# XXX通用API协议

**V0.1.0**

**修改记录**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **文件编号** | **版本号** | **拟制人/**  **修改人** | **拟制/**  **修改日期** | **更改理由** | **主要更改内容**  **（写要点即可）** |
|  | V0.1.0 | XXX | 2020-2-18 | 新建 |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
| 注：文件第一次拟制时，“更改理由”、“主要更改内容”栏写“无”。 | | | | | |

## 接口规则

### 协议规则

接入本平台服务，调用API必须遵循以下规则：

|  |  |
| --- | --- |
| 传输方式 | 为保证数据安全性，采用HTTPS传输 |
| 提交方式 | 采用POST方法提交 |
| 数据格式 | 提交和返回数据都为JSON格式 |
| 字符编码 | 统一采用UTF-8字符编码 |
| 签名算法 | MD5/HMAC-SHA256 |
| 签名要求 | 请求和接收数据均需要校验签名，详细方法请参考[安全规范-签名算法](https://pay.weixin.qq.com/wiki/doc/api/micropay.php?chapter=4_3" \t "/home/svenaugustus/Documents\\x/_blank) |
| 判断逻辑 | 先判断协议字段返回，再判断业务返回，最后判断交易状态 |

### 参数规定

1、appId

在本平台中唯一表示某一个业务系统，接口中用于区分系统来源。

2、tenantId

在本平台中唯一表示某一个租户，接口中用于区分租户来源。

3、transactionId

租户唯一表示API流水，字母和数字组成的最大32位字符串。

4、时间戳

标准北京时间，时区为东八区，自1970年1月1日 0点0分0秒以来的秒数。注意：部分系统取到的值为毫秒级，需要转换成秒(10位数字)。

### 安全规范

#### 1、签名算法

签名生成的通用步骤如下：

第一步，设所有发送或者接收到的数据为集合M，将集合M内非空参数值的参数按照参数名ASCII码从小到大排序（字典序），使用URL键值对的格式（即key1=value1&key2=value2…）拼接成字符串stringA。

特别注意以下重要规则：

◆ 参数名ASCII码从小到大排序（字典序）；

◆ 如果参数的值为空不参与签名；

◆ 参数名区分大小写；

◆ 验证调用返回或本平台主动通知签名时，传送的**sign、signType**参数不参与签名，将生成的签名与该sign值作校验。

◆ 本平台API接口可能增加字段，验证签名时必须支持增加的扩展字段

第二步，在stringA最后拼接上key得到stringSignTemp字符串，并对stringSignTemp进行MD5运算，再将得到的字符串所有字符转，得到sign值signValue。

◆ key设置路径：由本平台提供

举例：

假设传送的参数如下：

userId : 100098

userType : staff

appId:ctgd678efh567hg6787

tenantId:ctg\_hn

transactionId:1009660380201506130728806387

timestamp:202002181459

nonceStr:C380BEC2BFD727A4B6845133519F3AD6

signType:HMAC-SHA256

第一步：对参数按照key=value的格式，并按照参数名ASCII字典序排序如下：

stringA="appId=ctgd678efh567hg6787&nonceStr=C380BEC2BFD727A4B6845133519F3AD6&tenantId=ctg\_hn&timestamp=202002181459&transactionId=1009660380201506130728806387&userId=100098&userType=staff";

第二步：拼接API密钥：

stringSignTemp=stringA+"&key=本平台" //注：key为本平台提供的密钥key

sign=MD5(stringSignTemp)="15acb2bdbe48ec6a9a71ca9d859903cd" //注：MD5签名方式

sign=SHA256(stringSignTemp,key)="7e139ffa4ad62be0b46b3deae6577ffa258f4126d9b444b6e1e77481ab285f41" //注：SHA256签名方式

sign=hash\_hmac("sha256",stringSignTemp,key)="6c21e7b3cc2f9e7ec5ae552155ed70baebf8c4b4ba95924c5a4ff626650cc262" //注：HMAC-SHA256签名方式

最终得到最终发送的数据：

{

"appId": "ctgd678efh567hg6787",

"tenantId": "ctg\_hn",

"timestamp": "202002181459",

"transactionId": "1009660380201506130728806387",

"nonceStr": "C380BEC2BFD727A4B6845133519F3AD6",

"sign": "6c21e7b3cc2f9e7ec5ae552155ed70baebf8c4b4ba95924c5a4ff626650cc262",

"signType": "HMAC-SHA256",

"userType": "staff",

"userId": "100098"

