errmsg": "ok"
4. }

**版本信息：**  
v1.2以上版本有效

### 更新应用回调

**请求方式：**POST  
**请求地址：**https://saastest2.gov.weixin.qq.com/cgi-bin/agent/updatecallback?access\_token=ACCESS\_TOKEN

**请求示例:**

1. {
2. "callback\_url": "http://xxx.xxx.xxx",
3. "token": "hJqcu3uJ9Tn2gXPmxx2w9kkCkCE2EPYo",
4. "aeskey": "6qkdMrq68nTKduznJYO1A37W2oEgpkMUvkttRToqhUt",
5. "agentid": 1000027
6. }

**参数说明：**

| **参数** | **必须** | **说明** |
| --- | --- | --- |
| access\_token | 是 | 调用接口凭证 |
| agentid | 是 | 需要更新回调的单位应用id |
| callback\_url | 否 | 单位应用的回调地址 |
| token | 否 | 可任意填写，用于生成签名（长度为3~32之间的英文或数字） |
| aeskey | 否 | 用于消息体的加密，是AES密钥的Base64编码（43位的英文或数字） |

**权限说明：**

不可以操作系统应用。 应用管理助手凭证具有操作自建应用的权限，自建应用凭证仅能操作凭证对应的应用

**返回结果：**

1. {
2. "errcode": 0,
3. "errmsg": "ok"
4. }

**版本信息：**  
v1.2以上版本有效

### 停用应用回调

**请求方式：**POST  
**请求地址：**https://saastest2.gov.weixin.qq.com/cgi-bin/agent/closecallback?access\_token=ACCESS\_TOKEN

**请求示例:**

1. {
2. "agentid": 1000027
3. }

**参数说明：**

| **参数** | **必须** | **说明** |
| --- | --- | --- |
| access\_token | 是 | 调用接口凭证 |
| agentid | 是 | 需要更新回调的单位应用id |

**权限说明：**

不可以操作系统应用。 应用管理助手凭证具有操作自建应用的权限，自建应用凭证仅能操作凭证对应的应用

**返回结果：**

1. {
2. "errcode": 0,
3. "errmsg": "ok"
4. }

**版本信息：**  
v1.2以上版本有效

### 创建应用菜单

自定义菜单接口可实现多种类型按钮，如下：

| **字段值** | **功能名称** | **说明** |
| --- | --- | --- |
| click | 点击推事件 | 成员点击click类型按钮后，企业微信服务器会通过消息接口推送消息类型为event 的结构给开发者（参考消息接口指南），并且带上按钮中开发者填写的key值，开发者可以通过自定义的key值与成员进行交互； |
| view | 跳转URL | 成员点击view类型按钮后，企业微信客户端将会打开开发者在按钮中填写的网页URL，可与网页授权获取成员基本信息接口结合，获得成员基本信息。 |
| scancode\_push | 扫码推事件 | 成员点击按钮后，企业微信客户端将调起扫一扫工具，完成扫码操作后显示扫描结果（如果是URL，将进入URL），且会将扫码的结果传给开发者，开发者可用于下发消息。 |
| scancode\_waitmsg | 扫码推事件 且弹出“消息接收中”提示框 | 成员点击按钮后，企业微信客户端将调起扫一扫工具，完成扫码操作后，将扫码的结果传给开发者，同时收起扫一扫工具，然后弹出“消息接收中”提示框，随后可能会收到开发者下发的消息。 |
| pic\_sysphoto | 弹出系统拍照发图 | 弹出系统拍照发图 成员点击按钮后，企业微信客户端将调起系统相机，完成拍照操作后，会将拍摄的相片发送给开发者，并推送事件给开发者，同时收起系统相机，随后可能会收到开发者下发的消息。 |
| pic\_photo\_or\_album | 弹出拍照或者相册发图 | 成员点击按钮后，企业微信客户端将调起企业微信相册，完成选择操作后，将选择的相片发送给开发者的服务器，并推送事件给开发者，同时收起相册，随后可能会收到开发者下发的消息。 |
| pic\_weixin | 弹出企业微信相册发图器 | 成员点击按钮后，企业微信客户端将调起企业微信相册，完成选择操作后，将选择的相片发送给开发者的服务器，并推送事件给开发者，同时收起相册，随后可能会收到开发者下发的消息。 |
| location\_select | 弹出地理位置选择器 | 成员点击按钮后，企业微信客户端将调起地理位置选择工具，完成选择操作后，将选择的地理位置发送给开发者的服务器，同时收起位置选择工具，随后可能会收到开发者下发的消息。 |

**请求方式：**POST  
**请求地址：**https://saastest2.gov.weixin.qq.com/cgi-bin/menu/create?access\_token=ACCESS\_TOKEN&agentid=AGENTID

**示例：构造click和view类型的请求包如下**

1. {
2. "button":[
3. {
4. "type":"click",
5. "name":"今日歌曲",
6. "key":"V1001\_TODAY\_MUSIC"
7. },
8. {
9. "name":"菜单",
10. "sub\_button":[
11. {
12. "type":"view",
13. "name":"搜索",
14. "url":"http://www.soso.com/"
15. },
16. {
17. "type":"click",
18. "name":"赞一下我们",
19. "key":"V1001\_GOOD"
20. }
21. ]
22. }
23. ]
24. }

**示例：其他新增按钮类型的请求**

1. {
2. "button": [
3. {
4. "name": "扫码",
5. "sub\_button": [
6. {
7. "type": "scancode\_waitmsg",
8. "name": "扫码带提示",
9. "key": "rselfmenu\_0\_0",
10. "sub\_button": [ ]
11. },
12. {
13. "type": "scancode\_push",
14. "name": "扫码推事件",
15. "key": "rselfmenu\_0\_1",
16. "sub\_button": [ ]
17. }
18. ]
19. },
20. {
21. "name": "发图",
22. "sub\_button": [
23. {
24. "type": "pic\_sysphoto",
25. "name": "系统拍照发图",
26. "key": "rselfmenu\_1\_0",
27. "sub\_button": [ ]
28. },
29. {
30. "type": "pic\_photo\_or\_album",
31. "name": "拍照或者相册发图",
32. "key": "rselfmenu\_1\_1",
33. "sub\_button": [ ]
34. },
35. {
36. "type": "pic\_weixin",
37. "name": "微信相册发图",
38. "key": "rselfmenu\_1\_2",
39. "sub\_button": [ ]
40. }
41. ]
42. },
43. {
44. "name": "发送位置",
45. "type": "location\_select",
46. "key": "rselfmenu\_2\_0"
47. }
48. ]
49. }

**参数说明：**

| **参数** | **必须** | **说明** |
| --- | --- | --- |
| access\_token | 是 | 调用接口凭证 |
| agentid | 是 | 单位应用的id，整型。可在应用的设置页面查看 |
| button | 是 | 一级菜单数组，个数应为1~3个 |
| sub\_button | 否 | 二级菜单数组，个数应为1~5个 |
| type | 是 | 菜单的响应动作类型 |
| key | click等点击类型必须 | 菜单KEY值，用于消息接口推送，不超过128字节 |
| url | view类型必须 | 网页链接，成员点击菜单可打开链接，不超过1024字节 |

**权限说明 :**

不可以操作系统应用。 应用管理助手凭证具有操作自建应用的权限，自建应用凭证仅能操作凭证对应的应用

**返回结果 ：**

1. {
2. "errcode":0,
3. "errmsg":"ok"
4. }

**版本信息：**  
v1.0以上版本有效

### 获取应用菜单

**请求方式：**GET  
**请求地址：**https://saastest2.gov.weixin.qq.com/cgi-bin/menu/get?access\_token=ACCESS\_TOKEN&agentid=AGENTID

**参数说明：**

| **参数** | **必须** | **说明** |
| --- | --- | --- |
| access\_token | 是 | 调用接口凭证 |
| agentid | 是 | 需要获取菜单的应用id |

**权限说明：**

不可以操作系统应用。 应用管理助手凭证具有操作自建应用的权限，自建应用凭证仅能操作凭证对应的应用

**返回结果：**

返回结果与请参考菜单创建接口

**版本信息：**  
v1.0以上版本有效

### 删除应用菜单

**请求方式：**GET  
**请求地址：**https://saastest2.gov.weixin.qq.com/cgi-bin/menu/delete?access\_token=ACCESS\_TOKEN&agentid=AGENTID

**参数说明：**

| **参数** | **必须** | **说明** |
| --- | --- | --- |
| access\_token | 是 | 调用接口凭证 |
| agentid | 是 | 需要删除菜单的应用id |

**权限说明：**

不可以操作系统应用。 应用管理助手凭证具有操作自建应用的权限，自建应用凭证仅能操作凭证对应的应用

**返回结果：**

1. {
2. "errcode":0,
3. "errmsg":"ok"
4. }

**版本信息：**  
v1.0以上版本有效

### 获取应用列表

**请求方式：**POST  
**请求地址：**https://saastest2.gov.weixin.qq.com/cgi-bin/agent/list?access\_token=ACCESS\_TOKEN

**请求示例：**

1. {
2. "agent\_status": 1
3. }

**参数说明：**

| **参数** | **必须** | **说明** |
| --- | --- | --- |
| access\_token | 是 | 调用凭证 |
| agent\_status | 否 | 0：获取所有停用的应用； 1： 获取所有打开的应用 |

**权限说明：**

自建应用凭证仅能操作凭证对应的应用，应用管理助手凭证具有获取所有非系统应用的权限

**返回结果：**

1. {
2. "errcode": 0,
3. "errmsg": "ok",
4. "agentlist": [
5. {
6. "agentid": 1000030,
7. "name": "hahah",
8. "logo\_mediaid": "xxxxxxxxxx",
9. "description": "hahah",
10. "allow\_userinfos": {
11. "user": [
12. {
13. "userid": "userid1"
14. }
15. ]
16. },
17. "allow\_partys": {
18. "partyid": [
19. 1
20. ]
21. },
22. "allow\_tags": {
23. "tagid": [
24. 33
25. ]
26. },
27. "close": 0,
28. "redirect\_domain": "",
29. "isreportenter": 0,
30. "home\_url": "",
31. "callback\_open": 1,
32. "groupid": 0,
33. "arch\_rw\_flag": 0
34. },
35. {
36. "agentid": 1000031,
37. "name": "franktest10",
38. "logo\_mediaid": "xxxxxxxxx",
39. "description": "",
40. "allow\_userinfos": {
41. "user": [
42. {
43. "userid": "userid1"
44. }
45. ]
46. },
47. "allow\_partys": {
48. "partyid": [
49. 1
50. ]
51. },
52. "allow\_tags": {
53. "tagid": [
54. 34
55. ]
56. },
57. "close": 0,
58. "redirect\_domain": "",
59. "isreportenter": 0,
60. "home\_url": "",
61. "callback\_open": 1,
62. "groupid": 0,
63. "arch\_rw\_flag": 0
64. }
65. ]
66. }

| **参数** | **说明** |
| --- | --- |
| agentid | 单位应用id |
| name | 单位应用名称 |
| logo\_mediaid | 单位应用的头像 |
| description | 单位应用详情 |
| allow\_userinfos | 单位应用可见范围（人员），其中包括userid |
| allow\_partys | 单位应用可见范围（部门） |
| allow\_tags | 单位应用可见范围（标签） |
| close | 单位应用是否被禁用 |
| redirect\_domain | 单位应用可信域名 |
| report\_location\_flag | 单位应用是否打开地理位置上报 0：不上报；1：进入会话上报； |
| isreportenter | 是否上报用户进入应用事件。0：不接收；1：接收 |
| home\_url | 应用主页url |
| callback\_open | 应用回调是否开启 |

**版本信息：**  
v1.0以上版本有效

## 分组管理

### 创建应用分组

**请求方式：**POST

**请求地址：**https://saastest2.gov.weixin.qq.com/cgibin/group/create?access\_token=ACCESS\_TOKEN

**请求示例：**

1. {
2. "name": NAME
3. }

**参数说明：**

| **参数** | **必须** | **说明** |
| --- | --- | --- |
| access\_token | 是 | 调用接口凭证 |
| name | 是 | 应用分组的名字16个字符以内 |

**返回结果：**

1. {
2. "errcode": 0,
3. "errmsg": "ok",
4. "groupid": 123
5. }

| **参数** | **说明** |
| --- | --- |
| groupid | 创建应用分组的id,有效的值范围是(0, 2^32) |

**版本信息：**  
v1.2及以上版本有效

### 更新应用分组

**请求方式：**POST  
**请求地址：**https://saastest2.gov.weixin.qq.com/cgibin/group/create?access\_token=ACCESS\_TOKEN

**请求示例：**

1. {
2. "groupid": 124,
3. "name": NAME,
4. "order":12
5. }

**参数说明：**

| **参数** | **必须** | **说明** |
| --- | --- | --- |
| access\_token | 是 | 调用接口凭证 |
| groupid | 是 | 应用分组的id |
| name | 否 | 应用分组的名字， 1-16个字符 |
| order | 否 | 应用分组的排序，有效的值范围是(0, 2^32) |

**权限说明 ：**

仅有应用管理助手的凭证具有操作权限

**返回结果：**

1. {
2. "errcode": 0,
3. "errmsg": "ok",
4. }

**版本信息：**  
v1.2及以上版本有效

### 删除应用分组

**请求方式：**POST  
**请求地址：**https://saastest2.gov.weixin.qq.com/cgibin/group/delete?access\_token=ACCESS\_TOKEN

**请求示例：**

1. {
2. "groupid": 5
3. }

**参数说明：**

| **参数** | **必须** | **说明** |
| --- | --- | --- |
| access\_token | 是 | 调用接口凭证 |
| groupid | 是 | 需要删除的分组id |

**权限说明 ：**

仅有应用管理助手的凭证具有操作权限

**接口说明：**  
分组删除后，分组中原有的应用将被添加到默认分组中。

**返回结果：**

1. {
2. "errcode": 0,
3. "errmsg": "ok"
4. }

**版本信息：**  
v1.2及以上版本有效

### 增加分组应用

**请求方式：**POST  
**请求地址：**https://saastest2.gov.weixin.qq.com/cgibin/group/addapps?access\_token=ACCESS\_TOKEN

**请求示例：**

1. {
2. "groupid":5,
3. "agentid\_list": [12, 34]
4. }

**参数说明：**

| **参数** | **必须** | **说明** |
| --- | --- | --- |
| access\_token | 是 | 调用接口凭证 |
| groupid | 是 | 应用分组的id |
| agentid\_list | 是 | 需要添加的应用id 列表 |

**返回值：**

1. {
2. "errcode": 0,
3. "errmsg": "ok"
4. }

**版本信息：**  
v1.2及以上版本有效

### 删除分组应用

**请求方式：**POST  
**请求地址：**https://saastest2.gov.weixin.qq.com/cgibin/group/removeapp?access\_token=ACCESS\_TOKEN

**请求示例：**

1. {
2. "groupid": 5,
3. "agentid":12
4. }

**参数说明：**

| **参数** | **必须** | **说明** |
| --- | --- | --- |
| access\_token | 是 | 调用接口凭证 |
| groupid | 是 | 应用分组的id |
| agentid | 是 | 需要从分组中删除的应用id |

权限说明 ：

仅有应用管理助手的凭证具有操作权限

接口说明：  
从分组中移除的应用被添加到默认分组

**返回值：**

1. {
2. "errcode": 0,
3. "errmsg": "ok"
4. }

**版本信息：**  
v1.2及以上版本有效

### 获取指定分组的应用列表

**请求方式：**POST  
**请求地址：**https://saastest2.gov.weixin.qq.com/cgibin/group/applist?access\_token=ACCESS\_TOKEN

**请求示例：**

1. {
2. "groupid": 5
3. }

**参数说明：**

| **参数** | **必须** | **说明** |
| --- | --- | --- |
| access\_token | 是 | 调用接口凭证 |
| groupid | 是 | 应用分组的id |

**返回值：**

1. {
2. "errcode": 0,
3. "errmsg": "ok",
4. "applist": [
5. {
6. "agentid": 3010034,
7. "name": "xxxxx",
8. "order": 1,
9. "imgid": "",
10. "groupid": 4
11. },
12. {
13. "agentid": 3010034,
14. "name": "xxxxx",
15. "order": 1,
16. "imgid": "",
17. "groupid": 4
18. }
19. ]
20. }

**参数说明：**

| **参数** | **说明** |
| --- | --- |
| agentid | 单位应用id |
| name | 单位应用的名字 |
| imgid | 应用的头像 |
| order | 应用在分组中的排序 |
| groupid | 应用分组id |

**版本信息：**  
v1.2及以上版本有效

### 获取分组列表

**请求方式：**GET  
**请求地址：**https://saastest2.gov.weixin.qq.com/cgibin/group/list?access\_token=ACCESS\_TOKEN

**参数说明：**

| **参数** | **必须** | **说明** |
| --- | --- | --- |
| access\_token | 是 | 调用接口凭证 |

权限说明 ：

仅有应用管理助手的凭证具有操作权限

**返回值：**

1. {
2. "errcode": 0,
3. "errmsg": "ok",
4. "grouplist": [
5. {
6. "groupid": 5,
7. "name": "测试2",
8. "order": 1,
9. "applist": [
10. {
11. "agentid": 1000010,
12. "name": "测试新窗口打开",
13. "order": 10000,
14. "imgid": "xxxxxx",
15. "groupid": 5
16. },
17. {
18. "agentid": 1000015,
19. "name": "test",
20. "order": 20000,
21. "imgid": "xxxxx",
22. "groupid": 5
23. }
24. ]
25. },
26. {
27. "groupid": 0,
28. "name": "",
29. "order": 0,
30. "applist": [
31. {
32. "agentid": 3010004,
33. "name": "xxxx",
34. "order": 10,
35. "imgid": "xxxxxxx",
36. "groupid": 0
37. }
38. ]
39. }
40. ]
41. }

**参数说明：**

| **参数** | **说明** |
| --- | --- |
| apps | 分组中的应用列表 |
| agentid | 单位应用id |
| name | 单位应用的名字 |
| imgid | 应用的头像 |
| order | 应用在分组中的排序 |
| groupid | 应用分组id |

**版本信息：**  
v1.2及以上版本有效

## ****设置工作台显示方式****

**请求方式：**POST  
**请求地址：**https://saastest2.gov.weixin.qq.com/cgi-bin/corp/setworkbench?access\_token=ACCESS\_TOKEN

**请求示例：**

1. {
2. "show\_type": 1
3. }

**参数说明：**

| **参数** | **必须** | **说明** |
| --- | --- | --- |
| access\_token | 是 | 调用接口凭证 |
| show\_type | 是 | 1：九宫格格式； 0：列表格式 |

权限说明 ：

仅有应用管理助手的凭证具有操作权限

**返回值：**

1. {
2. "errcode": 0,
3. "errmsg": "ok"
4. }

**版本信息：**  
v1.2以上版本有效

# 消息推送

## 主动发送消息

### 接口定义

应用支持推送文本、图片、视频、文件、图文等类型。

**请求方式：**POST  
**请求地址：** http://wwlocal.qq.com/cgi-bin/message/send?access\_token=ACCESS\_TOKEN

**参数说明：**

| **参数** | **是否必须** | **说明** |
| --- | --- | --- |
| access\_token | 是 | 调用接口凭证 |

各个消息类型的具体POST格式参考以下文档。

**返回示例：**

1. {
2. "errcode" : 0,
3. "errmsg" : "ok",
4. "invaliduser" : "UserID1", // 不区分大小写，返回的列表都统一转为小写
5. "invalidparty" : "PartyID1",
6. }

1、如果无权限，则本次发送失败  
2、如果接收人不存在，发送仍然执行  
以上两种情况下均返回无效的部分

### 消息类型

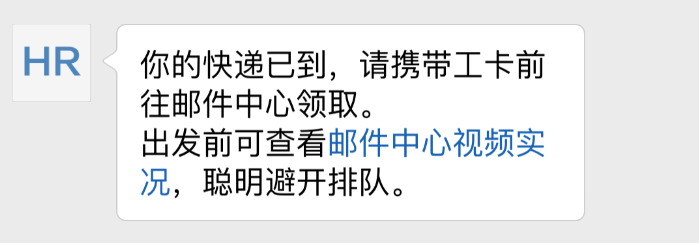
#### 文本消息

**请求示例：**

1. {
2. "touser" : "UserID1|UserID2|UserID3",
3. "toparty" : " PartyID1|PartyID2 ",
4. "totag" : " TagID1 | TagID2 ",
5. "msgtype" : "text",
6. "agentid" : 1,
7. "text" : {
8. "content" : "你的快递已到，请携带工卡前往邮件中心领取。<br>出发前可查看<a href=\"http://wwlocal.qq.com\">邮件中心视频实况</a>，聪明避开排队。"
9. }
10. }

**参数说明：**

| **参数** | **是否必须** | **说明** |
| --- | --- | --- |
| touser | 否 | 成员ID列表（消息接收者，多个接收者用‘|’分隔，最多支持1000个）。特殊情况：指定为@all，则向该企业应用的全部成员发送 |
| toparty | 否 | 部门ID列表，多个接收者用‘|’分隔，最多支持100个。当touser为@all时忽略本参数 |
| totag | 否 | 标签ID列表，多个接收者用‘|’分隔，最多支持100个。当touser为@all时忽略本参数 |
| msgtype | 是 | 消息类型，此时固定为：text |
| agentid | 是 | 企业应用的id，整型。可在应用的设置页面查看 |
| content | 是 | 消息内容，最长不超过2048个字节 |