}

#### 2、生成随机数算法

本平台API接口协议中包含字段nonceStr，主要保证签名不可预测。我们推荐生成随机数算法如下：调用随机数函数生成，将得到的值转换为字符串。

# 云台席通用API协议

**V0.6.0**

**修改记录**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **文件编号** | **版本号** | **拟制人/**  **修改人** | **拟制/**  **修改日期** | **更改理由** | **主要更改内容**  **（写要点即可）** |
|  | V0.1.0 | 蔡政滦 | 2020-2-18 | 新建 |  |
|  | V0.2.0 | 蔡政滦 | 2020-2-19 | 修改 | 补充反向API列表 |
|  | V0.3.0 | 丁晓涛 | 2020-2-19 | 修改 | 补充与CRM对接的对比认证，支付反向接口 |
|  | V0.4.0 | 丁晓涛 | 2020-2-20 | 修改 | 补充营业员服务信息查询和认证信息查询正向接口，去掉对比认证反向接口 |
|  | V0.5.5 | 蔡政滦 | 2020-2-21 | 修改 | 补充支付链接推送、免填单无纸化链接推送接口，去掉反向-支付接口，增加支付结果查询接口 |
|  | V0.6.0 | 蔡政滦 | 2020-2-22 | 修改 | 修改签名算法说明 |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
| 注：文件第一次拟制时，“更改理由”、“主要更改内容”栏写“无”。 | | | | | |

## 接口规则

### 协议规则

接入云台席服务，调用API必须遵循以下规则：

|  |  |
| --- | --- |
| 传输方式 | 为保证数据安全性，采用HTTPS传输 |
| 提交方式 | 采用POST方法提交 |
| 数据格式 | 提交和返回数据都为JSON格式 |
| 字符编码 | 统一采用UTF-8字符编码 |
| 签名算法 | MD5/SHA256 |
| 签名要求 | 请求和接收数据均需要校验签名，详细方法请参考[安全规范-签名算法](https://pay.weixin.qq.com/wiki/doc/api/micropay.php?chapter=4_3" \t "/home/svenaugustus/Documents\\x/_blank) |
| 判断逻辑 | 先判断协议字段返回，再判断业务返回，最后判断交易状态 |

### 参数规定

1、appId

在云台席平台中唯一表示某一个业务系统，接口中用于区分系统来源。

2、tenantId

在云台席平台中唯一表示某一个租户，接口中用于区分租户来源。

3、transactionId

租户唯一表示API流水，字母和数字组成的最大32位字符串。

4、时间戳

标准北京时间，时区为东八区，自1970年1月1日 0点0分0秒以来的秒数。注意：部分系统取到的值为毫秒级，需要转换成秒(10位数字)。

### 安全规范

#### 1、签名算法

签名生成的通用步骤如下：

第一步，设所有发送或者接收到的数据为集合M，将集合M内非空参数值的参数按照参数名ASCII码从小到大排序（字典序），使用URL键值对的格式（即key1=value1&key2=value2…）拼接成字符串stringA。

特别注意以下重要规则：

◆ 参数名ASCII码从小到大排序（字典序）；

◆ 如果参数的值为空不参与签名；

◆ 参数名区分大小写；

◆ 验证调用返回或云台席主动通知签名时，传送的**sign、signType**参数不参与签名，将生成的签名与该sign值作校验。

◆ 云台席API接口可能增加字段，验证签名时必须支持增加的扩展字段

第二步，在stringA最后拼接上key得到stringSignTemp字符串，并对stringSignTemp进行MD5运算，再将得到的字符串所有字符，得到sign值signValue。

◆ key设置路径：由云台席提供

举例：

假设传送的参数如下：

userId : 100098

userType : staff

appId:ctgd678efh567hg6787

tenantId:ctg\_hn

transactionId:1009660380201506130728806387

timestamp:202002181459

nonceStr:C380BEC2BFD727A4B6845133519F3AD6

signType:SHA256

第一步：对参数按照key=value的格式，并按照参数名ASCII字典序排序如下：

stringA="appId=ctgd678efh567hg6787&nonceStr=C380BEC2BFD727A4B6845133519F3AD6&tenantId=ctg\_hn&timestamp=202002181459&transactionId=1009660380201506130728806387&userId=100098&userType=staff";

第二步：拼接API密钥：

stringSignTemp=stringA+"&key=xxx" //注：key为云台席提供的密钥key

sign=MD5(stringSignTemp)="15acb2bdbe48ec6a9a71ca9d859903cd" //注：MD5签名方式

sign=hash\_hmac("sha256",stringSignTemp,key)="7e139ffa4ad62be0b46b3deae6577ffa258f4126d9b444b6e1e77481ab285f41" //注：SHA256签名方式