**文本消息展现：**  


**特殊说明：**  
其中text参数的content字段可以支持换行、以及A标签，即可打开自定义的网页（可参考以上示例代码）

#### 图片消息

**请求示例：**

1. {
2. "touser" : "UserID1|UserID2|UserID3",
3. "toparty" : " PartyID1|PartyID2 ",
4. "totag" : " TagID1 | TagID2 ",
5. "msgtype" : "image",
6. "agentid" : 1,
7. "image" : {
8. "media\_id" : "MEDIA\_ID"
9. }
10. }
11. }
12. }

**请求参数：**

| **参数** | **是否必须** | **说明** |
| --- | --- | --- |
| touser | 否 | 成员ID列表（消息接收者，多个接收者用‘|’分隔，最多支持1000个）。特殊情况：指定为@all，则向关注该企业应用的全部成员发送 |
| toparty | 否 | 部门ID列表，多个接收者用‘|’分隔，最多支持100个。当touser为@all时忽略本参数 |
| totag | 否 | 标签ID列表，多个接收者用‘|’分隔，最多支持100个。当touser为@all时忽略本参数 |
| msgtype | 是 | 消息类型，此时固定为：image |
| agentid | 是 | 企业应用的id，整型。可在应用的设置页面查看 |
| media\_id | 是 | 图片媒体文件id，可以调用上传临时素材接口获取 |

#### 语音消息

**请求示例：**

1. {
2. "touser" : "UserID1|UserID2|UserID3",
3. "toparty" : " PartyID1|PartyID2 ",
4. "totag" : " TagID1 | TagID2 ",
5. "msgtype" : "voice",
6. "agentid" : 1,
7. "voice" : {
8. "media\_id" : "MEDIA\_ID"
9. }
10. }

**参数说明：**

| **参数** | **是否必须** | **说明** |
| --- | --- | --- |
| touser | 否 | 成员ID列表（消息接收者，多个接收者用‘|’分隔，最多支持1000个）。特殊情况：指定为@all，则向关注该企业应用的全部成员发送 |
| toparty | 否 | 部门ID列表，多个接收者用‘|’分隔，最多支持100个。当touser为@all时忽略本参数 |
| totag | 否 | 标签ID列表，多个接收者用‘|’分隔，最多支持100个。当touser为@all时忽略本参数 |
| msgtype | 是 | 消息类型，此时固定为：voice |
| agentid | 是 | 企业应用的id，整型。可在应用的设置页面查看 |
| media\_id | 是 | 语音文件id，可以调用上传临时素材接口获取 |

#### 视频消息

**请求示例：**

1. {
2. "touser" : "UserID1|UserID2|UserID3",
3. "toparty" : " PartyID1|PartyID2 ",
4. "totag" : " TagID1 | TagID2 ",
5. "msgtype" : "video",
6. "agentid" : 1,
7. "video" : {
8. "media\_id" : "MEDIA\_ID",
9. "title" : "Title",
10. "description" : "Description"
11. }
12. }

**参数说明：**

| **参数** | **是否必须** | **说明** |
| --- | --- | --- |
| touser | 否 | 成员ID列表（消息接收者，多个接收者用‘|’分隔，最多支持1000个）。特殊情况：指定为@all，则向关注该企业应用的全部成员发送 |
| toparty | 否 | 部门ID列表，多个接收者用‘|’分隔，最多支持100个。当touser为@all时忽略本参数 |
| totag | 否 | 标签ID列表，多个接收者用‘|’分隔，最多支持100个。当touser为@all时忽略本参数 |
| msgtype | 是 | 消息类型，此时固定为：video |
| agentid | 是 | 企业应用的id，整型。可在应用的设置页面查看 |
| media\_id | 是 | 视频媒体文件id，可以调用上传临时素材接口获取 |
| title | 否 | 视频消息的标题，不超过128个字节，超过会自动截断 |
| description | 否 | 视频消息的描述，不超过512个字节，超过会自动截断 |

#### 文件消息

**请求示例：**

1. {
2. "touser" : "UserID1|UserID2|UserID3",
3. "toparty" : " PartyID1|PartyID2 ",
4. "totag" : " TagID1 | TagID2 ",
5. "msgtype" : "file",
6. "agentid" : 1,
7. "file" : {
8. "media\_id" : "1Yv-zXfHjSjU-7LH-GwtYqDGS-zz6w22KmWAT5COgP7o"
9. }
10. }

**参数说明：**

| **参数** | **是否必须** | **说明** |
| --- | --- | --- |
| touser | 否 | 成员ID列表（消息接收者，多个接收者用‘|’分隔，最多支持1000个）。特殊情况：指定为@all，则向关注该企业应用的全部成员发送 |
| toparty | 否 | 部门ID列表，多个接收者用‘|’分隔，最多支持100个。当touser为@all时忽略本参数 |
| totag | 否 | 标签ID列表，多个接收者用‘|’分隔，最多支持100个。当touser为@all时忽略本参数 |
| msgtype | 是 | 消息类型，此时固定为：file |
| agentid | 是 | 企业应用的id，整型。可在应用的设置页面查看 |
| media\_id | 是 | 文件id，可以调用上传临时素材接口获取 |

**文件消息展现：**  


#### 文本卡片消息

**请求示例：**

1. {
2. "touser" : "UserID1|UserID2|UserID3",
3. "toparty" : " PartyID1 | PartyID2 ",
4. "totag" : " TagID1 | TagID2 ",
5. "msgtype" : "textcard",
6. "agentid" : 1,
7. "textcard" : {
8. "title" : "领奖通知",
9. "description" : "<div class=\"gray\">2016年9月26日</div> <div class=\"normal\">恭喜你抽中iPhone 7一台，领奖码：xxxx</div><div class=\"highlight\">请于2016年10月10日前联系行政同事领取</div>",
10. "url" : "URL"
11. }
12. }

**参数说明：**

| **参数** | **是否必须** | **说明** |
| --- | --- | --- |
| touser | 否 | 成员ID列表（消息接收者，多个接收者用‘|’分隔，最多支持1000个）。特殊情况：指定为@all，则向关注该企业应用的全部成员发送 |
| toparty | 否 | 部门ID列表，多个接收者用‘|’分隔，最多支持100个。当touser为@all时忽略本参数 |
| totag | 否 | 标签ID列表，多个接收者用‘|’分隔，最多支持100个。当touser为@all时忽略本参数 |
| msgtype | 是 | 消息类型，此时固定为：textcard |
| agentid | 是 | 企业应用的id，整型。可在应用的设置页面查看 |
| title | 是 | 标题，不超过128个字节，超过会自动截断 |
| description | 是 | 描述，不超过512个字节，超过会自动截断 |
| url | 是 | 点击后跳转的链接。 |
| btntxt | 否 | 按钮文字。 默认为“详情”， 不超过4个文字，超过自动截断。 |

**文本卡片消息展现 ：**  


**特殊说明**：  
卡片消息的展现形式非常灵活，支持使用br标签或者空格来进行换行处理，也支持使用div标签来使用不同的字体颜色，目前内置了3种文字颜色：灰色(gray)、高亮(highlight)、默认黑色(normal)，将其作为div标签的class属性即可，具体用法请参考上面的示例。

#### 图文消息

**请求示例：**

1. {
2. "touser" : "UserID1|UserID2|UserID3",
3. "toparty" : " PartyID1 | PartyID2 ",
4. "totag" : " TagID1 | TagID2 ",
5. "msgtype" : "news",
6. "agentid" : 1,
7. "news" : {
8. "articles" : [
9. {
10. "title" : "中秋节礼品领取",
11. "description" : "今年中秋节公司有豪礼相送",
12. "url" : "URL",
13. "picurl" : "http://res.mail.qq.com/node/ww/wwopenmng/images/independent/doc/test\_pic\_msg1.png"
14. }
15. ]
16. }
17. }

**参数说明：**

| **参数** | **是否必须** | **说明** |
| --- | --- | --- |
| touser | 否 | 成员ID列表（消息接收者，多个接收者用‘|’分隔，最多支持1000个）。特殊情况：指定为@all，则向关注该企业应用的全部成员发送 |
| toparty | 否 | 部门ID列表，多个接收者用‘|’分隔，最多支持100个。当touser为@all时忽略本参数 |
| totag | 否 | 标签ID列表，多个接收者用‘|’分隔，最多支持100个。当touser为@all时忽略本参数 |
| msgtype | 是 | 消息类型，此时固定为：news |
| agentid | 是 | 企业应用的id，整型。可在应用的设置页面查看 |
| articles | 是 | 图文消息，一个图文消息支持1到8条图文 |
| title | 是 | 标题，不超过128个字节，超过会自动截断 |
| description | 否 | 描述，不超过512个字节，超过会自动截断 |
| url | 是 | 点击后跳转的链接。 |
| picurl | 否 | 图文消息的图片链接，支持JPG、PNG格式，较好的效果为大图640*320，小图80*80。 |
| btntxt | 否 | 按钮文字，仅在图文数为1条时才生效。 默认为“阅读全文”， 不超过4个文字，超过自动截断。 |

**图文消息展现：**  

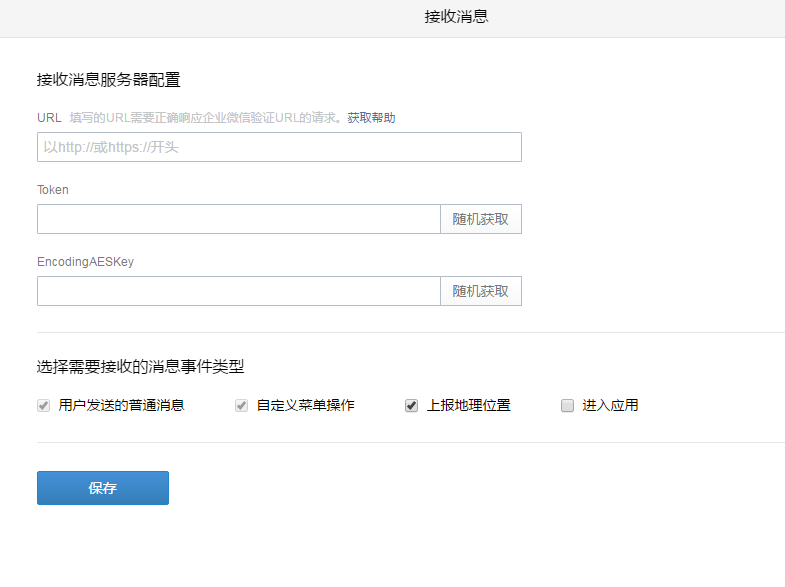

**版本信息：**  
v1.0以上版本有效

## 接收消息服务器配置

### 关于接收消息

企业微信为企业提供了开发自定义应用的能力，为了能够让自定义的应用和企业微信的后台进行双向通信，企业可以在应用的管理后台开启接收消息模式。开启接收消息模式后，企业可以配置需要获取的成员操作或者地理位置信息，当触发时企业微信会把对应的数据同步到企业的后台，企业后台也可以根据行为作出被动响应。

### 开启接收消息

1、设置接收消息的参数  
在企业的管理端后台，进入需要设置接收消息的目标应用，点击“接收消息”的“设置”，进入如下页面  
  
企业微信会要求你填写应用的URL、Token、EncodingAESKey三个参数，验证URL、Token以及加密的详细处理请参考[接收消息时的加解密处理][signature]章节。

* URL是企业应用接收企业微信推送请求的访问协议和地址，支持http或https协议。
* Token可由企业任意填写，用于生成签名。
* EncodingAESKey用于消息体的加密，是AES密钥的Base64编码。

2、验证URL有效性  
当点击“保存”提交以上信息时，企业微信将发送GET请求到填写的URL上，GET请求携带四个参数，企业在获取时需要做Urldecode处理，否则会验证不成功，你可以访问[接口调试工具][devtool]进行调试，依次选择 建立连接 > 接收消息。

* 参数说明

| **参数** | **必须** | **说明** |
| --- | --- | --- |
| msg\_signature | 是 | 企业微信加密签名，msg\_signature结合了企业填写的token、请求中的timestamp、nonce参数、加密的消息体 |
| timestamp | 是 | 时间戳 |
| nonce | 是 | 随机数 |
| echostr | 是 | 加密的随机字符串，以msg\_encrypt格式提供。需要解密并返回echostr明文，解密后有random、msg\_len、msg、$CorpID四个字段，其中msg即为echostr明文 |

企业通过参数msg\_signature对请求进行校验，如果确认此次GET请求来自企业微信，那么企业应该对echostr参数解密并原样返回echostr明文(不能加引号，不能带bom头，不能带换行符)，则接入验证生效，接收消息才能开启。  
后续推送消息给企业时都会在请求URL中带上以上参数（echostr除外），校验方式与首次验证URL一致。

### 使用接收消息

企业微信在推送消息给企业时，会对消息体本身做AES加密，以XML格式POST到企业应用的URL上。  
企业在被动响应时，也需要对数据加密，以XML格式返回给企业微信。企业的回复支持文本、图片、语音、视频、图文等格式。  
企业微信服务器在五秒内收不到响应会断掉连接，并且重新发起请求，总共重试三次。如果在调试中，发现成员无法收到响应的消息，可以检查是否消息处理超时。  
当接收成功后，http头部返回200表示接收ok，其他错误码一律当做失败并发起重试  
关于重试的消息排重，有msgid的消息推荐使用msgid排重。事件类型消息推荐使用FromUserName + CreateTime排重。  
假如企业无法保证在五秒内处理并回复，可以直接回复空串，企业微信不会对此作任何处理，并且不会发起重试。这种情况下，可以使用发消息接口进行异步回复。

假设企业的接收消息的URL设置为http://api.3dept.com。

**请求地址 ：**

http://api.3dept.com/?msg\_signature=ASDFQWEXZCVAQFASDFASDFSS&timestamp=13500001234&nonce=123412323

**接收数据格式 ：**

1. <xml>
2. <ToUserName><![CDATA[toUser]]</ToUserName>
3. <AgentID><![CDATA[toAgentID]]</AgentID>
4. <Encrypt><![CDATA[msg\_encrypt]]</Encrypt>
5. </xml>
6. 1.msg\_encrypt为经过加密的密文
7. 2.AgentID为接收的应用id，可在应用的设置页面获取
8. 3.ToUserName为企业微信的CorpID

企业需要对msg\_signature进行校验，并解密msg\_encrypt，得出msg的原文。

**被动响应给企业微信的数据格式：**

1. <xml>
2. <Encrypt><![CDATA[msg\_encrypt]]></Encrypt>
3. <MsgSignature><![CDATA[msg\_signature]]></MsgSignature>
4. <TimeStamp>timestamp</TimeStamp>
5. <Nonce><![CDATA[nonce]]></Nonce>
6. </xml>
7. 1.msg\_encrypt为经过加密的密文，算法参见附录
8. 2.MsgSignature为签名，算法参见附录
9. 3.TimeStamp为时间戳，Nonce为随机数，由企业自行生成

### 消息的加解密处理

企业可以直接使用微信提供的库进行加解密的处理，目前提供的有c++/python/php/java/c#等语言版本。代码提供了解密、加密、验证URL三个接口，企业可根据自身需要下载 （参见附录的加解密库下载）。以下为库函数的使用说明(以c++为例)，更详细的加解密方案请参考（附录）。

1、解密函数

1. int DecryptMsg(const string &sMsgSignature, const string &sTimeStamp, const string &sNonce, const string &sPostData, string &sMsg);

* 参数说明

| **参数** | **必须** | **说明** |
| --- | --- | --- |
| sMsgSignature | 是 | 从接收消息的URL中获取的msg\_signature参数 |
| sTimeStamp | 是 | 从接收消息的URL中获取的timestamp参数 |
| sNonce | 是 | 从接收消息的URL中获取的nonce参数 |
| sPostData | 是 | 从接收消息的URL中获取的整个post数据 |
| sMsg | 是 | 用于返回解密后的msg，以xml组织 |

* 返回说明

请参阅附录加解密部分。

2、加密函数

1. int EncryptMsg(const string &sReplyMsg, const string &sTimeStamp, const string &sNonce, string &sEncryptMsg);

* 参数说明

| **参数** | **必须** | **说明** |
| --- | --- | --- |
| sReplyMsg | 是 | 返回的消息体原文 |
| sTimeStamp | 是 | 时间戳，调用方生成 |
| sNonce | 是 | 随机数，调用方生成 |
| sEncryptMsg | 是 | 用于返回的密文，以xml组织 |

* 返回说明

请参阅附录加解密部分。

3、验证URL函数

1. int VerifyURL(const string &sMsgSignature, const string &sTimeStamp, const string &sNonce, const string &sEchoStr, string &sReplyEchoStr);

* 参数说明

| **参数** | **必须** | **说明** |
| --- | --- | --- |
| sMsgSignature | 是 | 从接收消息的URL中获取的msg\_signature参数 |
| sTimeStamp | 是 | 从接收消息的URL中获取的timestamp参数 |
| sNonce | 是 | 从接收消息的URL中获取的nonce参数 |
| sEchoStr | 是 | 从接收消息的URL中获取的echostr参数。注意，此参数必须是urldecode后的值 |
| sReplyEchoStr | 是 | 解密后的echostr，用于回包。注意，必须原样返回，不要做加引号或其它处理 |

* 返回说明

请参阅附录[加解密部分][algorithm]。

**版本信息：**  
v1.0以上版本有效

## 接收事件推送

### 接收事件推送

当企业成员通过企业微信APP或微信插件触发进入应用、上报地理位置、点击菜单等事件时，企业微信会通过接收消息的方式将这些事件通知企业应用的后台。使用此功能需开启接收消息模式，关于如何设置接收消息请参考接收消息设置。

如接受消息地址设置为：http://api.3dept.com/，企业微信将按照如下方式给该地址发送请求：

**请求方式：**POST  
**请求地址：**http://api.3dept.com/?msg\_signature=ASDFQWEXZCVAQFASDFASDFSS&timestamp=13500001234&nonce=123412323

### 事件类型

#### 成员关注/取消关注事件

本事件在成员关注（取消关注）应用时触发

请求示例：

1. <xml>
2. <ToUserName><![CDATA[toUser]]></ToUserName>
3. <FromUserName><![CDATA[UserID]]></FromUserName>
4. <CreateTime>1348831860</CreateTime>
5. <MsgType><![CDATA[event]]></MsgType>
6. <Event><![CDATA[subscribe]]></Event>
7. <AgentID>1</AgentID>
8. </xml>

参数说明：

| **参数** | **说明** |
| --- | --- |
| ToUserName | 企业微信CorpID |
| FromUserName | 成员UserID |
| CreateTime | 消息创建时间（整型） |
| MsgType | 消息类型，此时固定为：event |
| Event | 事件类型，subscribe(订阅)、unsubscribe(取消订阅) |
| EventKey | 事件KEY值，此事件该值为空 |
| AgentID | 企业应用的id，整型。可在应用的设置页面查看 |

#### 进入应用

本事件在成员进入企业微信的应用时触发

请求示例：

1. <xml>
2. <toUserName><![CDATA[toUser]]></ToUserName>
3. <FromUserName><![CDATA[FromUser]]></FromUserName>
4. <CreateTime>1408091189</CreateTime>
5. <MsgType><![CDATA[event]]></MsgType>
6. <Event><![CDATA[enter\_agent]]></Event>
7. <EventKey><![CDATA[]]></EventKey>
8. <AgentID>1</AgentID>
9. </xml>

参数说明：

| **参数** | **说明** |
| --- | --- |
| ToUserName | 企业微信CorpID |
| FromUserName | 成员UserID |
| CreateTime | 消息创建时间（整型） |
| MsgType | 消息类型，此时固定为：event |
| Event | 事件类型：enter\_agent |
| EventKey | 事件KEY值，此事件该值为空 |
| AgentID | 企业应用的id，整型。可在应用的设置页面查看 |

#### 上报地理位置

成员同意上报地理位置后，每次在进入应用会话时都会上报一次地理位置，或在进入应用会话后每5秒上报一次地理位置。  
企业可以在管理端修改应用是否需要获取地理位置权限。

请求示例：

1. <xml>
2. <ToUserName><![CDATA[toUser]]></ToUserName>
3. <FromUserName><![CDATA[FromUser]]></FromUserName>
4. <CreateTime>123456789</CreateTime>
5. <MsgType><![CDATA[event]]></MsgType>
6. <Event><![CDATA[LOCATION]]></Event>
7. <Latitude>23.104105</Latitude>
8. <Longitude>113.320107</Longitude>
9. <Precision>65.000000</Precision>
10. <AgentID>1</AgentID>
11. </xml>

参数说明：

| **参数** | **说明** |
| --- | --- |
| ToUserName | 企业微信CorpID |
| FromUserName | 成员UserID |
| CreateTime | 消息创建时间（整型） |
| MsgType | 消息类型，此时固定为：event |
| Event | 事件类型：LOCATION |
| Latitude | 地理位置纬度 |
| Longitude | 地理位置经度 |
| Precision | 地理位置精度 |
| AgentID | 企业应用的id，整型。可在应用的设置页面查看 |

#### 异步任务完成事件推送

本事件是成员在使用异步任务接口时，用于接收任务执行完毕的结果通知。

请求示例：

1. <xml>
2. <ToUserName><![CDATA[wx28dbb14e37208abe]]></ToUserName>
3. <FromUserName><![CDATA[FromUser]]></FromUserName>
4. <CreateTime>1425284517</CreateTime>
5. <MsgType><![CDATA[event]]></MsgType>
6. <Event><![CDATA[batch\_job\_result]]></Event>
7. <BatchJob><JobId><![CDATA[S0MrnndvRG5fadSlLwiBqiDDbM143UqTmKP3152FZk4]]></JobId>
8. <JobType><![CDATA[sync\_user]]></JobType>
9. <ErrCode>0</ErrCode>
10. <ErrMsg><![CDATA[ok]]></ErrMsg>
11. </BatchJob>
12. </xml>

参数说明：

| **参数** | **说明** |
| --- | --- |
| ToUserName | 企业微信CorpID |
| FromUserName | 成员UserID |
| CreateTime | 消息创建时间（整型） |
| MsgType | 消息类型，此时固定为：event |
| Event | 事件类型：batch\_job\_result |
| JobId | 异步任务id，最大长度为64字符 |
| JobType | 操作类型，字符串，目前分别有：sync\_user(增量更新成员)、 replace\_user(全量覆盖成员）、invite\_user(邀请成员关注）、replace\_party(全量覆盖部门) |
| ErrCode | 返回码 |
| ErrMsg | 对返回码的文本描述内容 |

#### 通讯录变更事件

企业微信的成员可在客户端变更自己的个人信息。当企业通过通讯录助手开通通讯录写权限后，成员的变更会通知给企业。变更的事件，将推送到企业微信管理端通讯录助手中的‘接收事件服务器’。

请求示例：

1. <xml>
2. <ToUserName><![CDATA[toUser]]></ToUserName>
3. <FromUserName><![CDATA[fromUser]]></FromUserName>
4. <CreateTime>1403610513</CreateTime>
5. <MsgType><![CDATA[event]]></MsgType>
6. <Event><![CDATA[change\_contact]]></Event>
7. <ChangeType>update\_user</ChangeType>
8. <UserID><![CDATA[zhangsan]]></UserID>
9. <Name><![CDATA[张三]]></Name>
10. <Mobile>15913215421</Mobile>
11. <Gender>1</Gender>
12. <Email><![CDATA[zhangsan@gzdev.com]]></Email>
13. <Avatar><![CDATA[http://wx.qlogo.cn/mmopen/ajNVdqHZLLA3WJ6DSZUfiakYe37PKnQhBIeOQBO4czqrnZDS79FH5Wm5m4X69TBicnHFlhiafvDwklOpZeXYQQ2icg/0]]></Avatar>
14. <ExtAttr>
15. <Item>
16. <Name><![CDATA[爱好]]></Name>
17. <Value><![CDATA[旅游]]></Value>
18. </Item>
19. <Item>
20. <Name><![CDATA[卡号]]></Name>
21. <Value><![CDATA[1234567234]]></Value>
22. </Item>
23. </ExtAttr>
24. </xml>

参数说明：

| **参数** | **说明** |
| --- | --- |
| ToUserName | 企业微信CorpID |
| FromUserName | 此事件该值固定为sys，表示该消息由系统生成 |
| CreateTime | 消息创建时间 （整型） |
| MsgType | 消息的类型，此时固定为event |
| Event | 事件的类型，此时固定为change\_contact |
| ChangeType | 此时固定为update\_user |
| UserID | 变更信息的成员UserID |
| Name | 成员名称，变更时推送 |
| Mobile | 手机号码，变更时推送 |
| Gender | 性别，变更时推送。1表示男性，2表示女性 |
| Email | 邮箱，变更时推送 |
| ExtAttr | 扩展属性，变更时推送 |

#### 菜单事件

成员点击自定义菜单后，企业微信会把点击事件推送给应用。  
点击菜单弹出子菜单，不会产生上报。  
企业微信iPhone1.2.2/Android1.2.2版本开始支持菜单事件，旧版本企业微信成员点击后将没有回应，应用不能正常接收到事件推送。  
自定义菜单可以在管理后台的应用设置界面配置。

##### 点击菜单拉取消息的事件推送

1. <xml>
2. <ToUserName><![CDATA[toUser]]></ToUserName>
3. <FromUserName><![CDATA[FromUser]]></FromUserName>
4. <CreateTime>123456789</CreateTime>
5. <MsgType><![CDATA[event]]></MsgType>
6. <Event><![CDATA[click]]></Event>
7. <EventKey><![CDATA[EVENTKEY]]></EventKey>
8. <AgentID>1</AgentID>
9. </xml>

参数说明：

| **参数** | **说明** |
| --- | --- |
| ToUserName | 企业微信CorpID |
| FromUserName | 成员UserID |
| CreateTime | 消息创建时间（整型） |
| MsgType | 消息类型，此时固定为：event |
| Event | 事件类型：click |
| EventKey | 事件KEY值，与自定义菜单接口中KEY值对应 |
| AgentID | 企业应用的id，整型。可在应用的设置页面查看 |

##### 点击菜单跳转链接的事件推送

1. <xml>
2. <ToUserName><![CDATA[toUser]]></ToUserName>
3. <FromUserName><![CDATA[FromUser]]></FromUserName>
4. <CreateTime>123456789</CreateTime>
5. <MsgType><![CDATA[event]]></MsgType>
6. <Event><![CDATA[view]]></Event>
7. <EventKey><![CDATA[www.qq.com]]></EventKey>
8. <AgentID>1</AgentID>
9. </xml>

参数说明：

| 参数 | 说明 |
| --- | --- |
| ToUserName | 企业微信CorpID |
| FromUserName | 成员UserID |
| CreateTime | 消息创建时间（整型） |
| MsgType | 消息类型，此时固定为：event |
| Event | 事件类型：view |
| EventKey | 事件KEY值，设置的跳转URL |
| AgentID | 企业应用的id，整型。可在应用的设置页面查看 |

##### 扫码推事件的事件推送

1. <xml><ToUserName><![CDATA[toUser]]></ToUserName>
2. <FromUserName><![CDATA[FromUser]]></FromUserName>
3. <CreateTime>1408090502</CreateTime>
4. <MsgType><![CDATA[event]]></MsgType>
5. <Event><![CDATA[scancode\_push]]></Event>
6. <EventKey><![CDATA[6]]></EventKey>
7. <ScanCodeInfo><ScanType><![CDATA[qrcode]]></ScanType>
8. <ScanResult><![CDATA[1]]></ScanResult>
9. </ScanCodeInfo>
10. <AgentID>1</AgentID>
11. </xml>

参数说明：

| **参数** | **说明** |
| --- | --- |
| ToUserName | 企业微信CorpID |
| FromUserName | 成员UserID |
| CreateTime | 消息创建时间（整型） |
| MsgType | 消息类型，此时固定为：event |
| Event | 事件类型：scancode\_push |
| EventKey | 事件KEY值，与自定义菜单接口中KEY值对应 |
| ScanCodeInfo | 扫描信息 |
| ScanType | 扫描类型，一般是qrcode |
| ScanResult | 扫描结果，即二维码对应的字符串信息 |
| AgentID | 企业应用的id，整型。可在应用的设置页面查看 |

##### 扫码推事件且弹出“消息接收中”提示框的事件推送

1. <xml><ToUserName><![CDATA[toUser]]></ToUserName>
2. <FromUserName><![CDATA[FromUser]]></FromUserName>
3. <CreateTime>1408090606</CreateTime>
4. <MsgType><![CDATA[event]]></MsgType>
5. <Event><![CDATA[scancode\_waitmsg]]></Event>
6. <EventKey><![CDATA[6]]></EventKey>
7. <ScanCodeInfo><ScanType><![CDATA[qrcode]]></ScanType>
8. <ScanResult><![CDATA[2]]></ScanResult>
9. </ScanCodeInfo>
10. <AgentID>1</AgentID>
11. </xml>

参数说明：

| **参数** | **说明** |
| --- | --- |
| ToUserName | 企业微信CorpID |
| FromUserName | 成员UserID |
| CreateTime | 消息创建时间（整型） |
| MsgType | 消息类型，此时固定为：event |
| Event | 事件类型：scancode\_waitmsg |
| EventKey | 事件KEY值，与自定义菜单接口中KEY值对应 |
| ScanCodeInfo | 扫描信息 |
| ScanType | 扫描类型，一般是qrcode |
| ScanResult | 扫描结果，即二维码对应的字符串信息 |
| AgentID | 企业应用的id，整型。可在应用的设置页面查看 |

##### 弹出系统拍照发图的事件推送

1. <xml><ToUserName><![CDATA[toUser]]></ToUserName>
2. <FromUserName><![CDATA[FromUser]]></FromUserName>
3. <CreateTime>1408090651</CreateTime>
4. <MsgType><![CDATA[event]]></MsgType>
5. <Event><![CDATA[pic\_sysphoto]]></Event>
6. <EventKey><![CDATA[6]]></EventKey>
7. <SendPicsInfo><Count>1</Count>
8. <PicList><item><PicMd5Sum><![CDATA[1b5f7c23b5bf75682a53e7b6d163e185]]></PicMd5Sum>
9. </item>
10. </PicList>
11. </SendPicsInfo>
12. <AgentID>1</AgentID>
13. </xml>

参数说明：

| **参数** | **说明** |
| --- | --- |
| ToUserName | 企业微信CorpID |
| FromUserName | 成员UserID |
| CreateTime | 消息创建时间（整型） |
| MsgType | 消息类型，此时固定为：event |
| Event | 事件类型：pic\_sysphoto |
| EventKey | 事件KEY值，与自定义菜单接口中KEY值对应 |
| ScanCodeInfo | 发送的图片信息 |
| Count | 发送的图片数量 |
| PicList | 图片列表 |
| PicMd5Sum | 图片的MD5值，开发者若需要，可用于验证接收到图片 |
| AgentID | 企业应用的id，整型。可在应用的设置页面查看 |

##### 弹出拍照或者相册发图的事件推送

1. <xml><ToUserName><![CDATA[toUser]]></ToUserName>
2. <FromUserName><![CDATA[FromUser]]></FromUserName>
3. <CreateTime>1408090816</CreateTime>
4. <MsgType><![CDATA[event]]></MsgType>
5. <Event><![CDATA[pic\_photo\_or\_album]]></Event>
6. <EventKey><![CDATA[6]]></EventKey>
7. <SendPicsInfo><Count>1</Count>
8. <PicList><item><PicMd5Sum><![CDATA[5a75aaca956d97be686719218f275c6b]]></PicMd5Sum>
9. </item>
10. </PicList>
11. </SendPicsInfo>
12. <AgentID>1</AgentID>
13. </xml>

参数说明：

| **参数** | **说明** |
| --- | --- |
| ToUserName | 企业微信CorpID |
| FromUserName | 成员UserID |
| CreateTime | 消息创建时间（整型） |
| MsgType | 消息类型，此时固定为：event |
| Event | 事件类型：pic\_photo\_or\_album |
| EventKey | 事件KEY值，与自定义菜单接口中KEY值对应 |
| ScanCodeInfo | 发送的图片信息 |
| Count | 发送的图片数量 |
| PicList | 图片列表 |
| PicMd5Sum | 图片的MD5值，开发者若需要，可用于验证接收到图片 |
| AgentID | 企业应用的id，整型。可在应用的设置页面查看 |

##### 弹出微信相册发图器的事件推送

1. <xml><ToUserName><![CDATA[toUser]]></ToUserName>
2. <FromUserName><![CDATA[FromUser]]></FromUserName>
3. <CreateTime>1408090816</CreateTime>
4. <MsgType><![CDATA[event]]></MsgType>
5. <Event><![CDATA[pic\_weixin]]></Event>
6. <EventKey><![CDATA[6]]></EventKey>
7. <SendPicsInfo><Count>1</Count>
8. <PicList><item><PicMd5Sum><![CDATA[5a75aaca956d97be686719218f275c6b]]></PicMd5Sum>
9. </item>
10. </PicList>
11. </SendPicsInfo>
12. <AgentID>1</AgentID>
13. </xml>

参数说明：

| **参数** | **说明** |
| --- | --- |
| ToUserName | 企业微信CorpID |
| FromUserName | 成员UserID |
| CreateTime | 消息创建时间（整型） |
| MsgType | 消息类型，此时固定为：event |
| Event | 事件类型：pic\_weixin |
| EventKey | 事件KEY值，与自定义菜单接口中KEY值对应 |
| ScanCodeInfo | 发送的图片信息 |
| Count | 发送的图片数量 |
| PicList | 图片列表 |
| PicMd5Sum | 图片的MD5值，开发者若需要，可用于验证接收到图片 |
| AgentID | 企业应用的id，整型。可在应用的设置页面查看 |

##### 弹出地理位置选择器的事件推送

请求示例：

1. <xml><ToUserName><![CDATA[toUser]]></ToUserName>
2. <FromUserName><![CDATA[FromUser]]></FromUserName>
3. <CreateTime>1408091189</CreateTime>
4. <MsgType><![CDATA[event]]></MsgType>
5. <Event><![CDATA[location\_select]]></Event>
6. <EventKey><![CDATA[6]]></EventKey>
7. <SendLocationInfo><Location\_X><![CDATA[23]]></Location\_X>
8. <Location\_Y><![CDATA[113]]></Location\_Y>
9. <Scale><![CDATA[15]]></Scale>
10. <Label><![CDATA[ 广州市海珠区客村艺苑路 106号]]></Label>
11. <Poiname><![CDATA[]]></Poiname>
12. </SendLocationInfo>
13. <AgentID>1</AgentID>
14. </xml>

参数说明：

| **参数** | **说明** |
| --- | --- |
| ToUserName | 企业微信CorpID |
| FromUserName | 成员UserID |
| CreateTime | 消息创建时间（整型） |
| MsgType | 消息类型，此时固定为：event |
| Event | 事件类型：location\_select |
| EventKey | 事件KEY值，与自定义菜单接口中KEY值对应 |
| SendLocationInfo | 发送的位置信息 |
| Location\_X | X坐标信息 |
| Location\_Y | Y坐标信息 |
| Scale | 精度，可理解为精度或者比例尺、越精细的话 scale越高 |
| Label | 地理位置的字符串信息 |
| Poiname | POI的名字，可能为空 |
| AgentID | 企业应用的id，整型。可在应用的设置页面查看 |

**版本信息：**  
v1.0以上版本有效

## 接收普通消息

### 接收普通消息

当企业成员给企业微信应用发送消息时，企业微信会通过“接收消息”接口将消息转发到企业应用的后台，消息类型支持：文本、图片、语音、视频、位置以及链接信息。关于如何回调模式，请参考接收消息设置。

如果回调地址设置为：http://api.3dept.com/，企业微信将按照如下方式给该地址发送请求：

**请求方式：**POST  
**请求地址：**http://api.3dept.com/?msg\_signature=ASDFQWEXZCVAQFASDFASDFSS&timestamp=13500001234&nonce=123412323

### 消息类型

#### 文本消息

**消息示例：**

1. <xml>
2. <ToUserName><![CDATA[toUser]]></ToUserName>
3. <FromUserName><![CDATA[fromUser]]></FromUserName>
4. <CreateTime>1348831860</CreateTime>
5. <MsgType><![CDATA[text]]></MsgType>
6. <Content><![CDATA[this is a test]]></Content>
7. <MsgId>1234567890123456</MsgId>
8. <AgentID>1</AgentID>
9. </xml>

**参数说明：**

| **参数** | **说明** |
| --- | --- |
| ToUserName | 企业微信CorpID |
| FromUserName | 成员UserID |
| CreateTime | 消息创建时间（整型） |
| MsgType | 消息类型，此时固定为：text |
| Content | 文本消息内容 |
| MsgId | 消息id，64位整型 |
| AgentID | 企业应用的id，整型。可在应用的设置页面查看 |

#### 图片消息

**消息示例：**

1. <xml>
2. <ToUserName><![CDATA[toUser]]></ToUserName>
3. <FromUserName><![CDATA[fromUser]]></FromUserName>
4. <CreateTime>1348831860</CreateTime>
5. <MsgType><![CDATA[image]]></MsgType>
6. <PicUrl><![CDATA[this is a url]]></PicUrl>
7. <MediaId><![CDATA[media\_id]]></MediaId>
8. <MsgId>1234567890123456</MsgId>
9. <AgentID>1</AgentID>
10. </xml>

**参数说明：**

| **参数** | **说明** |
| --- | --- |
| ToUserName | 企业微信CorpID |
| FromUserName | 成员UserID |
| CreateTime | 消息创建时间（整型） |
| MsgType | 消息类型，此时固定为：image |
| PicUrl | 图片链接 |
| MediaId | 图片媒体文件id，可以调用获取媒体文件接口拉取 |
| MsgId | 消息id，64位整型 |
| AgentID | 企业应用的id，整型。可在应用的设置页面查看 |

#### 语音消息

**消息示例：**

1. <xml>
2. <ToUserName><![CDATA[toUser]]></ToUserName>
3. <FromUserName><![CDATA[fromUser]]></FromUserName>
4. <CreateTime>1357290913</CreateTime>
5. <MsgType><![CDATA[voice]]></MsgType>
6. <MediaId><![CDATA[media\_id]]></MediaId>
7. <Format><![CDATA[Format]]></Format>
8. <MsgId>1234567890123456</MsgId>
9. <AgentID>1</AgentID>
10. </xml>

**参数说明：**

| **参数** | **说明** |
| --- | --- |
| ToUserName | 企业微信CorpID |
| FromUserName | 成员UserID |
| CreateTime | 消息创建时间（整型） |
| MsgType | 消息类型，此时固定为：voice |
| MediaId | 语音媒体文件id，可以调用获取媒体文件接口拉取数据 |
| Format | 语音格式，如amr，speex等 |
| MsgId | 消息id，64位整型 |
| AgentID | 企业应用的id，整型。可在应用的设置页面查看 |

#### 视频消息

**消息示例：**

1. <xml>
2. <ToUserName><![CDATA[toUser]]></ToUserName>
3. <FromUserName><![CDATA[fromUser]]></FromUserName>
4. <CreateTime>1357290913</CreateTime>
5. <MsgType><![CDATA[video]]></MsgType>
6. <MediaId><![CDATA[media\_id]]></MediaId>
7. <ThumbMediaId><![CDATA[thumb\_media\_id]]></ThumbMediaId>
8. <MsgId>1234567890123456</MsgId>
9. <AgentID>1</AgentID>
10. </xml>

**参数说明：**

| **参数** | **说明** |
| --- | --- |
| ToUserName | 企业微信CorpID |
| FromUserName | 成员UserID |
| CreateTime | 消息创建时间（整型） |
| MsgType | 消息类型，此时固定为：video |
| MediaId | 视频媒体文件id，可以调用获取媒体文件接口拉取数据 |
| ThumbMediaId | 视频消息缩略图的媒体id，可以调用获取媒体文件接口拉取数据 |
| MsgId | 消息id，64位整型 |
| AgentID | 企业应用的id，整型。可在应用的设置页面查看 |

#### 位置消息

**消息示例：**

1. <xml>
2. <ToUserName><![CDATA[toUser]]></ToUserName>
3. <FromUserName><![CDATA[fromUser]]></FromUserName>
4. <CreateTime>1351776360</CreateTime>
5. <MsgType><![CDATA[location]]></MsgType>
6. <Location\_X>23.134521</Location\_X>
7. <Location\_Y>113.358803</Location\_Y>
8. <Scale>20</Scale>
9. <Label><![CDATA[位置信息]]></Label>
10. <MsgId>1234567890123456</MsgId>
11. <AgentID>1</AgentID>
12. </xml>

**参数说明：**

| **参数** | **说明** |
| --- | --- |
| ToUserName | 企业微信CorpID |
| FromUserName | 成员UserID |
| CreateTime | 消息创建时间（整型） |
| MsgType | 消息类型，此时固定为： location |
| Location\_X | 地理位置纬度 |
| Location\_Y | 地理位置经度 |
| Scale | 地图缩放大小 |
| Label | 地理位置信息 |
| MsgId | 消息id，64位整型 |
| AgentID | 企业应用的id，整型。可在应用的设置页面查看 |

#### 链接消息

**消息示例：**

1. <xml>
2. <ToUserName><![CDATA[toUser]]></ToUserName>
3. <FromUserName><![CDATA[fromUser]]></FromUserName>
4. <CreateTime>1348831860</CreateTime>
5. <MsgType><![CDATA[link]]></MsgType>
6. <Title><![CDATA[this is a title！]]></Title>
7. <Description><![CDATA[this is a description！]]></Description>
8. <PicUrl><![CDATA[this is a url]]></PicUrl>
9. <MsgId>1234567890123456</MsgId>
10. <AgentID>1</AgentID>
11. </xml>