最终得到最终发送的数据：

{

"appId": "ctgd678efh567hg6787",

"tenantId": "ctg\_hn",

"timestamp": "202002181459",

"transactionId": "1009660380201506130728806387",

"nonceStr": "C380BEC2BFD727A4B6845133519F3AD6",

"sign": "7e139ffa4ad62be0b46b3deae6577ffa258f4126d9b444b6e1e77481ab285f41",

"signType": "SHA256",

"userType": "staff",

"userId": "100098"

}

#### 2、生成随机数算法

云台席API接口协议中包含字段nonceStr，主要保证签名不可预测。我们推荐生成随机数算法如下：调用随机数函数生成，将得到的值转换为字符串。

#### 3、Java工具类的示例代码

参考以下代码：  
import java.util.HashMap;  
import java.util.Map;  
import org.apache.commons.lang3.RandomStringUtils;  
  
public class Test {  
  
 public static void main(String[] args) {  
 // 应用ID, 向云台席申请，云台席提供给业务系统固定使用  
 String appId = "ctgd678efh567hg6787";  
 // 租户ID, 向云台席申请，云台席提供给业务系统固定使用   
 String tenantId = "ctg\_hn";  
 // 应用密钥，：云台席提供的密钥  
 String key = "xxxx";  
  
 Map<String, String> paramObj = new HashMap<>(16);  
 paramObj.**put**("appId", appId);  
 paramObj.**put**("tenantId", tenantId);  
 // api 唯一流水号， 保证业务调用唯一，用于排查和记录  
 paramObj.**put**("transactionId", "1009660380201506130728806387");  
 // 14位的时间戳，yyyyMMddHHmmss 格式  
 paramObj.**put**("timestamp", "20200218145945");  
 // 随机数  
 String randonStr = RandomStringUtils.randomAlphanumeric(32);  
 paramObj.**put**("nonceStr", randonStr);  
 paramObj.**put**("userType", "staff");  
 paramObj.**put**("userId", "100098");  
  
 // 签名算法 目前支持SHA256、HMAC-SHA256 和 MD5，默认为MD5， 推荐使用 HMAC-SHA256  
 String signType = "MD5";  
 SecretSpi secretSpi = null;  
  
 if ("MD5".**equals**(signType)) {  
 secretSpi = SignatureUtil.**MD5**;  
 }  
 else if ("SHA256".**equals**(signType)) {  
 secretSpi = SignatureUtil.**SHA256**;  
 }  
 else if ("HMAC-SHA256".**equals**(signType)) {  
 MAC mac = new MAC();  
 mac.**setKey**(key);  
 secretSpi = mac;  
 }  
  
 Map<String, String> params = new HashMap<String, String>(paramObj);  
 String sign = SignatureUtil.sign(params, key, "sign", secretSpi);  
 // 签名  
 paramObj.**put**("sign", sign);  
 paramObj.**put**("signType", signType);  
  
 // 转换为 JSON, 比如以下是 fastjson的示例  
 // JSONObject requestJson = (JSONObject) JSON.toJSON(paramObj);  
 // System.out.println(requestJson.toString());  
 }  
}

## 正向API列表（？->本平台）

### 单点登录

#### 应用场景

该接口提供业务系统以员工或客户身份单点登录。

#### 接口链接

URL地址：https://本平台服务地址/login/sso

#### 请求参数

| 字段名 | 变量名 | 必填 | 类型 | 示例值 | 描述 |
| --- | --- | --- | --- | --- | --- |
| 应用ID | appId | 是 | String(32) | ctgd678efh567hg6787 | 本平台分配的应用ID |
| 租户ID | tenantId | 是 | String(32) | ctg\_hn | 本平台分配的租户ID |
| 流水号 | transactionId | 是 | String(32) | 1009660380201506130728806387 | API流水，唯一表示一笔业务请求 |
| 请求时间戳 | timestamp | 是 | String(17) | 202002181459 | 时间戳,年月日时分秒 |
| 随机字符串 | nonceStr | 是 | String(32) | C380BEC2BFD727A4B6845133519F3AD6 | 随机字符串，不长于32位。推荐**[随机数生成算法](https://pay.weixin.qq.com/wiki/doc/api/micropay.php?chapter=4_3" \t "/home/svenaugustus/Documents\\x/_blank)** |
| 签名 | sign | 是 | String(32) | 5K8264ILTKCH16CQ2502SI8ZNMTM67VS | 通过签名算法计算得出的签名值，[详见](https://pay.weixin.qq.com/wiki/doc/api/micropay.php?chapter=4_3" \t "/home/svenaugustus/Documents\\x/_blank)**[签名生成算法](https://pay.weixin.qq.com/wiki/doc/api/micropay.php?chapter=4_3" \t "/home/svenaugustus/Documents\\x/_blank)** |
| 签名类型 | signType | 否 | String(32) | HMAC-SHA256 | 签名类型，目前支持HMAC-SHA256和MD5，默认为MD5 |
| 用户类型 | userType | 是 | String(32) | staff | 客户customer,  员工 staff |
| 用户标识 | userId | 是 | String(32) | 100098 | 客户送客户ID，  员工送工号ID |