**参数说明：**

| **参数** | **说明** |
| --- | --- |
| ToUserName | 企业微信CorpID |
| FromUserName | 成员UserID |
| CreateTime | 消息创建时间（整型） |
| MsgType | 消息类型，此时固定为：link |
| Title | 标题 |
| Description | 描述 |
| PicUrl | 封面缩略图的url |
| MsgId | 消息id，64位整型 |
| AgentID | 企业应用的id，整型。可在应用的设置页面查看 |

**版本信息：**  
v1.0以上版本有效

## 被动回复消息

### 被动响应

* 当企业后台收到推送过来的普通消息、菜单click事件、成员进入应用等事件后，可以进行被动响应
* 企业响应的消息同样应该经过加密，并带上msg\_signature、timestamp、nonce及密文，其中timestamp、nonce由企业指定，msg\_signature、密文经特定算法生成，具体算法参见附录。

标准的回包：

1. <xml>
2. <Encrypt><![CDATA[msg\_encrypt]]></Encrypt>
3. <MsgSignature><![CDATA[msg\_signature]]></MsgSignature>
4. <TimeStamp>timestamp</TimeStamp>
5. <Nonce><![CDATA[nonce]]></Nonce>
6. </xml>

### 消息类型

#### 文本消息

明文XML结构：

1. <xml>
2. <ToUserName><![CDATA[toUser]]></ToUserName>
3. <FromUserName><![CDATA[fromUser]]></FromUserName>
4. <CreateTime>1348831860</CreateTime>
5. <MsgType><![CDATA[text]]></MsgType>
6. <Content><![CDATA[this is a test]]></Content>
7. </xml>

参数说明：

| **参数** | **说明** |
| --- | --- |
| ToUserName | 成员UserID |
| FromUserName | 企业微信CorpID |
| CreateTime | 消息创建时间（整型） |
| MsgType | 消息类型，此时固定为：text |
| Content | 文本消息内容 |

#### 图片消息

明文XML结构：

1. <xml>
2. <ToUserName><![CDATA[toUser]]></ToUserName>
3. <FromUserName><![CDATA[fromUser]]></FromUserName>
4. <CreateTime>1348831860</CreateTime>
5. <MsgType><![CDATA[image]]></MsgType>
6. <Image>
7. <MediaId><![CDATA[media\_id]]></MediaId>
8. </Image>
9. </xml>

参数说明：

| **参数** | **说明** |
| --- | --- |
| ToUserName | 成员UserID |
| FromUserName | 企业微信CorpID |
| CreateTime | 消息创建时间（整型） |
| MsgType | 消息类型，此时固定为：image |
| MediaId | 图片媒体文件id，可以调用获取媒体文件接口拉取 |

#### 语音消息

明文XML结构：

1. <xml>
2. <ToUserName><![CDATA[toUser]]></ToUserName>
3. <FromUserName><![CDATA[fromUser]]></FromUserName>
4. <CreateTime>1357290913</CreateTime>
5. <MsgType><![CDATA[voice]]></MsgType>
6. <Voice>
7. <MediaId><![CDATA[media\_id]]></MediaId>
8. </Voice>
9. </xml>

参数说明：

| **参数** | **说明** |
| --- | --- |
| ToUserName | 成员UserID |
| FromUserName | 企业微信CorpID |
| CreateTime | 消息创建时间（整型） |
| MsgType | 消息类型，此时固定为：voice |
| MediaId | 语音文件id，可以调用获取媒体文件接口拉取 |

#### 视频消息

明文XML结构：

1. <xml>
2. <ToUserName><![CDATA[toUser]]></ToUserName>
3. <FromUserName><![CDATA[fromUser]]></FromUserName>
4. <CreateTime>1357290913</CreateTime>
5. <MsgType><![CDATA[video]]></MsgType>
6. <Video>
7. <MediaId><![CDATA[media\_id]]></MediaId>
8. <Title><![CDATA[title]]></Title>
9. <Description><![CDATA[description]]></Description>
10. </Video>
11. </xml>

参数说明：

| **参数** | **说明** |
| --- | --- |
| ToUserName | 成员UserID |
| FromUserName | 企业微信CorpID |
| CreateTime | 消息创建时间（整型） |
| MsgType | 消息类型，此时固定为：video |
| MediaId | 视频文件id，可以调用获取媒体文件接口拉取 |
| Title | 视频消息的标题 |
| Description | 视频消息的描述 |

#### 图文消息

明文XML结构：

1. <xml>
2. <ToUserName><![CDATA[toUser]]></ToUserName>
3. <FromUserName><![CDATA[fromUser]]></FromUserName>
4. <CreateTime>12345678</CreateTime>
5. <MsgType><![CDATA[news]]></MsgType>
6. <ArticleCount>2</ArticleCount>
7. <Articles>
8. <item>
9. <Title><![CDATA[title1]]></Title>
10. <Description><![CDATA[description1]]></Description>
11. <PicUrl><![CDATA[picurl]]></PicUrl>
12. <Url><![CDATA[url]]></Url>
13. </item>
14. <item>
15. <Title><![CDATA[title]]></Title>
16. <Description><![CDATA[description]]></Description>
17. <PicUrl><![CDATA[picurl]]></PicUrl>
18. <Url><![CDATA[url]]></Url>
19. </item>
20. </Articles>
21. </xml>

参数说明：

| **参数** | **说明** |
| --- | --- |
| ToUserName | 成员UserID |
| FromUserName | 企业微信CorpID |
| CreateTime | 消息创建时间（整型） |
| MsgType | 消息类型，此时固定为：news |
| ArticleCount | 图文消息的数量 |
| Title | 标题，不超过128个字节，超过会自动截断 |
| Description | 描述，不超过512个字节，超过会自动截断 |
| Url | 点击后跳转的链接。 |
| PicUrl | 图文消息的图片链接，支持JPG、PNG格式，较好的效果为大图640*320，小图80*80。 |

**版本信息：**  
v1.0以上版本有效

# 素材管理

## 上传临时素材文件

**请求方式：**POST  
**请求地址：**https://saastest2.gov.weixin.qq.com/cgi-bin/media/upload?access\_token=ACCESS\_TOKEN&type=TYPE

使用http multipart/form-data上传文件， 文件标识名为”media”.  
**参数说明：**

| **参数** | **必须** | **说明** |
| --- | --- | --- |
| access\_token | 是 | 调用接口凭证 |
| type | 是 | 媒体文件类型，分别有图片（image）、语音（voice）、视频（video），普通文件(file) |

**权限说明：**

完全公开，media\_id在同一单位内应用之间可以共享

**返回数据：**

1. {
2. "errcode": 0,
3. "errmsg": ""，
4. "type": "image",
5. "media\_id": "1G6nrLmr5EC3MMb\_-zK1dDdzmd0p7cNliYu9V5w7o8K0",
6. "created\_at": "1380000000"
7. }

**参数说明：**

| **参数** | **说明** |
| --- | --- |
| type | 媒体文件类型，分别有图片（image）、语音（voice）、视频（video），普通文件(file) |
| media\_id | 媒体文件上传后获取的唯一标识 |
| created\_at | 媒体文件上传时间戳 |

**上传的媒体文件限制 ：**  
所有文件size必须大于5个字节

* 图片（image）：2MB，支持JPG,PNG格式
* 语音（voice） ：2MB，播放长度不超过60s，支持AMR格式
* 视频（video） ：10MB，支持MP4格式
* 普通文件（file）：20MB
* **版本信息：**  
  v1.0以上版本有效

## 获取临时素材文件

**请求方式：**GET  
**请求地址：**https://saastest2.gov.weixin.qq.com/cgi-bin/media/get?access\_token=ACCESS\_TOKEN&media\_id=MEDIA\_ID

**参数说明 ：**

| 参数 | 必须 | 说明 |
| --- | --- | --- |
| access\_token | 是 | 调用接口凭证 |
| media\_id | 是 | 媒体文件id |

**权限说明：**  
完全公开，media\_id在同一单位内所有应用之间可以共享。

**返回说明 ：**  
正确时返回（和普通的http下载相同，请根据http头做相应的处理）：

1. {
2. HTTP/1.1 200 OK
3. Connection: close
4. Content-Type: image/jpeg
5. Content-disposition: attachment; filename="MEDIA\_ID.jpg"
6. Date: Sun, 06 Jan 2013 10:20:18 GMT
7. Cache-Control: no-cache, must-revalidate
8. Content-Length: 339721
9. Xxxx
10. }

错误时返回（这里省略了HTTP首部）：

1. {
2. "errcode": 40007,
3. "errmsg": "invalid media\_id"
4. }

**版本信息：**  
v1.0以上版本有效

# 网页开发



## JS-SDK

### 概述

企业微信JS-SDK是企业微信面向网页开发者提供的基于企业微信内的网页开发工具包。

通过使用企业微信JS-SDK，网页开发者可借助企业微信高效地使用拍照、选图、语音、位置等手机系统的能力，同时可以直接使用企业微信分享、扫一扫等企业微信特有的能力，为企业微信用户提供更优质的网页体验。

此文档面向网页开发者介绍企业微信JS-SDK如何使用及相关注意事项。

### 使用说明

**注意： 所有的JS接口只能在企业微信应用的可信域名下调用(包括子域名)，可在企业微信的管理后台“我的应用”里设置应用可信域名。**

#### 步骤一：引入JS文件

在需要调用JS接口的页面引入如下JS文件，（支持https）：http://res.wx.qq.com/open/js/jweixin-1.2.0.js

为支持微信插件功能，请引用此文件。原企业微信的js文件在微信插件侧不生效。

备注：支持使用 AMD/CMD 标准模块加载方法加载

#### 步骤二：通过config接口注入权限验证配置

**所有需要使用JS-SDK的页面必须先注入配置信息，否则将无法调用（同一个url仅需调用一次，对于变化url的SPA的web app可在每次url变化时进行调用）。**

1. wx.config({
2. debug: true, // 开启调试模式,调用的所有api的返回值会在客户端alert出来，若要查看传入的参数，可以在pc端打开，参数信息会通过log打出，仅在pc端时才会打印。
3. appId: '', // 必填，企业微信的cropID
4. timestamp: , // 必填，生成签名的时间戳
5. nonceStr: '', // 必填，生成签名的随机串
6. signature: '',// 必填，签名，见附录1
7. jsApiList: [] // 必填，需要使用的JS接口列表，所有JS接口列表见附录2
8. });

#### 步骤三：通过ready接口处理成功验证

1. wx.ready(function(){
2. // config信息验证后会执行ready方法，所有接口调用都必须在config接口获得结果之后，config是一个客户端的异步操作，所以如果需要在页面加载时就调用相关接口，则须把相关接口放在ready函数中调用来确保正确执行。对于用户触发时才调用的接口，则可以直接调用，不需要放在ready函数中。
3. });

#### 步骤四：通过error接口处理失败验证

1. wx.error(function(res){
2. // config信息验证失败会执行error函数，如签名过期导致验证失败，具体错误信息可以打开config的debug模式查看，也可以在返回的res参数中查看，对于SPA可以在这里更新签名。
3. });

#### 接口调用说明

所有接口通过wx对象(也可使用jWeixin对象)来调用，参数是一个对象，除了每个接口本身需要传的参数之外，还有以下通用参数：

1. success：接口调用成功时执行的回调函数。
2. fail：接口调用失败时执行的回调函数。
3. complete：接口调用完成时执行的回调函数，无论成功或失败都会执行。
4. cancel：用户点击取消时的回调函数，仅部分有用户取消操作的api才会用到
5. trigger: 监听Menu中的按钮点击时触发的方法，该方法仅支持Menu中的相关接口。

**注意：不要尝试在trigger中使用ajax异步请求修改本次分享的内容，因为客户端分享操作是一个同步操作，这时候使用ajax的回包会还没有返回**。

以上几个函数都带有一个参数，类型为对象，其中除了每个接口本身返回的数据之外，还有一个通用属性errMsg，其值格式如下：

1. 调用成功时：”xxx:ok” ，其中xxx为调用的接口名
2. 用户取消时：”xxx:cancel”，其中xxx为调用的接口名
3. 调用失败时：其值为具体错误信息

### 基础接口

判断当前客户端版本是否支持指定JS接口

1. wx.checkJsApi({
2. jsApiList: ['chooseImage'], // 需要检测的JS接口列表，所有JS接口列表见附录2,
3. success: function(res) {
4. // 以键值对的形式返回，可用的api值true，不可用为false
5. // 如：{"checkResult":{"chooseImage":true},"errMsg":"checkJsApi:ok"}
6. }
7. });

备注：checkJsApi接口是客户端1.2.2新引入的一个预留接口。

### 通讯录和会话接口

通讯录选人接口

1. wx.invoke("selectEnterpriseContact", {
2. "fromDepartmentId": -1,// 必填，-1表示打开的通讯录从自己所在部门开始展示, 0表示从最上层开始
3. "mode": "single",// 必填，选择模式，single表示单选，multi表示多选
4. "type": ["department", "user"],// 必填，选择限制类型，指定department、user中的一个或者多个
5. "selectedDepartmentIds": ["2","3"],// 非必填，已选部门ID列表。用于多次选人时可重入
6. "selectedUserIds": ["lisi","lisi2"]// 非必填，已选用户ID列表。用于多次选人时可重入
7. },function(res){
8. if (res.err\_msg == "selectEnterpriseContact:ok")
9. {
10. if(typeof res.result == 'string')
11. {
12. res.result = JSON.parse(res.result) //由于目前各个终端尚未完全兼容，需要开发者额外判断result类型以保证在各个终端的兼容性
13. }
14. var selectedDepartmentList = res.result.departmentList;// 已选的部门列表
15. for (var i = 0; i < selectedDepartmentList.length; i++)
16. {
17. var department = selectedDepartmentList[i];
18. var departmentId = department.id;// 已选的单个部门ID
19. var departemntName = department.name;// 已选的单个部门名称
20. }
21. var selectedUserList = res.result.userList; // 已选的成员列表
22. for (var i = 0; i < selectedUserList.length; i++)
23. {
24. var user = selectedUserList[i];
25. var userId = user.id; // 已选的单个成员ID
26. var userName = user.name;// 已选的单个成员名称
27. var userAvatar= user.avatar;// 已选的单个成员头像
28. }
29. }
30. }
31. );

此接口在企业微信1.3.11及以后版本支持，微信6.5.10及以后版本支持。调用此接口时，config接口必须传入beta参数

#### 创建会话接口

1. wx.openEnterpriseChat({
2. userIds: 'zhangshan;lisi;wangwu', // 必填，参与会话的成员列表。格式为userid1;userid2;...，用分号隔开，最大限制为2000个。userid单个时为单聊，多个时为群聊。
3. groupName: '讨论组', // 必填，会话名称。单聊时该参数传入空字符串""即可。
4. success: function(res) {
5. // 回调
6. },
7. fail: function(res) {
8. if(res.errMsg.indexOf('function not exist') > -1){
9. alert('版本过低请升级')
10. }
11. }
12. });

### 分享接口

获取“转发”按钮点击状态及自定义分享内容接口

1. wx.onMenuShareAppMessage({
2. title: '', // 分享标题
3. desc: '', // 分享描述
4. link: '', // 分享链接
5. imgUrl: '', // 分享图标
6. success: function () {
7. // 用户确认分享后执行的回调函数
8. },
9. cancel: function () {
10. // 用户取消分享后执行的回调函数
11. }
12. });

获取“微信”按钮点击状态及自定义分享内容接口

备注：若未实现onMenuShareWechat，则分享”微信”会获取onMenuShareAppMessage接口的内容。

1. wx.onMenuShareWechat({
2. title: '', // 分享标题
3. desc: '', // 分享描述
4. link: '', // 分享链接
5. imgUrl: '', // 分享图标
6. success: function () {
7. // 用户确认分享后执行的回调函数
8. },
9. cancel: function () {
10. // 用户取消分享后执行的回调函数
11. }
12. });

### 图像与文件接口

1. wx.chooseImage({
2. count: 1, // 默认9
3. sizeType: ['original', 'compressed'], // 可以指定是原图还是压缩图，默认二者都有
4. sourceType: ['album', 'camera'], // 可以指定来源是相册还是相机，默认二者都有
5. success: function (res) {
6. var localIds = res.localIds; // 返回选定照片的本地ID列表，localId可以作为img标签的src属性显示图片
7. }
8. });

拍照或从手机相册中选图接口

预览图片接口

1. wx.previewImage({
2. current: '', // 当前显示图片的http链接
3. urls: [] // 需要预览的图片http链接列表
4. });

上传图片接口

1. wx.uploadImage({
2. localId: '', // 需要上传的图片的本地ID，由chooseImage接口获得
3. isShowProgressTips: 1, // 默认为1，显示进度提示
4. success: function (res) {
5. var serverId = res.serverId; // 返回图片的服务器端ID
6. }
7. });

备注：上传图片有效期3天，可用企业微信多媒体接口下载图片到自己的服务器，此处获得的 serverId 即 media\_id。

下载图片接口

1. wx.downloadImage({
2. serverId: '', // 需要下载的图片的服务器端ID，由uploadImage接口获得
3. isShowProgressTips: 1, // 默认为1，显示进度提示
4. success: function (res) {
5. var localId = res.localId; // 返回图片下载后的本地ID
6. }
7. });

预览文件接口

1. wx.previewFile({
2. url: '', // 需要预览文件的地址(必填，可以使用相对路径)
3. name: '', // 需要预览文件的文件名(不填的话取url的最后部分)
4. size: 1048576 // 需要预览文件的字节大小(必填)
5. });

### 音频接口

开始录音接口

1. wx.startRecord();

停止录音接口

1. wx.stopRecord({
2. success: function (res) {
3. var localId = res.localId;
4. }
5. });

监听录音自动停止接口

1. wx.onVoiceRecordEnd({
2. // 录音时间超过一分钟没有停止的时候会执行 complete 回调
3. complete: function (res) {
4. var localId = res.localId;
5. }
6. });

播放语音接口

1. wx.playVoice({
2. localId: '' // 需要播放的音频的本地ID，由stopRecord接口获得
3. });

暂停播放接口

1. wx.playVoice({
2. localId: '' // 需要播放的音频的本地ID，由stopRecord接口获得
3. });

停止播放接口

1. wx.stopVoice({
2. localId: '' // 需要停止的音频的本地ID，由stopRecord接口获得
3. });

监听语音播放完毕接口

1. wx.onVoicePlayEnd({
2. success: function (res) {
3. var localId = res.localId; // 返回音频的本地ID
4. }
5. });

上传语音接口

1. wx.uploadVoice({
2. localId: '', // 需要上传的音频的本地ID，由stopRecord接口获得
3. isShowProgressTips: 1, // 默认为1，显示进度提示
4. success: function (res) {
5. var serverId = res.serverId; // 返回音频的服务器端ID
6. }
7. });

备注：上传语音有效期3天，可用企业微信多媒体接口下载语音到自己的服务器，此处获得的 serverId 即 media\_id。

下载语音接口

1. wx.downloadVoice({
2. serverId: '', // 需要下载的音频的服务器端ID，由uploadVoice接口获得
3. isShowProgressTips: 1, // 默认为1，显示进度提示
4. success: function (res) {
5. var localId = res.localId; // 返回音频的本地ID
6. }
7. });

### 设备信息

获取网络状态接口

1. wx.getNetworkType({
2. success: function (res) {
3. var networkType = res.networkType; // 返回网络类型2g，3g，4g，wifi
4. }
5. });

### 地理位置

使用企业微信内置地图查看位置接口

1. wx.openLocation({
2. latitude: 0, // 纬度，浮点数，范围为90 ~ -90
3. longitude: 0, // 经度，浮点数，范围为180 ~ -180。
4. name: '', // 位置名
5. address: '', // 地址详情说明
6. scale: 1, // 地图缩放级别,整形值,范围从1~28。默认为16
7. });

获取地理位置接口

### 界面操作

1. wx.getLocation({
2. type: 'wgs84', // 默认为wgs84的gps坐标，如果要返回直接给openLocation用的火星坐标，可传入'gcj02'
3. success: function (res) {
4. var latitude = res.latitude; // 纬度，浮点数，范围为90 ~ -90
5. var longitude = res.longitude; // 经度，浮点数，范围为180 ~ -180。
6. var speed = res.speed; // 速度，以米/每秒计
7. var accuracy = res.accuracy; // 位置精度
8. }
9. });

隐藏右上角菜单接口

1. wx.hideOptionMenu();

显示右上角菜单接口

1. wx.showOptionMenu();

关闭当前网页窗口接口

1. wx.closeWindow();

批量隐藏功能按钮接口

1. wx.hideMenuItems({
2. menuList: [] // 要隐藏的菜单项，所有menu项见附录3
3. });

批量显示功能按钮接口

1. wx.showMenuItems({
2. menuList: [] // 要显示的菜单项，所有menu项见附录3
3. });

隐藏所有非基础按钮接口

1. wx.hideAllNonBaseMenuItem();

显示所有功能按钮接口

1. wx.showAllNonBaseMenuItem();

### 企业微信扫一扫

调起企业微信扫一扫接口

1. javascript
2. wx.scanQRCode({
3. desc: 'scanQRCode desc',
4. needResult: 0, // 默认为0，扫描结果由企业微信处理，1则直接返回扫描结果，
5. scanType: ["qrCode", "barCode"], // 可以指定扫二维码还是一维码，默认二者都有
6. success: function(res) {
7. // 回调
8. },
9. error: function(res) {
10. if (res.errMsg.indexOf('function\_not\_exist') > 0) {
11. alert('版本过低请升级')
12. }
13. }
14. });

### 附录1-JS-SDK使用权限签名算法

**jsapi\_ticket**

生成签名之前必须先了解一下jsapi\_ticket，jsapi\_ticket是H5应用调用企业微信JS接口的临时票据。正常情况下，**jsapi\_ticket的有效期为7200秒**，通过access\_token来获取。由于获取jsapi\_ticket的api调用次数非常有限，频繁刷新jsapi\_ticket会导致api调用受限，影响自身业务，**开发者必须在自己的服务全局缓存jsapi\_ticket**。

获取access\_token（**有效期7200秒，开发者必须在自己的服务全局缓存access\_token**）

AccessToken是应用的全局唯一票据，调用接口时需携带AccessToken。

AccessToken需要用CorpID和Secret来换取，不同的Secret会返回不同的AccessToken。**正常情况下AccessToken有效期为7200秒，有效期内重复获取返回相同结果。access\_token至少保留512字节的存储空间。**

**请求说明**  
Https请求方式: GET  
http://wwlocal.qq.com/cgi-bin/gettoken?corpid=id&corpsecret=secrect

**参数说明**

| 参数 | 必须 | 说明 |
| --- | --- | --- |
| corpid | 是 | 企业Id |
| corpsecret | 是 | 应用secret |

**权限说明**

每个应用有自己的secret，代表了对应用的不同权限

**返回说明**

1. 正确的Json返回结果:
2. {
3. "access\_token": "accesstoken000001",
4. "expires\_in": 7200
5. }

| **参数** | **说明** |
| --- | --- |
| access\_token | 获取到的凭证。长度为64至512个字节 |
| expires\_in | 凭证的有效时间（秒） |

1. 错误的Json返回示例:
2. {
3. "errcode": 43003,
4. "errmsg": "require https"
5. }
6. 用第一步拿到的access\_token 采用http GET方式请求获得jsapi\_ticket（**有效期7200秒，开发者必须在自己的服务全局缓存jsapi\_ticket**）：http://wwlocal.qq.com/cgi-bin/get\_jsapi\_ticket?access\_token=ACCESS\_TOKEN

成功返回如下JSON：

1. {
2. "errcode":0,
3. "errmsg":"ok",
4. "ticket":"bxLdikRXVbTPdHSM05e5u5sUoXNKd8-41ZO3MhKoyN5OfkWITDGgnr2fwJ0m9E8NYzWKVZvdVtaUgWvsdshFKA",
5. "expires\_in":7200
6. }

jsapi\_ticket最长为512字节。获得jsapi\_ticket之后，就可以生成JS-SDK权限验证的签名了。

**签名算法**

签名生成规则如下：参与签名的字段包括noncestr（随机字符串）, 有效的jsapi\_ticket, timestamp（时间戳）, url（当前网页的URL，不包含#及其后面部分） 。对所有待签名参数按照字段名的ASCII 码从小到大排序（字典序）后，使用URL键值对的格式 （即 key1=value1&key2=value2…）拼接成字符串string1。这里需要注意的是所有参数名均为小写字符。对string1作sha1加密，字段名和字段值都采用原始值，不进行URL 转义。

即signature=sha1(string1)。 示例

1. noncestr=Wm3WZYTPz0wzccnW
2. jsapi\_ticket=sM4AOVdWfPE4DxkXGEs8VMCPGGVi4C3VM0P37wVUCFvkVAy\_90u5h9nbSlYy3-Sl-HhTdfl2fzFy1AOcHKP7qg
3. timestamp=1414587457
4. url=http://mp.weixin.qq.com

步骤1. 对所有待签名参数按照字段名的ASCII 码从小到大排序（字典序）后，使用URL键值对的格式（即key1=value1&key2=value2…）拼接成字符串string1：

1. jsapi\_ticket=sM4AOVdWfPE4DxkXGEs8VMCPGGVi4C3VM0P37wVUCFvkVAy\_90u5h9nbSlYy3-Sl-HhTdfl2fzFy1AOcHKP7qg&noncestr=Wm3WZYTPz0wzccnW&timestamp=1414587457&url=http://mp.weixin.qq.com

步骤2. 对string1进行sha1签名，得到signature：

1. 0f9de62fce790f9a083d5c99e95740ceb90c27ed

注意事项

1. 签名用的noncestr和timestamp必须与wx.config中的nonceStr和timestamp相同。
2. 签名用的url必须是调用JS接口页面的完整URL。
3. 出于安全考虑，开发者必须在服务器端实现签名的逻辑。

如出现invalid signature 等错误详见附录4常见错误及解决办法。

### 附录2-所有JS接口列表

* onMenuShareAppMessage
* onMenuShareWechat
* startRecord
* stopRecord
* onVoiceRecordEnd
* playVoice
* pauseVoice
* stopVoice
* onVoicePlayEnd
* uploadVoice
* downloadVoice
* chooseImage
* previewImage
* uploadImage
* downloadImage
* getNetworkType
* openLocation
* getLocation
* hideOptionMenu
* showOptionMenu
* hideMenuItems
* showMenuItems
* hideAllNonBaseMenuItem
* showAllNonBaseMenuItem
* closeWindow
* scanQRCode

### 附录3-所有菜单项列表

基本类

* 调整字体: “menuItem:setFont”
* 刷新: “menuItem:refresh”

传播类

* 转发: “menuItem:share:appMessage”
* 微信: “menuItem:share:wechat”
* 收藏: “menuItem:favorite”

保护类

* 复制链接: “menuItem:copyUrl”
* Safari: “menuItem:openWithSafari”
* 邮件: “menuItem:share:email”

### 附录4-常见错误及解决方法

调用config接的时候传入参数debug: true 可以开启debug模式，页面会alert出错误信息。以下为常见错误及解决方法：

1. invalid url domain：当前页面所在域名与使用的corpid没有绑定（可在该应用的可信域名中配置域名）。
2. invalid signature签名错误：建议按如下顺序检查：
3. 确认签名算法正确，可用http://wwlocal.qq.com/api/jsapisign页面工具进行校验。
4. 确认config中nonceStr（js中驼峰标准大写S）, timestamp与用以签名中的对应noncestr, timestamp一致。
5. 确认url是页面完整的url(请在当前页面alert(location.href.split(‘#’)[0])确认)，包括’http(s)://‘部分，以及’？’后面的GET参数部分,但不包括’#’hash后面的部分。
6. 确认config中的appid与用来获取jsapi\_ticket的corpid一致。
7. 确保一定缓存access\_token和jsapi\_ticket。
8. **确保你获取用来签名的url是动态获取的，动态页面可参见实例代码中php的实现方式。如果是html的静态页面在前端通过ajax将url传到后台签名，前端需要用js获取当前页面除去’#’hash部分的链接（可用location.href.split(‘#’)[0]获取,而且需要encodeURIComponent），因为页面一旦分享，企业微信客户端会在你的链接末尾加入其它参数，如果不是动态获取当前链接，将导致分享后的页面签名失败**。
9. the permission value is offline verifying：这个错误是因为config没有正确执行，或者是调用的JSAPI没有传入config的jsApiList参数中。建议按如下顺序检查：
10. 确认config正确通过。
11. 如果是在页面加载好时就调用了JSAPI，则必须写在wx.ready的回调中。
12. 确认config的jsApiList参数包含了这个JSAPI。
13. permission denied：该应用没有权限使用这个接口。
14. function not exist：当前客户端版本不支持该接口，请升级到新版体验。
15. 服务上线之后无法获取jsapi\_ticket，自己测试时没问题：因为access\_token和jsapi\_ticket必须要在自己的服务器缓存，否则上线后会触发频率限制。请确保一定对token和ticket做缓存以减少服务器请求，不仅可以避免触发频率限制，还加快你们自己的服务速度。目前为了方便测试提供了1w的获取量，超过阀值后，服务将不再可用，请确保在**服务上线前一定全局缓存access\_token和jsapi\_ticket，两者有效期均为7200秒（以返回结果中的expires\_in为准），否则一旦上线触发频率限制，服务将不再可用**。
16. uploadImage怎么传多图：目前只支持一次上传一张，多张图片需等前一张图片上传之后再调用该接口。
17. 没法对本地选择的图片进行预览：chooseImage接口本身就支持预览，不需要额外支持。
18. 通过a链接(例如先通过企业微信授权登录)跳转到b链接，invalid signature签名失败：后台生成签名的链接为使用jssdk的当前链接，也就是跳转后的b链接，请不要用企业微信登录的授权链接进行签名计算，后台签名的url一定是使用jssdk的当前页面的完整url除去’#’部分。
19. 出现config:fail错误：这是由于传入的config参数不全导致，请确保传入正确的appId、timestamp、nonceStr、signature和需要使用的jsApiList。
20. 如何把jsapi上传到企业微信的多媒体资源下载到自己的服务器：请参见文档中uploadVoice和uploadImage接口的备注说明。
21. Android通过jssdk上传到企业微信服务器，第三方再从企业微信下载到自己的服务器，会出现杂音：企业微信团队已经修复此问题，目前后台已优化上线。
22. 绑定父级域名，是否其子域名也是可用的：是的，合法的子域名在绑定父域名之后是完全支持的。
23. 是否需要对低版本自己做兼容：jssdk都是兼容低版本的，不需要第三方自己额外做更多工作，但有的接口是新引入的，只有新版才可调用。
24. getLocation返回的坐标在openLocation有偏差，因为getLocation返回的是gps坐标，openLocation打开的腾讯地图为火星坐标，getLocation也可以直接获取火星坐标。

**版本信息：**  
该部分v1.0及以上版本有效

## 网页授权

### 概述

企业微信提供了OAuth的授权登录方式，可以让网页和企业微信共享用户ID，从而免去登录的环节。

此文档面向网页开发者介绍企业微信网页授权如何使用及相关注意事项。

### 关于网页授权的可信域名

1. 在开始使用网页授权之前，开发者需要先登录到企业管理端后台，选择“企业应用”选项卡，进入需要使用网页授权的应用并编辑“可信域名”表单项，此选项将用于网页OAuth2.0授权的时候进行安全验证。请注意，这里填写的是域名（是一个字符串），而不是URL，因此请勿加 http:// 等协议头
2. 可信域名配置规范为全域名，且**需要通过ICP备案**(否则在微信侧jssdk功能失效)。比如需要网页授权的域名为：www.qq.com，配置以后此域名下面的页面http://www.qq.com/music.html 、 http://www.qq.com/login.html 都可以进行OAuth2.0鉴权。但http://pay.qq.com 、 http://music.qq.com 、 http://qq.com无法进行OAuth2.0鉴权
3. 域名的验证逻辑说明：由于一个企业可以自定义多个应用，每个应用都可以配置一个安全域名，在做oauth跳转的时候，只需要携带coprid即可，企业微信的后台会遍历该企业下面配置的所有应用并检查其配置的可信域名，只要任意一个应用的可信域名匹配则校验成功。

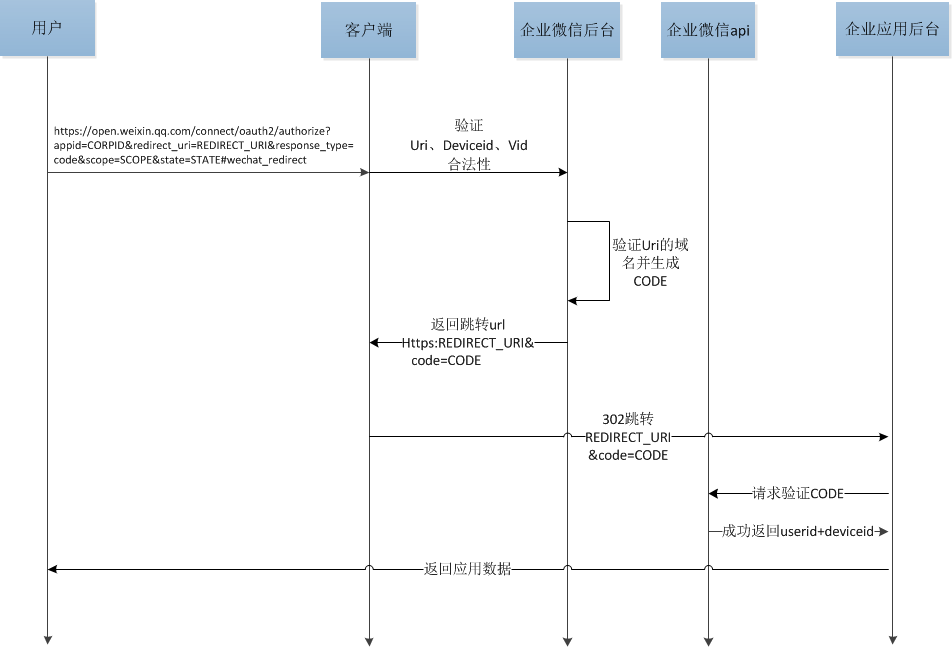
### 关于UserID机制

UserId用于在一个企业内唯一标识一个用户，通过网页授权接口可以获取到当前用户的UserId信息，如果需要获取用户的更多信息可以调用通讯录管理的成员接口来获取。

### 接入流程说明

企业应用中的URL链接（包括自定义菜单或者消息中的链接），均可通过OAuth2.0验证接口来获取成员的UserId身份信息。注意：此URL的域名，必须完全匹配企业应用设置项中的“可信域名”（如果你的redirect\_uri有端口号，那么“可信域名”也必须加上端口号），否则跳转时会提示redirect\_uri参数错误。

通过此接口获取成员身份会有一定的时间开销。对于频繁获取成员身份的场景，建议采用如下方案：  
1、企业应用中的URL链接直接填写企业自己的页面地址  
2、成员操作跳转到步骤1的企业页面时，企业后台校验是否有标识成员身份的cookie信息，此cookie由企业生成  
3、如果没有匹配的cookie，则重定向到OAuth验证链接，获取成员的身份信息后，由企业后台植入标识成员身份的cookie信息  
4、根据cookie获取成员身份后，再进入相应的页面



OAuth2.0流程图

### 关键步骤



#### 获取code

如果企业需要在打开的网页里面携带用户的身份信息，第一步需要构造如下的链接来获取code参数：  
https://open.weixin.qq.com/connect/oauth2/authorize?appid=CORPID&redirect\_uri=REDIRECT\_URI&response\_type=code&scope=SCOPE&agentid=AGENTID&state=STATE#wechat\_redirect

**参数说明：**

| **参数** | **必须** | **说明** |
| --- | --- | --- |
| appid | 是 | 企业的CorpID |
| redirect\_uri | 是 | 授权后重定向的回调链接地址，请使用urlencode对链接进行处理 |
| response\_type | 是 | 返回类型，此时固定为：code |
| scope | 是 | 应用授权作用域。 snsapi\_base：静默授权，可获取成员的基础信息； snsapi\_userinfo：静默授权，可获取成员的详细信息，但不包含手机、邮箱； snsapi\_privateinfo：手动授权，可获取成员的详细信息，包含手机、邮箱。 |
| agentid | 否 | 企业应用的id。 当scope是snsapi\_userinfo或snsapi\_privateinfo时，该参数必填。 注意redirect\_uri的域名必须与该应用的可信域名一致。 |
| state | 否 | 重定向后会带上state参数，企业可以填写a-zA-Z0-9的参数值，长度不可超过128个字节 |
| #wechat\_redirect | 是 | 终端使用此参数判断是否需要带上身份信息 |

员工点击后，页面将跳转至 redirect\_uri?code=CODE&state=STATE，企业可根据code参数获得员工的userid。code长度最大为512字节。

**权限说明：**  
企业无限制；第三方使用snsapi\_privateinfo的scope时，应用必须有’成员敏感信息授权’的权限。

#### 根据code获取成员信息

**请求方式：**GET

**请求地址：**http://wwlocal.qq.com/cgi-bin/user/getuserinfo?access\_token=ACCESS\_TOKEN&code=CODE

**参数说明：**

| **参数** | **必须** | **说明** |
| --- | --- | --- |
| access\_token | 是 | 调用接口凭证 |
| code | 是 | 通过成员授权获取到的code，最大为512字节。每次成员授权带上的code将不一样，code只能使用一次，5分钟未被使用自动过期。 |

**权限说明：**

跳转的域名须完全匹配access\_token对应应用的可信域名。

**返回结果：**

1. 当用户为企业成员时返回示例如下：
2. {
3. "errcode": 0,
4. "errmsg": "ok",
5. "UserId":"USERID",
6. "DeviceId":"DEVICEID",
7. "user\_ticket": "USER\_TICKET"，
8. "expires\_in":7200
9. }

| **参数** | **说明** |
| --- | --- |
| errcode | 返回码 |
| errmsg | 对返回码的文本描述内容 |
| UserId | 成员UserID |
| DeviceId | 手机设备号(由企业微信在安装时随机生成，删除重装会改变，升级不受影响) |
| user\_ticket | 成员票据，最大为512字节。 scope为snsapi\_userinfo或snsapi\_privateinfo，且用户在应用可见范围之内时返回此参数。 后续利用该参数可以获取用户信息或敏感信息。 |
| expires\_in | user\_token的有效时间（秒），随user\_ticket一起返回 |

1. 非企业成员授权时返回示例如下：
2. {
3. "errcode": 0,
4. "errmsg": "ok",
5. "OpenId":"OPENID",
6. "DeviceId":"DEVICEID"
7. }

| **参数** | **说明** |
| --- | --- |
| errcode | 返回码 |
| errmsg | 对返回码的文本描述内容 |
| OpenId | 非企业成员的标识，对当前企业唯一 |
| DeviceId | 手机设备号(由企业微信在安装时随机生成，删除重装会改变，升级不受影响) |

**出错返回示例：**

1. {
2. "errcode": 40029,
3. "errmsg": "invalid code"
4. }

#### 使用user\_ticket获取成员详情

**请求方式：**POST  
**请求地址：**http://wwlocal.qq.com/cgi-bin/user/getuserdetail?access\_token=ACCESS\_TOKEN

**请求包体：**

1. {
2. "user\_ticket": "USER\_TICKET"
3. }

**参数说明：**

| **参数** | **必须** | **说明** |
| --- | --- | --- |
| access\_token | 是 | 调用接口凭证 |
| user\_ticket | 是 | 成员票据 |

**权限说明：**  
需要有对应应用的使用权限，且成员必须在授权应用的可见范围内。

**返回结果：**

1. {
2. "userid":"lisi",
3. "name":"李四",
4. "department":[3],
5. "position": "后台工程师",
6. "mobile":"15050495892",
7. "gender":1,
8. "email":"xxx@xx.com",
9. "avatar":"http://shp.qpic.cn/bizmp/xxxxxxxxxxx/0"
10. }

**参数说明：**

| **参数** | **说明** |
| --- | --- |
| userid | 成员UserID |
| name | 成员姓名 |
| department | 成员所属部门 |
| position | 职位信息 |
| mobile | 成员手机号，仅在用户同意snsapi\_privateinfo授权时返回 |
| gender | 性别。0表示未定义，1表示男性，2表示女性 |
| email | 成员邮箱，仅在用户同意snsapi\_privateinfo授权时返回 |
| avatar | 头像url。注：如果要获取小图将url最后的”/0”改成”/64”即可 |

**版本信息：**  
v1.0以上版本有效

# 企业微信登录

## 移动应用

### IOS应用

**iOS接入指南**

**接入说明：**  
任何需要调用企业微信API的应用，都需要先在企业微信管理后台创建一个应用，设置完成后，即可使用对应的功能。

**创建应用**

1. 管理员登录企业微信管理后台，选择**单位应用**
2. 选择已有应用或添加应用，进入应用详情页面。
3. 选择“企业微信授权登录”，在设置界面填写iOS App的Bundleid，设置完成后系统自动生成应用程序**schema**。

**下载企业微信终端SDK文件**

SDK文件包括 libWXWorkApi.a，WWKApi.h，WWKApiObject.h 三个。  
请前往 资源下载 下载最新SDK包sdk.zip  
下载包包含4个文件，其名称和含义分别是：  
**WWKApi.h** 定义SDK接口的头文件  
**WWKObject.h** 定义SDK接口用来进行参数传递的对象  
**libWXWorkApi.a** 第三方工程需要引用的SDK静态库文件  
**demo.tar.gz** 一个供参考的企业微信iOS端SDK使用的示例工程，这个工程介绍了当前版本SDK提供的所有接口的使用方法

**搭建开发环境**

1. 在XCode中建立你的工程。
2. 将SDK文件中包含的 **libWXWorkApi.a**，**WXApi.h**，**WXApiObject.h** 三个文件添加到你所建的工程中。具体实现可参考 资源下载 页面中提供的SDK Sample Demo工程(下载sdk.zip压缩包后，解压其中的**demo.tar.gz**即为一个供参考的企业微信iOS端SDK使用的示例工程，这个工程介绍了当前版本SDK提供的所有接口的使用方法)。
3. 在你的工程文件中选中Search Paths，并添加 **libWeChatSDK.a** ，**WXApi.h**，**WXApiObject.h**，文件所在位置。
4. 在XCode中，选择你的工程设置项，选中“TARGETS”一栏，在“info”标签栏的“URL type“添加“URL scheme”，其内容为你所注册的应用程序将获得应用程序**ID**和**schema**。
5. 在你需要使用企业微信终端API的文件中 import WWKApi.h 并增加 WXApiDelegate 协议。
6. #import "AppDelegate.h"
7. #import "WWKApi.h"
8. @interface AppDelegate () <WWKApiDelegate>
9. @end
10. @implementation AppDelegate
11. @end

**在代码中使用SDK进行初始化注册**

1.你的程序启动后企业微信终端能响应你的程序，必须在代码中向企业微信终端注册你的schema。(如下面代码所示，在 AppDelegate 的 didFinishLaunchingWithOptions 函数中向企业微信注册id)。  
**参数说明**  
**registerApp :** 必填字段 用户填写第三方应用Bundleid完成上述注册流程后，系统根据corpId(单位ID 见下描述)与agentId(应用ID 见下描述)计算得到的一个schema字符串。用于企业微信完成SSO登录流程后回调返回用户的第三方应用  
**corpId:** 必填字段 用户登录单位管理页面的时候在 “我的单位” 页面会得到一个corpId，这个corpid即为需要填在这里的单位ID  
**agentId:** 必填字段 当用户为自己单位新加一个第三方iOS应用的时候，系统会为每个iOS应用分配不同的单位内部唯一的应用ID

1. - (BOOL)application:(UIApplication \*)application didFinishLaunchingWithOptions:(NSDictionary \*)launchOptions {
2. /\*! @brief 调用这个方法前需要先到管理端进行注册 走管理端的注册方式
3. \*
4. \* 在管理端通过注册(可能需要等待审批)，获得schema+corpid+agentid
5. \* @param registerApp 第三方App的Schema
6. \* @param corpId 第三方App所属单位的ID
7. \* @param agentId 第三方App在单位内部的ID
8. \*/
9. [WWKApi registerApp:@"wwauth700000a5f2191000014" corpId:@"ww700000a5f2191" agentId:@"1000014"];
10. return YES;
11. }

2.AppDelegate的handleOpenURL和openURL方法，第三方iOS App需要在这两个方法中调用SDK的handleOpenURL接口方法，让SDK有机会在第三方App内部处理SSO结果返回给第三方iOS App后的后续处理工作

1. - (BOOL)application:(UIApplication \*)application openURL:(NSURL \*)url sourceApplication:(nullable NSString \*)sourceApplication annotation:(id)annotation {
2. return [self handleOpenURL:url sourceApplication:sourceApplication];
3. }
4. - (BOOL)handleOpenURL:(NSURL \*)url sourceApplication:(NSString \*)sourceApplication {
5. return [WWKApi handleOpenURL:url delegate:self];
6. }

3.现在，你的程序要实现和企业微信终端交互的具体请求与回应，因此需要实现WWKApiDelegate协议的两个方法。第三方iOS App可以根据自身需求实现这两个方法

1. // 如果第三方程序向企业微信发送了sendReq的请求，那么onResp会被回调。sendReq请求调用后，会切到企业微信终端程序界面。
2. - (void)onResp:(WWKBaseResp \*)resp {
3. }
4. // onReq是企业微信终端向第三方程序发起请求，要求第三方程序响应。第三方程序响应完后必须调用sendRsp返回。在调用sendRsp返回时，会切回到企业微信终端程序界面。
5. -(void) onReq:(WWKBaseReq\*)req {
6. }

具体在此两方法中所要完成的内容由你定义，具体可参考企业微信开发工具包中的SDK Sample Demo源码。

**接入企业微信登录**

使用SDK中的企业微信登录功能，获取企业微信用户信息。

1. // 其中req参数为WWKSendMessageReq类型 根据发送的消息类型不同需要设置不同的req变量
2. -(BOOL) sendReq:(BaseReq\*)req
3. (IBAction)SSO:(id)sender {
4. WWKSSOReq \*req = [[WWKSSOReq alloc] init];
5. // state参数为这次请求的唯一标示，客户端需要维护其唯一性。SSO回调时会原样返回
6. req.state = @"adfasdfasdf23412341fqw4df14t134rtflajssf8934haioefy";
7. [WWKApi sendReq:req];
8. }

需要特别指出的是，使用了SSO接口进行登录的程序需要在 handleOpenURL 函数中处理来自企业微信的回调，企业微信返回的是获取用户信息的一次性code，第三方app需要继续使用此code通过后端服务器换取用户的信息。

**通过code获取用户信息**

**请求方式：**GET  
**请求地址：**https://saastest2.gov.weixin.qq.com/cgi-bin/user/getuserinfo?access\_token=ACCESS\_TOKEN&code=CODE

**参数说明：**

| 参数 | 必须 | 说明 |
| --- | --- | --- |
| access\_token | 是 | 调用接口凭证 |
| code | 是 | 通过成员授权获取到的code，每次成员授权带上的code将不一样，code只能使用一次，5分钟未被使用自动过期 |

**权限说明：**  
跳转的域名须完全匹配单位内任一应用的可信域名。

**返回结果：**

1. {
2. "UserId":"USERID",
3. }

| **参数** | **说明** |
| --- | --- |
| UserId | 成员UserID |

**出错返回示例：**

**版本信息：**  
v1.0以上版本有效

1. {
2. "errcode": 40029,
3. "errmsg": "invalid code"
4. }

### Android应用

**Android接入指南**

**接入说明：**  
任何需要调用企业微信API的应用，都需要先在企业微信管理后台创建一个应用，设置完成后，即可使用对应的功能。

**创建应用**

1. 管理员登录企业微信管理后台，选择**单位应用**
2. 选择已有应用或添加应用，进入应用详情页面。
3. 选择“企业微信授权登录”，在设置界面填写Android的 App的签名&包名，设置完成后系统自动生成应用程序**schema**。

企业微信分享的设置，复用企业微信授权登录的设置，设置以后登录、分享任何一个都可以使用。

1. 下载企业微信终端开发工具包  
   wwapi.zip  
   每个第三方应用必须要导入该sdk库，用于实现与企业微信的通信  
   示例程序
2. 搭建开发环境  
   在你需要使用企业微信终端API的文件中导入相应的类。  
   import com.tencent.wework.api.model.WWMediaText;
3. 在代码中使用开发工具包
   * 注册到企业微信  
     要使你的程序启动后企业微信终端能响应你的程序，必须在代码中向企业微信终端注册你的id。可以在程序入口Activity的onCreate回调函数处，或其他合适的地方将你的应用schema注册到企业微信。
4. private static final String APPID = "WW1e933be11645237c";
5. private static final String AGENTID = "1000012";
6. private static final String SCHEMA = "wwauth1e933be11645237c000012";
7. @Override
8. protected void onCreate(Bundle savedInstanceState) {
9. super.onCreate(savedInstanceState);
10. setContentView(R.layout.activity\_main);
11. stringId = getApplicationInfo().labelRes;
12. iwwapi = WWAPIFactory.createWWAPI(this);
13. iwwapi.registerApp(SCHEMA);
14. }

如果是修改了包名的定制包，需要增加该定制包的包名。

1. iwwapi.addPackageName("com.tencent.weworklocal");
   * 发送请求和接收返回值  
     现在，你的程序要发送请求到企业微信终端，可以通过IWWAPI的sendMessage方法来实现。  
     boolean sendMessage(BaseMessage msg);  
     sendMessage是第三方app主动发送消息给企业微信，发送完成之后会切回到第三方app界面。  
     boolean sendMessage(BaseMessage req, IWWAPIEventHandler callback)  
     带回调的sendMessage是第三方app向企业微信请求数据，企业微信回应数据之后会切回到第三方app界面。  
     sendMessage的实现示例，如下所示：
2. WWMediaText txt = new WWMediaText(((EditText) findViewById(R.id.et3)).getText().toString());
3. txt.appPkg = getPackageName();
4. txt.appName = getString(stringId);
5. txt.appId = APPID;
6. txt.agentId = AGENTID;
7. iwwapi.sendMessage(txt);

带回调的sendMessage的实现，如下所示：

1. final WWAuthMessage.Req req = new WWAuthMessage.Req();
2. req.sch = SCHEMA;
3. req.appId = APPID;
4. req.agentId = AGENTID;
5. req.state = "dd";
6. iwwapi.sendMessage(req, new IWWAPIEventHandler() {
7. @Override
8. public void handleResp(BaseMessage resp) {
9. if (resp instanceof WWAuthMessage.Resp) {
10. WWAuthMessage.Resp rsp = (WWAuthMessage.Resp) resp;
11. if (rsp.errCode == WWAuthMessage.ERR\_CANCEL) {
12. Toast.makeText(MainActivity.this, "登陆取消", Toast.LENGTH\_SHORT).show();
13. }else if (rsp.errCode == WWAuthMessage.ERR\_FAIL) {
14. Toast.makeText(MainActivity.this, "登陆失败", Toast.LENGTH\_SHORT).show();
15. } else if (rsp.errCode == WWAuthMessage.ERR\_OK) {
16. Toast.makeText(MainActivity.this, "登陆成功：" + rsp.code,
17. Toast.LENGTH\_SHORT).show();
18. }
19. }
20. }
21. });

具体要发送的内容由第三方app开发者定义，具体可参考微信开发工具包中的SDK apitest源码。  
注意：  
如果需要混淆代码，为了保证sdk的正常使用，需要在proguard.cfg加上下面两行配置：

1. -keep class com.tencent.wework.api.\*\* {
2. \*;
3. }

至此，你已经能使用企业微信Android开发工具包的API内容了。如果想更详细了解每个API函数的用法，请查阅 Android 平台参考手册 或自行下载阅读企业微信apitest源码。

**授权登录接入**

开发者需要配合使用企业微信提供的SDK进行授权登录请求接入。正确接入SDK后开发者移动应用会在终端本地拉起企业微信应用进行授权登录，企业微信用户确认后企业微信将回调授权临时票据（code），同时返回开发者移动应用。  
Android平台应用授权登录接入代码示例（请参考Android接入指南）：

1. final WWAuthMessage.Req req = new WWAuthMessage.Req();
2. req.sch = SCHEMA;
3. req.appId = APPID;
4. req.agentId = AGENTID;
5. req.state = "dd";
6. iwwapi.sendMessage(req, new IWWAPIEventHandler() {
7. @Override
8. public void handleResp(BaseMessage resp) {
9. if (resp instanceof WWAuthMessage.Resp) {
10. WWAuthMessage.Resp rsp = (WWAuthMessage.Resp) resp;
11. if (rsp.errCode == WWAuthMessage.ERR\_CANCEL) {
12. Toast.makeText(MainActivity.this, "登陆取消", Toast.LENGTH\_SHORT).show();
13. }else if (rsp.errCode == WWAuthMessage.ERR\_FAIL) {
14. Toast.makeText(MainActivity.this, "登陆失败", Toast.LENGTH\_SHORT).show();
15. } else if (rsp.errCode == WWAuthMessage.ERR\_OK) {
16. Toast.makeText(MainActivity.this, "登陆成功：" + rsp.code,
17. Toast.LENGTH\_SHORT).show();
18. }
19. }
20. }
21. });

参数说明

| **参数** | **是否必须** | **说明** |
| --- | --- | --- |
| appid | 是 | 单位唯一标识。创建单位后显示在，我的单位 CorpID字段 |
| agentid | 是 | 应用唯一标识。显示在具体应用下的 AgentId字段 |
| sch | 是 | 应用跳转标识，显示在具体应用下的 Schema字段 |
| state | 否 | 用于保持请求和回调的状态，授权请求后原样带回给第三方。该参数可用于防止csrf攻击（跨站请求伪造攻击），建议第三方带上该参数，可设置为简单的随机数加session进行校验 |

参数示例：  
appid: WW1e933be11645237c  
agentid: 1000012  
sch：wwauth1e933be11645237c000012  
state: wwapitest

可拉起企业微信打开授权登录页：  
返回说明  
用户点击授权后，企业微信客户端会被拉起，跳转至授权界面，用户在该界面点击允许或取消，SDK通过回调返回数据给调用方。  
返回值  
说明

1. errCode
2. ERR\_OK = 0(用户同意)
3. ERR\_FAIL = 1（授权失败）
4. ERR\_CANCEL = -1（用户取消）

code  
用户换取access\_token的code，仅在ErrCode为0时有效  
state  
第三方程序发送时用来标识其请求的唯一性的标志，由第三方程序调用sendMessage时传入，由企业微信终端回传，state字符串长度不能超过1K

**通过code获取用户信息**

**请求方式：**GET  
**请求地址：**https://saastest2.gov.weixin.qq.com/cgi-bin/user/getuserinfo?access\_token=ACCESS\_TOKEN&code=CODE

**参数说明：**

| 参数 | 必须 | 说明 |
| --- | --- | --- |
| access\_token | 是 | 调用接口凭证 |
| code | 是 | 通过成员授权获取到的code，每次成员授权带上的code将不一样，code只能使用一次，5分钟未被使用自动过期 |

**权限说明：**  
跳转的域名须完全匹配单位内任一应用的可信域名。

**返回结果：**

1. {
2. "UserId":"USERID",
3. }

| 参数 | 必须 | 说明 |
| --- | --- | --- |
| UserId | 成员UserID | UserId |

**出错返回示例：**

1. {
2. "errcode": 40029,
3. "errmsg": "invalid code"
4. }

**版本信息：**  
v1.0以上版本有效

### 资源下载

**iOS资源下载**

**开发工具包（SDK）及范例Xcode工程**

使用企业微信登录、分享等功能需要用到的库以及头文件。点击下载 iOS开发工具包集合（0.15版本）。

该下载包包含4个文件，其名称和含义分别是：  
WWKApi.h 定义SDK接口的头文件  
WWKObject.h 定义SDK接口用来进行参数传递的对象  
libWXWorkApi.a 第三方工程需要引用的SDK静态库文件  
demo.tar.gz 一个供参考的企业微信iOS端SDK使用的示例工程，这个工程介绍了当前版本SDK提> 供的所有接口的使用方法  
sdk.zip md5: 2c4a9b32529c75a0ce8866d3e769b045

**Android资源下载**

**开发SDK、签名工具以及Demo**

使用企业微信登录、分享功能需要用到文件如下。  
点击下载Android开发工具包集合 。每个第三方应用必须要导入该SDK库，用于实现与企业微信的通信  
点击下载 签名生成工具。安装此工具，根据指引生成的应用签名字符串并填写在应用管理平台上，未填写或者填写错误将导致接口无法使用。  
点击下载接口使用Demo。开发者可以参考Demo中的接口使用方式进行开发。

## 网页应用

### 概叙

网页应用登录是基于“企业微信网页授权”构建的企业微信OAuth2.0授权登录系统。

在进行企业微信授权登录接入之前，在单位的管理端后台创建一个具备“企业微信授权登录”能力的应用（见“开启网页接入”）。

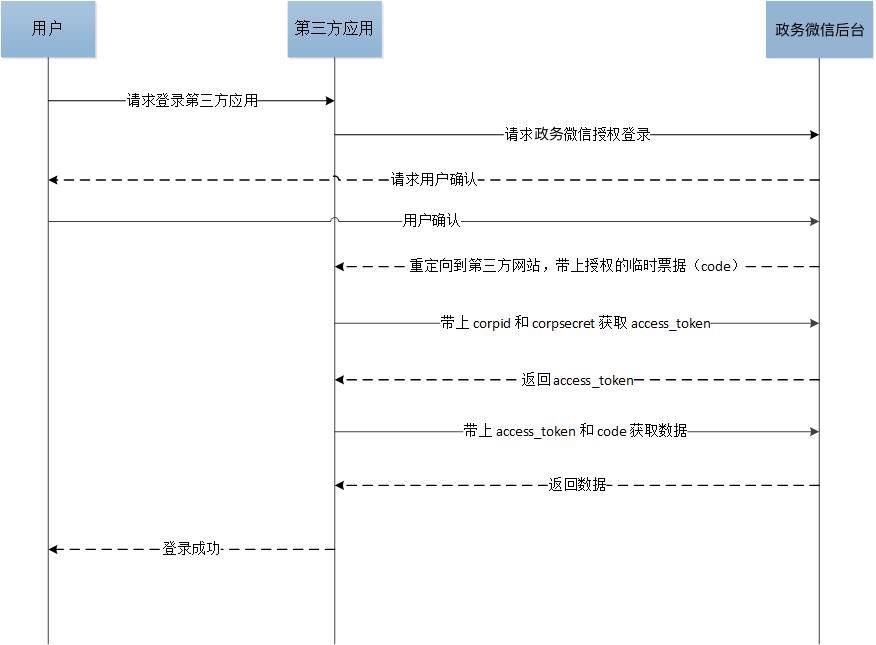
**授权流程说明**

企业微信授权登录让企业微信用户使用企业微信身份安全登录第三方应用或网站，在企业微信的用户授权登录已接入企业微信的第三方应用，通过code和access\_code可以进行企业微信后台授权关系接口调用，从而可实现获取企业微信用户基本开放信息和帮助用户实现基础开放功能等。

企业微信授权登录目前支持authorization\_code模式，适用于拥有server端的应用授权。该模式整体流程为：

1. 第三方发起微信授权登录请求，企业微信用户允许授权第三方应用后，企业微信会重定向到第三方网站，并且带上授权临时票据code参数；
2. 然后第三方应用通过 corpid 和 corpsecret 获取access\_code（见“开始开发”） ；
3. 第三方应用通过 access\_code 和 code 获取用户基本数据资源或帮助用户实现基本操作。

**时序图如下：**



### 开启网页接入

在单位的管理端后台，进入需要开启的目标应用，点击**企业微信授权登录**的“设置”，进入如下页面  


然后点击 “设置授权回调域”，输入回调域名，点击“保存”。

### 使用说明

**步骤一：请求code**

需要接入企业微信网页登录授权的页面加入一个跳转链接引导，链接地址如下：

<https://saastest2.gov.weixin.qq.com/wwopen/sso/qrConnect?appid=ww100000a5f2191&agentid=1000000&redirect_uri=http%3A%2F%2Fwww.oa.com&state=web_login@gyoss9>

若提示“该链接无法访问”，请检查参数是否填写错误，如redirect\_uri的域名与网页应用的可信域名不一致

**参数说明**

| 参数 | 必须 | 说明 |
| --- | --- | --- |
| appid | 是 | 企业微信的cropID，在 企业微信管理端->我的单位 中查看 |
| agentid | 是 | 授权方的网页应用ID，在具体的网页应用中查看 |
| redirect\_uri | 是 | 重定向地址，需要进行UrlEncode |
| state | 否 | 用于保持请求和回调的状态，授权请求后原样带回给第三方。该参数可用于防止csrf攻击（跨站请求伪造攻击），建议第三方带上该参数，可设置为简单的随机数加session进行校验 |

**返回说明**  
用户允许授权后，将会重定向到redirect\_uri的网址上，并且带上code和state参数

redirect\_uri?code=CODE&state=STATE

若用户禁止授权，则重定向后不会带上code参数，仅会带上state参数

redirect\_uri?state=STATE

为了满足网站更定制化的需求，我们还提供了第二种获取code的方式，支持网站将企业微信登录二维码内嵌到自己页面中，用户使用企业微信扫码授权后通过JS将code返回给网站。  
JS企业微信登录主要用途：网站希望用户在网站内就能完成登录，无需跳转到企业微信域下登录后再返回，提升企业微信登录的流畅性与成功率。 网站内嵌二维码企业微信登录JS实现办法：

**步骤一：引入JS文件**

在需要展示企业微信网页登录二维码的网站引入如下JS文件：  
<https://saastest2.gov.weixin.qq.com/js/sso/wwLogin-1.0.0.js>

**步骤二：在需要使用微信登录的地方实例以下JS对象**

1. window.WwLogin({
2. "id" : "wx\_reg",
3. "appid" : "",
4. "agentid" : "",
5. "redirect\_uri" :"",
6. "state" : "",
7. "href" : "",
8. });

**参数说明**

| 参数 | 必须 | 说明 |
| --- | --- | --- |
| id | 是 | 第三方页面显示二维码的容器id |
| appid | 是 | 企业微信的cropID，在 企业微信管理端->我的单位 中查看 |
| agentid | 是 | 授权方的网页应用ID，在具体的网页应用中查看 |
| redirect\_uri | 是 | 重定向地址，需要进行UrlEncode |
| state | 否 | 用于保持请求和回调的状态，授权请求后原样带回给第三方。该参数可用于防止csrf攻击（跨站请求伪造攻击），建议第三方带上该参数，可设置为简单的随机数加session进行校验 |
| href | 否 | 自定义样式链接，第三方可根据实际需求覆盖默认样式。详见文档底部FAQ |

**步骤二：通过code获取用户信息**

**请求方式：**GET  
**请求地址：**https://saastest2.gov.weixin.qq.com/cgi-bin/user/getuserinfo?access\_token=ACCESS\_TOKEN&code=CODE

**参数说明：**

| 参数 | 必须 | 说明 |
| --- | --- | --- |
| access\_token | 是 | 调用接口凭证 |
| code | 是 | 通过成员授权获取到的code，每次成员授权带上的code将不一样，code只能使用一次，5分钟未被使用自动过期 |

**权限说明：**  
跳转的域名须完全匹配单位内任一应用的可信域名。

**返回结果：**

1. {
2. "UserId":"USERID",
3. }

| 参数 | 说明 |
| --- | --- |
| UserId | 成员UserID |

**出错返回示例：**

1. {
2. "errcode"**: 40029,**
3. **"errmsg": "invalid code"**
4. **}**

### F.A.Q

**网站内嵌二维码企业微信登录JS代码中href字段作用？**  
答：如果第三方觉得企业微信团队提供的默认样式与自己的页面样式不匹配，可以自己提供样式文件来覆盖默认样式。举个例子，如第三方觉得默认二维码过大，可以提供相关css样式文件，并把链接地址填入href字段（只支持https协议的资源地址）：

1. .impowerBox .qrcode {width: 200px;}
2. .impowerBox .title {display: none;}
3. .impowerBox .info {width: 200px;}
4. .status\_icon {display：none !important}
5. .impowerBox .status {text-align: center;}

相关效果如下：  
 

# SDK

## SDK

企业微信提供了网页开发JS-SDK，用于开发体验更好的网页应用，同时提供了移动端登录、分享SDK，帮助企业实现登录帐号统一，内容的快速分享。

### SDK介绍

JS-SDK

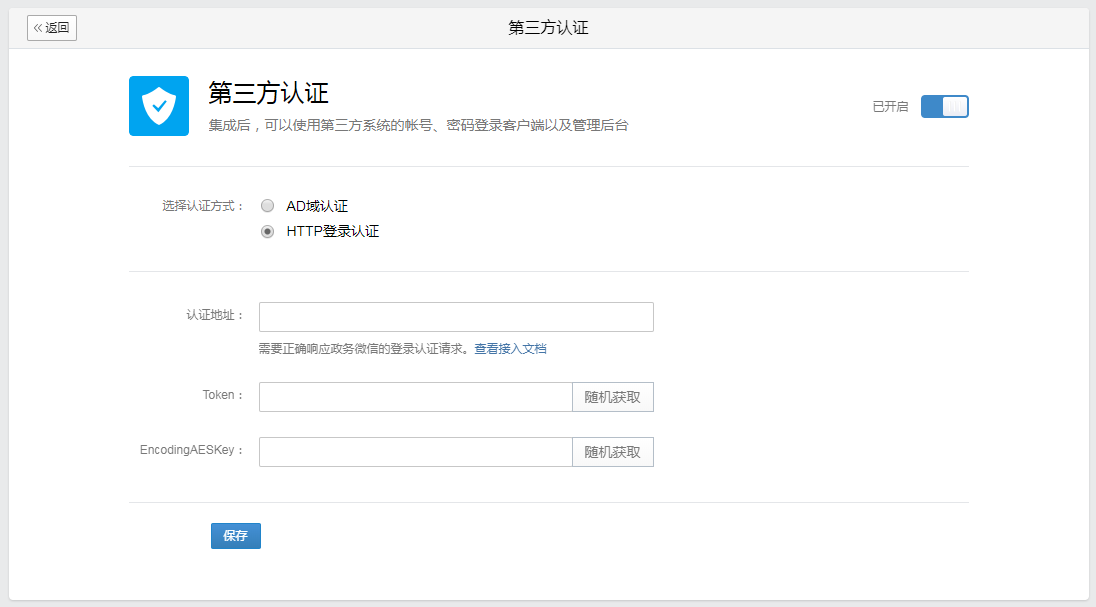
开发者可以使用网页开发JS-SDK，调用企业微信客户端Native的能力，开发体验更好的网页应用。

# 第三方认证

输入的认证地址要求能够处理如下HTTP请求：

**请求方法：**POST  
请求包和回包要求json格式

设置第三方认证的参数  
在单位的管理端后台，点击“管理工具”的“第三方认证”，进入如下页面



企业微信会要求你填写应用的URL、Token、EncodingAESKey三个参数，验证URL、Token以及加密的详细处理请参考[接收消息时的加解密处理](https://saastest2.gov.weixin.qq.com/api/doc/#10514/消息的加解密处理)章节。

请求包明文结构

1. {
2. "username":"USERNAME",
3. "password":"PASSWORD"
4. }

| 参数 | 必须 | 说明 |
| --- | --- | --- |
| username | 是 | 企业微信账号 |
| password | 是 | 企业微信明文密码 |

第三方服务器返回参数

返回：

1. {
2. "status":0,
3. "errmsg":""
4. }

| 参数 | 必须 | 说明 |
| --- | --- | --- |
| status | 是 | 验证结果，返回0表示验证通过，非0值表示验证失败 |
| errmsg | 否 | Utf8编码的错误信息，会在app登录失败的时候显示提示 |

**版本信息：**  
v1.1.1以上版本有效

# 短信邮件网关对接

## 短信邮件服务器配置

### 开启短信邮件对接

1、设置接收消息的参数  
在单位的管理端后台，进入管理工具，点击“短信邮件网关”的“设置”，进入如下页面  
  
需要选择或填写对接类型、Token、EncodingAESKey、邮件发件人四个参数，验证URL、Token以及加密的详细处理请参考接收消息时的加解密处理章节。

* 类型是选择对接短信或邮件的类型，如果只选择其中的一种方式，则改类型对接填写的地址发送，另外一种类型走腾讯的服务器发送。
* 网关URL是短信邮件网关服务器协议和地址，支持http或https协议。
* Token可由单位任意填写，用于生成签名。
* EncodingAESKey用于消息体的加密，是AES密钥的Base64编码。
* 邮件发件人是邮件发送者地址信息，该项若为空，则使用 [wework@tencent.com](mailto:wework@tencent.com) 作为默认发送者地址信息。

2、验证URL有效性  
当点击“保存”提交以上信息时，企业微信将发送GET请求到填写的URL上，GET请求携带四个参数，单位在获取时需要做Urldecode处理，否则会验证不成功，你可以访问接口调试工具进行调试，依次选择 建立连接 > 接收消息。

* 参数说明

| **参数** | **必须** | **说明** |
| --- | --- | --- |
| msg\_signature | 是 | 企业微信加密签名，msg\_signature结合了单位填写的token、请求中的timestamp、nonce参数、加密的消息体 |
| timestamp | 是 | 时间戳 |
| nonce | 是 | 随机数 |
| echostr | 是 | 加密的随机字符串，以msg\_encrypt格式提供。需要解密并返回echostr明文，解密后有random、msg\_len、msg、$CorpID四个字段，其中msg即为echostr明文 |

单位通过参数msg\_signature对请求进行校验，如果确认此次GET请求来自企业微信，那么单位应该对echostr参数解密并原样返回echostr明文(不能加引号，不能带bom头，不能带换行符)，则接入验证生效，接收消息才能开启。  
后续推送消息给单位时都会在请求URL中带上以上参数（echostr除外），校验方式与首次验证URL一致。

### 使用网关接口

私有化部署服务器后台在收到短信或邮件发送请求时，会对消息体本身做AES加密，以XML格式POST到单位的URL上。当单位服务器接收成功后，需在http头部返回200表示接收成功，其他错误一律当做失败并发起重试以，总共重试三次。  
假设单位的接收消息的URL设置为http://api.3dept.com。  
**请求地址 ：**

http://api.3dept.com/?msg\_signature=ASDFQWEXZCVAQFASDFASDFSS&timestamp=13500001234&nonce=123412323

**接收数据格式 ：**

1. <xml>
2. <Encrypt><![CDATA[msg\_encrypt]]</Encrypt>
3. </xml>
4. 其中：msg\_encrypt为经过加密的密文

单位需要对msg\_signature进行校验，并解密msg\_encrypt，得出msg的原文。详细处理请参考 接收消息时的加解密处理 章节。

**版本信息：**  
v1.2以上版本有效

## 短信邮件消息对接

当私有化部署服务后台收到短信或邮件通知信息的时候，会推送到单位设置的网关服务器上，假如单位设置的网关服务器的地址为：http://api.3dept.com/

**请求方式：**POST  
**请求地址：**http://api.3dept.com/?msg\_signature=ASDFQWEXZCVAQFASDFASDFSS&timestamp=13500001234&nonce=123412323

### 短信消息对接

请求示例：

1. <xml>
2. <MsgType><![CDATA[sms]]></MsgType>
3. <Content><![CDATA[mobile msg]]></Content>
4. <Mobile><![CDATA[13800000000]]></Mobile>
5. <CountryCode>![CDATA[+86]</CountryCode>
6. </xml>

参数说明：

| **参数** | **说明** |
| --- | --- |
| MsgType | 消息类型，此时固定为：sms |
| Content | 短信内容 |
| Mobile | 手机号码 |
| CountryCode | 手机号的国际码。如:中国香港地区为：+852 |

### 邮件消息对接

1. <xml>
2. <MsgType><![CDATA[email]]></MsgType>
3. <Content><![CDATA[eml msg]]></Content>
4. <Receive\_Email><![CDATA[13800000000@163.com]]></Receive\_Email>
5. <Sender\_Email>![CDATA[1229028612@qq.com]</Receive\_Email>
6. </xml>

参数说明：

| **参数** | **说明** |
| --- | --- |
| MsgType | 消息类型，此时固定为：email |
| Content | 标准的eml模板内容，模板包含邮件主题，发件人，收件人及邮件内容。 |
| Receive\_Email | 接收者的邮箱地址 |
| Sender\_Email | 发送者的邮箱地址 |

**版本信息：**  
v1.2以上版本有效

# 附录

## 加解密方案的详细说明

**术语及说明**

开启接收消息模式时，有以下术语需要了解：

* msg\_signature是签名，用于验证调用者的合法性。具体算法见以下’消息体签名’章节
* EncodingAESKey用于消息体的加密，长度固定为43个字符，从a-z, A-Z, 0-9共62个字符中选取，是AESKey的Base64编码。解码后即为32字节长的AESKey
* AESKey=Base64\_Decode(EncodingAESKey + “=”)，是AES算法的密钥，长度为32字节。AES采用CBC模式，数据采用PKCS#7填充至32字节的倍数；IV初始向量大小为16字节，取AESKey前16字节。具体详见：<http://tools.ietf.org/html/rfc2315>
* msg为消息体明文，格式为XML
* msg\_encrypt = Base64\_Encode( AES\_Encrypt[random(16B) + msg\_len(4B) + msg + $CorpID] )，是对明文消息msg加密处理后的Base64编码。其中random为16字节的随机字符串；msg\_len为4字节的msg长度，网络字节序；msg为消息体明文；$CorpID为企业微信的唯一标识

**消息体签名**

为了验证调用者的合法性，企业微信在接收消息的url中增加了消息签名，以参数msg\_signature标识，单位需要验证此参数的正确性后再解密。验证步骤：

* 单位计算签名：dev\_msg\_signature=sha1(sort(token、timestamp、nonce、msg\_encrypt))。sort的含义是将参数按照字母字典排序，然后从小到大拼接成一个字符串
* 比较dev\_msg\_signature和msg\_signature是否相等，相等则表示验证通过
* 在被动响应消息时，单位同样需要用如上方法生成签名并传给企业微信

**加解密方案说明**

**明文msg的加密过程**

msg\_encrypt = Base64\_Encode( AES\_Encrypt[random(16B) + msg\_len(4B) + msg + $CorpID] )  
AES加密的buf由16个字节的随机字符串、4个字节的msg长度、明文msg和$CorpID组成。其中msg\_len为msg的字节数，网络字节序；$CorpID为单位号的CorpID。经AESKey加密后，再进行Base64编码，即获得密文msg\_encrypt。

**和加密方案对应的解密方案**

* 对密文BASE64解码：aes\_msg=Base64\_Decode(msg\_encrypt)
* 使用AESKey做AES解密：rand\_msg=AES\_Decrypt(aes\_msg)
* 验证解密后$CorpID、msg\_len
* 去掉rand\_msg头部的16个随机字节，4个字节的msg\_len,和尾部的$CorpID即为最终的消息体原文msg

## 加解密库下载与返回码

### 加解密库的返回码

| **返回码** | **说明** |
| --- | --- |
| -40001 | 签名验证错误 |
| -40002 | xml解析失败 |
| -40003 | sha加密生成签名失败 |
| -40004 | AESKey 非法 |
| -40005 | corpid 校验错误 |
| -40006 | AES 加密失败 |
| -40007 | AES 解密失败 |
| -40008 | 解密后得到的buffer非法 |
| -40009 | base64加密失败 |
| -40010 | base64解密失败 |
| -40011 | 生成xml失败 |

### 加解密库下载及示例

#### c++库

[(2016年10月19日更新,点击下载)](https://saastest2.gov.weixin.qq.com/wwopen/downloadfile/c++.zip)

注意事项：

* WXBizMsgCrypt.h声明了WXBizMsgCrypt类，提供用户接入企业微信的三个接口。WXBizMsgCrypt.cpp文件提供了三个接口的实现。Sample.cpp文件提供了如何使用这三个接口的示例。
* WXBizMsgCrypt类封装了VerifyURL, DecryptMsg, EncryptMsg三个接口，分别用于开发者验证接收消息的url，收到用户回复消息的解密以及开发者回复消息的加密过程。使用方法可以参考Sample.cpp文件。
* 加解密协议请参考企业微信官方文档。
* 加解密过程使用了开源的openssl和tinyxml2库，请开发者自行安装之后使用。  
  openssl的版本号是openssl-1.0.1h，<http://www.openssl.org/>  
  tinyxml2的版本号是tinyxml2-2.1.0，<https://github.com/leethomason/tinyxml2>

#### python库

[(2016年10月19日更新,点击下载)](https://saastest2.gov.weixin.qq.com/wwopen/downloadfile/python.zip)

注意事项：

* WXBizMsgCrypt.py文件封装了WXBizMsgCrypt接口类，提供了用户接入企业微信的三个接口，Sample.py文件提供了如何使用这三个接口的示例，ierror.py提供了错误码。
* WXBizMsgCrypt封装了VerifyURL, DecryptMsg, EncryptMsg三个接口，分别用于开发者验证接收消息的url、接收消息的解密以及开发者回复消息的加密过程。使用方法可以参考Sample.py文件。
* 本代码用到了pycrypto第三方库，请开发者自行安装此库再使用。

#### php库

[(2016年10月19日更新,点击下载)](https://saastest2.gov.weixin.qq.com/wwopen/downloadfile/php.zip)

注意事项：

* WXBizMsgCrypt.php文件提供了WXBizMsgCrypt类的实现，是用户接入企业微信的接口类。Sample.php提供了示例以供开发者参考。errorCode.php, pkcs7Encoder.php, sha1.php, xmlparse.php文件是实现这个类的辅助类，开发者无须关心其具体实现。
* WXBizMsgCrypt类封装了VerifyURL, DecryptMsg, EncryptMsg三个接口，分别用于开发者验证接收消息的url、接收消息的解密以及开发者回复消息的加密过程。使用方法可以参考Sample.php文件。

#### java库

[(2016年10月19日更新,点击下载)](https://saastest2.gov.weixin.qq.com/wwopen/downloadfile/java.zip)

注意事项：

* com\qq\weixin\mp\aes目录下是用户需要用到的接入企业微信的接口，其中WXBizMsgCrypt.java文件提供的WXBizMsgCrypt类封装了用户接入企业微信的三个接口，其它的类文件用户用于实现加解密，用户无须关心。sample.java文件提供了接口的使用示例。
* WXBizMsgCrypt封装了VerifyURL, DecryptMsg, EncryptMsg三个接口，分别用于开发者验证接收消息的url、接收消息的解密以及开发者回复消息的加密过程。使用方法可以参考Sample.java文件。
* 请开发者使用jdk1.6或以上的版本。针对org.apache.commons.codec.binary.Base64，需要导入jar包commons-codec-1.9（或comm ons-codec-1.8等其他版本），我们有提供，官方下载地址：  
  <http://commons.apache.org/proper/commons-codec/download_codec.cgi>
* 异常java.security.InvalidKeyException:illegal Key Size的解决方案：在官方网站下载JCE无限制权限策略文件（请到官网下载对应的版本， 例如JDK7的下载地址：<http://www.oracle.com/technetwork/java/javase/downloads/jce-7-download-432124.html> )：下载后解压，可以看到local\_policy.jar和US\_export\_policy.jar以及readme.txt。  
  如果安装了JRE，将两个jar文件放到%JRE\_HOME% \lib\security目录下覆盖原来的文件，如果安装了JDK，将两个jar文件放到%JDK\_HOME%\jre\lib\security目录下覆盖原来文件。

#### c#库

[(2016年10月19日更新,点击下载)](https://saastest2.gov.weixin.qq.com/wwopen/downloadfile/csharp.zip)

注意事项：

* Cryptography.cs文件封装了AES加解密过程，用户无须关心具体实现。WXBizMsgCrypt.cs文件提供了用户接入企业微信的三个接口，Sample.cs文件提供了如何使用这三个接口的示例。
* WXBizMsgCrypt.cs封装了VerifyURL, DecryptMsg, EncryptMsg三个接口，分别用于开发者验证接收消息的url、接收消息的解密以及开发者回复消息的加密过程。使用方法可以参考Sample.cs文件。

## 接口代码参考示例

### 开启接收消息模式

开启接收消息模式的时候，企业微信会推送一个随机的加密数据到单位填写的接收消息地址后面，单位需要将密文解密后将明文返回到企业微信后台即可，示例如下：

1. <?php
2. //TODO:需要引入加密库脚本
3. require\_once "../lib/msgcrypt.php";
4. require\_once "../lib/helper.php";
5. //TODO:填写需要验证的应用配置信息
6. $token = $config->Token;
7. $corpId = $appConfigs->CorpId;
8. $encodingAesKey = $config->EncodingAESKey;
9. /\*
10. \* 单位开启接收消息模式时，单位号会向验证url发送一个get请求
11. \* 此逻辑需要先开通接收消息模式并将代码部署到服务器后进行验证
12. \*/
13. $sVerifyMsgSig = urldecode($\_GET["msg\_signature"]);
14. $sVerifyTimeStamp = urldecode($\_GET["timestamp"]);
15. $sVerifyNonce = urldecode($\_GET["nonce"]);
16. $sVerifyEchoStr = urldecode($\_GET["echostr"]);
17. // 需要返回的明文
18. $sEchoStr = "";
19. $wxcpt = new MsgCrypt($token, $encodingAesKey, $corpId);
20. $errCode = $wxcpt->VerifyURL($sVerifyMsgSig, $sVerifyTimeStamp, $sVerifyNonce, $sVerifyEchoStr, $sEchoStr);
21. if ($errCode == 0) {
22. // 验证URL成功，将sEchoStr返回
23. echo $sEchoStr;
24. exit(0);
25. } else {
26. print("ERR: " . $errCode . "\n\n");
27. }
28. ?>

### 通讯录接口

企业微信开放了两种权限，通讯录的读取以及编辑权限，下面以新增一个部门为例（其他接口与之类似）：

**第一步：**获取通讯录应用的access\_token

1. public function getAccessToken($corpid,$secret) {
2. //TODO: access\_token 应该全局存储与更新，以下代码以写入到文件中做示例
3. //TODO: 每个应用的access\_token应独立存储，此处用secret作为区分应用的标识
4. $path = "../cache/$secret.php";
5. $data = json\_decode(get\_php\_file($path));
6. if($data->expire\_time < time()) {
7. $url = "https://saastest2.gov.weixin.qq.com/cgi-bin/gettoken?corpid=$corpid&corpsecret=$secret";
8. $res = json\_decode(http\_get($url)["content"]);
9. $access\_token = $res->access\_token;
10. if($access\_token) {
11. $data->expire\_time = time() + 7000;
12. $data->access\_token = $access\_token;
13. set\_php\_file($path, json\_encode($data));
14. }
15. } else {
16. $access\_token = $data->access\_token;
17. }
18. return $access\_token;
19. }

**第二步：**封装数据包发送到企业微信后台

1. $info = $array("name" => "新增部门名称",
2. "parentid" => 1,
3. "order" => 10,
4. "id" => 12);
5. //TODO：此处传入你的单位corpid以及通讯录应用的secret信息
6. $txl\_access\_token = getAccessToken("corpid","txl\_secret");
7. print(http\_post("https://saastest2.gov.weixin.qq.com/cgi-bin/department/create?access\_token=$txl\_access\_token",$info);

### 应用管理

企业微信为单位提供了动态控制应用上下线以及菜单项的能力，下面以动态更新应用配置信息为例：  
**第一步：**获取自定义应用的access\_token  
参见通讯录接口的access\_token获取方式

**第二步：**封装数据包发送到企业微信后台

1. $info = $array("agentid" => "agentid", //TODO:配置需要更新的应用ID
2. "report\_location\_flag" => 0,
3. "name" => "HR助手",
4. "description" => "单位好助手",
5. "redirect\_domain" => "https://saastest2.gov.weixin.qq.com",
6. "isreportenter" => 0,
7. "home\_url" => "http://www.qq.com"
8. );
9. //TODO:此处传入你的单位的corpid以及对应应用的secret信息
10. $app\_access\_token = getAccessToken("corpid","app\_secret");
11. print(http\_post("https://saastest2.gov.weixin.qq.com/cgi-bin/agent/set?access\_token=$app\_access\_token",$info));

### 消息服务

消息服务可以为单位成员与单位后台之间建立一个双向通信的能力，如果单位需要主动给成员推送消息，步骤如下：  
**第一步：**获取自定义应用的access\_token  
参见通讯录接口的access\_token获取方式

**第二步：**封装数据包发送到企业微信后台

1. $msg = array(
2. 'touser'=>'UserID',
3. 'toparty'=>'1',
4. 'msgtype'=>'news',
5. 'agentid'=>1000007,
6. 'news'=>array(
7. "articles"=> array(0=>array(
8. "title"=>"中秋节礼品领取",
9. "description"=>"今年中秋节公司有豪礼相送",
10. "url"=>"http://qq.com", "picurl"=>"/wwopen/api/static/embed\_image/load/test\_pic\_msg1.png"
11. ))
12. )
13. );
14. $app\_access\_token = getAccessToken("corpid","app\_secret");
15. print(http\_post("https://saastest2.gov.weixin.qq.com/cgi-bin/message/send?access\_token=$app\_access\_token",$msg));

**接收用户消息处理逻辑示例：**

1. function receiveMsgFromQyWx(){
2. //TODO:配置接收消息的应用相关信息
3. $token = $config->Token;
4. $encodingAesKey = $config->EncodingAESKey;
5. $corpId = $appConfigs->CorpId;
6. $sReqMsgSig = $\_GET["msg\_signature"];
7. $sReqTimeStamp = $\_GET["timestamp"];
8. $sReqNonce = $\_GET["nonce"];
9. $sReqData = file\_get\_contents("php://input");
10. $sMsg = ""; // 解析之后的明文
11. $wxcpt = new MsgCrypt($token,$encodingAesKey,$corpId);
12. $errCode = $wxcpt->DecryptMsg($sReqMsgSig, $sReqTimeStamp, $sReqNonce, $sReqData, $sMsg);
13. if ($errCode == 0) {
14. // 解密成功，sMsg即为xml格式的明文
15. $xml = new DOMDocument();
16. $xml->loadXML($sMsg);
17. //取出发送消息的UserID
18. $FromUserName = $xml->getElementsByTagName('FromUserName')->item(0)->nodeValue;
19. //取出发送的消息内容体
20. $content = $xml->getElementsByTagName('Content')->item(0)->nodeValue;
21. //TODO ... 业务逻辑
22. } else {
23. print("ERR: " . $errCode . "\n\n");
24. }
25. }

**单位被动回复用户消息示例：**

1. function replyMsgToUser(){
2. // 需要发送的明文消息
3. // TODO：根据用户提交过来的操作封装此数据包
4. $sRespData = "<xml>
5. <ToUserName><![CDATA[mycreate]]></ToUserName>
6. <FromUserName><![CDATA[wx5823bf96d3bd56c7]]></FromUserName>
7. <CreateTime>1348831860</CreateTime>
8. <MsgType><![CDATA[text]]></MsgType>
9. <Content><![CDATA[this is a test]]></Content>
10. </xml>";
11. $sEncryptMsg = ""; //xml格式的密文
12. $wxcpt = new MsgCrypt($token,$encodingAesKey,$corpId);
13. $errCode = $wxcpt->EncryptMsg($sRespData, $sReqTimeStamp, $sReqNonce, $sEncryptMsg);
14. if ($errCode == 0) {
15. // 加密成功，单位需要将加密之后的sEncryptMsg返回
16. // TODO:向企业微信的后台回复消息
17. } else {
18. print("ERR: " . $errCode . "\n\n");
19. // exit(-1);
20. }
21. }

### 网页开发

如果你需要在应用的网页里面使用企业微信提供的强大jsapi能力，那么你可以这样在你的网页中使用：  
**第一步：**获取自定义应用的access\_token  
参见通讯录接口的access\_token获取方式

**第二步：**获取自定义应用的jsapi\_ticket

1. function getJsApiTicket($corpid,$secret) {
2. // jsapi\_ticket 应该全局存储与更新，以下代码以写入到文件中做示例
3. $path = "../cache/jsapi\_ticket.php";
4. $data = json\_decode(get\_php\_file($path));
5. $app\_access\_token = getAccessToken($corpid,$secret);
6. if($data->expire\_time < time()){
7. $accessToken = $this->accessToken->getAccessToken();
8. $url = "https://saastest2.gov.weixin.qq.com/cgi-bin/get\_jsapi\_ticket?access\_token=$app\_access\_token";
9. $res = json\_decode(http\_get($url)["content"]);
10. $ticket = $res->ticket;
11. if ($ticket) {
12. $data->expire\_time = time() + 7000;
13. $data->jsapi\_ticket = $ticket;
14. set\_php\_file($path, json\_encode($data));
15. }
16. } else {
17. $ticket = $data->jsapi\_ticket;
18. }
19. return $ticket;
20. }

**第三步：**获取自定义应用的签名对象

1. function getSignPackage() {
2. //TODO:填写你需要生成签名的单位ID以及应用密钥
3. $jsapiTicket = getJsApiTicket("corpid","app\_secret");
4. // 注意 URL 一定要动态获取，不能 hardcode.
5. $protocol = (!empty($\_SERVER['HTTPS']) && $\_SERVER['HTTPS'] !== 'off' || $\_SERVER['SERVER\_PORT'] == 443) ? "https://" : "http://";
6. $url = "$protocol$\_SERVER[HTTP\_HOST]$\_SERVER[REQUEST\_URI]";
7. $timestamp = time();
8. $nonceStr = createNonceStr();
9. //这里参数的顺序要按照 key 值 ASCII 码升序排序
10. $string = "jsapi\_ticket=$jsapiTicket&noncestr=$nonceStr&timestamp=$timestamp&url=$url";
11. $signature = sha1($string);
12. $signPackage = array(
13. "appId" => $this->appId,
14. "nonceStr" => $nonceStr,
15. "timestamp" => $timestamp,
16. "url" => $url,
17. "signature" => $signature,
18. "rawString" => $string
19. );
20. return $signPackage;
21. }

**第四步：**在页面上完成引用配置

1. <?php
2. require\_once "../lib/jssdk.php";
3. //第一步: 生成JSAPI签名
4. $jssdk = getSignPackage();
5. $signPackage = $jssdk->GetSignPackage();
6. ?>
7. <!DOCTYPE html>
8. <html lang="en">
9. <head>
10. <meta charset="UTF-8">
11. <title>JSAPI 签名输出测试</title>
12. </head>
13. <bopdy>
14. </body>
15. <!-- 第二步: 引用JSAPI 的脚本文件 -->
16. <script src="https://saastest2.gov.weixin.qq.com/wework\_admin/jweixin"></script>
17. <script>
18. /\*
19. \* 第三步: 配置jsapi的权限
20. \* 注意：所有的JS接口只能可信域名下调用
21. \*/
22. wx.config({
23. debug: true,
24. appId: '<?php echo $signPackage["appId"];?>', //此处的appId等同于单位的CorpID
25. timestamp: <?php echo $signPackage["timestamp"];?>,
26. nonceStr: '<?php echo $signPackage["nonceStr"];?>',
27. signature: '<?php echo $signPackage["signature"];?>',
28. jsApiList: [
29. // 所有要调用的 API 都要加到这个列表中
30. ]
31. });
32. wx.ready(function () {
33. //TODO： 执行和jsapi相关的初始化操作
34. });
35. </script>
36. </html>

### 代码示例下载

示例代码包含了开发过程中常用的操作，比如获取不同应用的access\_token、接收消息模式设置、jsapi的调用、通讯录管理、应用管理、消息服务、媒体上传等功能模块，[点击下载PHP版本的示例](https://saastest2.gov.weixin.qq.com/api/downloadfile/OpenApiSample.zip)

## 全局错误码

单位应用每次调用接口时，可能获得正确或错误的返回码，单位可以根据返回码信息调试接口，排查错误。

全局返回码说明如下：

返回码 说明  
**特殊说明：{“errcode”:-1,”errmsg”:”Invalid input”} 是指请求参数格式有问题， 比如整型传成string, string传成整型， 请严格按照文档修正参数格式。**

-1 系统繁忙  
0 请求成功  
40001 获取access\_token时Secret错误，或者access\_token无效  
40002 不合法的凭证类型  
40003 不合法的UserID  
40004 不合法的媒体文件类型  
40005 不合法的文件类型  
40006 不合法的文件大小  
40007 不合法的媒体文件id  
40008 不合法的消息类型  
40013 不合法的corpid  
40014 不合法的access\_token  
40015 不合法的菜单类型  
40016 不合法的按钮个数  
40017 不合法的按钮类型  
40018 不合法的按钮名字长度  
40019 不合法的按钮KEY长度  
40020 不合法的按钮URL长度  
40021 不合法的菜单版本号  
40022 不合法的子菜单级数  
40023 不合法的子菜单按钮个数  
40024 不合法的子菜单按钮类型  
40025 不合法的子菜单按钮名字长度  
40026 不合法的子菜单按钮KEY长度  
40027 不合法的子菜单按钮URL长度  
40028 不合法的自定义菜单使用成员  
40029 不合法的oauth\_code  
40031 不合法的UserID列表  
40032 不合法的UserID列表长度  
40033 不合法的请求字符，不能包含\uxxxx格式的字符  
40035 不合法的参数  
40038 不合法的请求格式  
40039 不合法的URL长度  
40040 不合法的插件token  
40041 不合法的插件id  
40042 不合法的插件会话  
40048 url中包含不合法domain  
40054 不合法的子菜单url域名  
40055 不合法的按钮url域名  
40056 不合法的agentid  
40057 不合法的callbackurl或者callbackurl验证失败  
40058 不合法的红包参数  
40059 不合法的上报地理位置标志位  
40060 设置上报地理位置标志位时没有设置callbackurl  
40061 设置应用头像失败  
40062 不合法的应用模式  
40063 参数为空  
40064 管理组名字已存在  
40065 不合法的管理组名字长度  
40066 不合法的部门列表  
40067 标题长度不合法  
40068 不合法的标签ID  
40069 不合法的标签ID列表  
40070 列表中所有标签（成员）ID都不合法  
40071 不合法的标签名字，标签名字已经存在  
40072 不合法的标签名字长度  
40073 不合法的openid  
40074 news消息不支持指定为高保密消息  
40077 不合法的预授权码  
40078 不合法的临时授权码  
40079 不合法的授权信息  
40080 不合法的suitesecret  
40082 不合法的suitetoken  
40083 不合法的suiteid  
40084 不合法的永久授权码  
40085 不合法的suiteticket  
40086 不合法的第三方应用appid  
40092 导入文件存在不合法的内容  
40093 不合法的跳转target  
40094 不合法的URL  
40095 修改失败，并发冲突  
40155 请勿添加其他公众号的主页链接  
41001 缺少access\_token参数  
41002 缺少corpid参数  
41003 缺少refresh\_token参数  
41004 缺少secret参数  
41005 缺少多媒体文件数据  
41006 缺少media\_id参数  
41007 缺少子菜单数据  
41008 缺少oauth code  
41009 缺少UserID  
41010 缺少url  
41011 缺少agentid  
41012 缺少应用头像mediaid  
41013 缺少应用名字  
41014 缺少应用描述  
41015 缺少Content  
41016 缺少标题  
41017 缺少标签ID  
41018 缺少标签名字  
41021 缺少suiteid  
41022 缺少suitetoken  
41023 缺少suiteticket  
41024 缺少suitesecret  
41025 缺少永久授权码  
41034 缺少login\_ticket  
41035 缺少跳转target  
42001 access\_token过期  
42002 refresh\_token过期  
42003 oauth\_code过期  
42004 插件token过期  
42007 预授权码失效  
42008 临时授权码失效  
42009 suitetoken失效  
43001 需要GET请求  
43002 需要POST请求  
43003 需要HTTPS  
43004 需要成员已关注  
43005 需要好友关系  
43006 需要订阅  
43007 需要授权  
43008 需要支付授权  
43010 需要处于接收消息模式  
43011 需要单位授权  
43013 应用对成员不可见  
44001 多媒体文件为空  
44002 POST的数据包为空  
44003 图文消息内容为空  
44004 文本消息内容为空  
45001 多媒体文件大小超过限制  
45002 消息内容大小超过限制  
45003 标题大小超过限制  
45004 描述大小超过限制  
45005 链接长度超过限制  
45006 图片链接长度超过限制  
45007 语音播放时间超过限制  
45008 图文消息的文章数量不能超过8条  
45009 接口调用超过限制  
45010 创建菜单个数超过限制  
45015 回复时间超过限制  
45016 系统分组，不允许修改  
45017 分组名字过长  
45018 分组数量超过上限  
45022 应用名字长度不合法，合法长度为2-16个字  
45024 帐号数量超过上限  
45025 同一个成员每周只能邀请一次  
45026 触发删除用户数的保护  
45027 mpnews每天只能发送100次  
45028 素材数量超过上限  
45029 media\_id对该应用不可见  
45032 作者名字长度超过限制  
46001 不存在媒体数据  
46002 不存在的菜单版本  
46003 不存在的菜单数据  
46004 不存在的成员  
47001 解析JSON/XML内容错误  
48001 Api未授权  
48002 Api禁用(一般是管理组类型与Api不匹配，例如普通管理组调用会话服务的Api)  
48003 suitetoken无效  
48004 授权关系无效  
48005 Api已废弃  
50001 redirect\_uri未授权  
50002 成员不在权限范围  
50003 应用已停用  
50004 成员状态不正确，需要成员为单位验证中状态  
50005 单位已禁用  
60001 部门长度不符合限制  
60002 部门层级深度超过限制  
60003 部门不存在  
60004 父部门不存在  
60005 不允许删除有成员的部门  
60006 不允许删除有子部门的部门  
60007 不允许删除根部门  
60008 部门ID或者部门名称已存在  
60009 部门名称含有非法字符  
60010 部门存在循环关系  
60011 权限不足，user/department/agent无权限(只有通迅录同步助手才有通迅录写权限，同时要开启写权限 )  
60012 不允许删除默认应用  
60013 不允许关闭应用  
60014 不允许开启应用  
60015 不允许修改默认应用可见范围  
60016 不允许删除存在成员的标签  
60017 不允许设置单位  
60019 不允许设置应用地理位置上报开关  
60020 访问ip不在白名单之中  
60023 已授权的应用不允许单位管理组调用接口修改菜单  
60025 主页型应用不支持的消息类型  
60027 不支持第三方修改主页型应用字段  
60028 应用已授权予第三方，不允许通过接口修改主页url  
60029 应用已授权予第三方，不允许通过接口修改可信域名  
60031 未设置管理组的登录授权域名  
60102 UserID已存在  
60103 手机号码不合法  
60104 手机号码已存在  
60105 邮箱不合法  
60106 邮箱已存在  
60107 微信号不合法  
60108 微信号已存在  
60109 QQ号已存在  
60110 用户同时归属部门超过20个  
60111 UserID不存在  
60112 成员姓名不合法  
60113 身份认证信息（微信号/手机/邮箱）不能同时为空  
60114 性别不合法  
60115 已关注成员微信不能修改  
60116 扩展属性已存在  
60118 成员无有效邀请字段，详情参考(邀请成员关注)的接口说明  
60119 成员已关注  
60120 成员已禁用  
60121 找不到该成员  
60122 邮箱已被外部管理员使用  
60123 无效的部门id  
60124 无效的父部门id  
60125 非法部门名字，长度超过限制、重名等，重名包括与csv文件中同级部门重名或者与旧组织架构包含成员的同级部门重名  
60126 创建部门失败  
60127 缺少部门id  
60128 字段不合法，可能存在主键冲突或者格式错误  
60129 用户设置了拒绝邀请  
60131 不合法的职位长度  
80001 可信域名不匹配，或者可信域名没有IPC备案（后续将不能在该域名下正常使用jssdk）  
81003 邀请额度已用完  
81004 部门数量超过上限  
82001 发送消息或者邀请的参数全部为空或者全部不合法  
82002 不合法的PartyID列表长度  
82003 不合法的TagID列表长度  
82004 微信版本号过低  
85002 包含不合法的词语  
86001 不合法的会话ID  
86003 不存在的会话ID  
86004 不合法的会话名  
86005 不合法的会话管理员  
86006 不合法的成员列表大小  
86007 不存在的成员  
86101 需要会话管理员权限  
86201 缺少会话ID  
86202 缺少会话名  
86203 缺少会话管理员  
86204 缺少成员  
86205 非法的会话ID长度  
86206 非法的会话ID数值  
86207 会话管理员不在用户列表中  
86208 消息服务未开启  
86209 缺少操作者  
86210 缺少会话参数  
86211 缺少会话类型（单聊或者群聊）  
86213 缺少发件人  
86214 非法的会话类型  
86215 会话已存在  
86216 非法会话成员  
86217 会话操作者不在成员列表中  
86218 非法会话发件人  
86219 非法会话收件人  
86220 非法会话操作者  
86221 单聊模式下，发件人与收件人不能为同一人  
86222 不允许消息服务访问的API  
86304 不合法的消息类型  
86305 客服服务未启用  
86306 缺少发送人  
86307 缺少发送人类型  
86308 缺少发送人id  
86309 缺少接收人  
86310 缺少接收人类型  
86311 缺少接收人id  
86312 缺少消息类型  
86313 缺少客服，发送人或接收人类型，必须有一个为kf  
86314 客服不唯一，发送人或接收人类型，必须只有一个为kf  
86315 不合法的发送人类型  
86316 不合法的发送人id。Userid不存在、openid不存在、kf不存在  
86317 不合法的接收人类型  
86318 不合法的接收人id。Userid不存在、openid不存在、kf不存在  
86319 不合法的客服，kf不在客服列表中  
86320 不合法的客服类型  
88001 缺少seq参数  
88002 缺少offset参数  
88003 非法seq  
90001 未认证摇一摇周边  
90002 缺少摇一摇周边ticket参数  
90003 摇一摇周边ticket参数不合法  
90004 摇一摇周边ticket过期  
90005 未开启摇一摇周边服务  
91004 卡券已被核销  
91011 无效的code  
91014 缺少卡券详情  
91015 代金券缺少least\_cost或者reduce\_cost参数  
91016 折扣券缺少discount参数  
91017 礼品券缺少gift参数  
91019 缺少卡券sku参数  
91020 缺少卡券有效期  
91021 缺少卡券有效期类型  
91022 缺少卡券logo\_url  
91023 缺少卡券code类型  
91025 缺少卡券title  
91026 缺少卡券color  
91027 缺少offset参数  
91028 缺少count参数  
91029 缺少card\_id  
91030 缺少卡券code  
91031 缺少卡券notice  
91032 缺少卡券description  
91033 缺少ticket类型  
91036 不合法的有效期  
91038 变更库存值不合法  
91039 不合法的卡券id  
91040 不合法的ticket type  
91041 没有创建，上传卡券logo，以及核销卡券的权限  
91042 没有该卡券投放权限  
91043 没有修改或者删除该卡券的权限  
91044 不合法的卡券参数  
91045 缺少团购券groupon结构  
91046 缺少现金券cash结构  
91047 缺少折扣券discount 结构  
91048 缺少礼品券gift结构  
91049 缺少优惠券coupon结构  
91050 缺少卡券必填字段  
91051 商户名称超过12个汉字  
91052 卡券标题超过9个汉字  
91053 卡券提醒超过16个汉字  
91054 卡券描述超过1024个汉字  
91055 卡券副标题长度超过18个汉字

301001 应用id已存在  
301002 accesstoken不允许操作其它应用。  
301004 不允许删除超级管理员  
301005 消息型应用不允许做此操作  
301006 不允许禁用超级管理员  
301008 主页型应用不允许做此操作  
301009 应用发送消息没有接收主体  
301010 部门名已存在  
301013 座机不合法  
301014 英文名称不合法  
301021 userid错误  
301022 获取打卡数据失败  
301023 useridlist非法或超过限额  
301024 获取打卡记录时间间隔超限

302001 批量同步成员存在userid为空的用户  
302002 管理员userid不存在  
302003 存在重复的userid  
302004 组织架构不合法（**1不是一棵树，2 多个一样的partyid，3 partyid空，4 partyid name 空，5 同一个父节点下有两个子节点 部门名字一样** 可能是以上情况，请一一排查）

500006 license过期

## 联系我们

当你在开发过程中遇到问题时，建议你先阅读接口相关文档，如果不能解决你的问题，欢迎通过下列邮箱联系我们。

* 地址：[wxwork-open@qq.com](mailto:wxwork-open@qq.com)
* 邮件主题：［企业微信开发问题反馈］

为方便定位问题，我们建议你在邮件中提供更多信息，包括但不限于：

* 公司名称、corpid
* 问题机型
* 操作系统
* 问题描述
* 是否必现
* 问题截图
* 关键代码截图

我们收到邮件后会第一时间处理并给予回复。