举例如下：

{

"appId": "ctgd678efh567hg6787",

"tenantId": "ctg\_hn",

"timestamp": "202002181459",

"transactionId": "1009660380201506130728806387",

"nonceStr": "C380BEC2BFD727A4B6845133519F3AD6",

"sign": "6c21e7b3cc2f9e7ec5ae552155ed70baebf8c4b4ba95924c5a4ff626650cc262",

"signType": "HMAC-SHA256",

"userType": "staff",

"userId": "100098"

}

#### 返回结果

| 字段名 | 变量名 | 必填 | 类型 | 示例值 | 描述 |
| --- | --- | --- | --- | --- | --- |
| 返回状态码 | returnCode | 是 | String(16) | SUCCESS | SUCCESS/FAIL此字段是接口通信情况标识，非交易成功与否的标识 |
| 返回信息 | returnMsg | 是 | String(128) | 处理成功 | 当return\_code为FAIL时返回信息为错误原因 ，例如  签名失败  参数格式校验错误 |

当returnCode为SUCCESS的时候，还会包括以下字段：

| 字段名 | 变量名 | 必填 | 类型 | 示例值 | 描述 |
| --- | --- | --- | --- | --- | --- |
| 应用ID | appId | 是 | String(32) | ctgd678efh567hg6787 | 本平台分配的应用ID |
| 租户ID | tenantId | 是 | String(32) | ctg\_hn | 本平台分配的租户ID |
| 接口流水ID | apiTransId | 是 | String(32) | 1009660380201506130728806387 | 对应云台席的接口日志ID |
| 响应时间戳 | timestamp | 是 | String(17) | 20200218145959 | 时间戳,年月日时分秒 |
| 随机字符串 | nonceStr | 是 | String(32) | C380BEC2BFD727A4B6845133519F3AD6 | 随机字符串，不长于32位。推荐**[随机数生成算法](https://pay.weixin.qq.com/wiki/doc/api/micropay.php?chapter=4_3" \t "/home/svenaugustus/Documents\\x/_blank)** |
| 签名 | sign | 是 | String(32) | 5K8264ILTKCH16CQ2502SI8ZNMTM67VS | 通过签名算法计算得出的签名值，[详见](https://pay.weixin.qq.com/wiki/doc/api/micropay.php?chapter=4_3" \t "/home/svenaugustus/Documents\\x/_blank)**[签名生成算法](https://pay.weixin.qq.com/wiki/doc/api/micropay.php?chapter=4_3" \t "/home/svenaugustus/Documents\\x/_blank)** |
| 签名类型 | signType | 否 | String(32) | HMAC-SHA256 | 签名类型，目前支持HMAC-SHA256和MD5，默认为MD5 |
| 业务结果 | resultCode | 是 | String(16) | SUCCESS | SUCCESS/FAIL |
| 错误代码 | errCode | 否 | String(32) | SYSTEMERROR | 详细参见错误列表 |
| 错误代码描述 | errCodeDes | 否 | String(128) | 系统错误 | 错误返回的信息描述 |

当returnCode 和resultCode都为SUCCESS的时，还会包括以下字段：

| 字段名 | 变量名 | 必填 | 类型 | 示例值 | 描述 |
| --- | --- | --- | --- | --- | --- |
| 鉴权Token | accessToken | 是 | String(32) | afac2215f5dc6wisdom5bae2a0665ef6 | 用于访问本平台的鉴权Token |

举例如下：

{

"returnCode": "SUCCESS",

"returnMsg": "处理成功",

"appId": "ctgd678efh567hg6787",

"tenantId": "ctg\_hn",

"nonceStr": "sBGiv4bgBEE88cXPTUYkApQ7WiXD87k0",

"timestamp": "20200220113032",

"sign": "d4ded85261eadd93d4bc18a47e2475740e638985f7c704826be99480706c13ab",

"signType": "HMAC-SHA256",

"resultCode": "SUCCESS",

"accessToken": "xxxxx"

}

#### 错误码

## 反向API列表（本平台->？）

### 员工服务状态通知

#### 应用场景

员工开始/结束服务时，向业务系统推送开始服务的通知。

#### 接口链接

在本平台配置对应租户和应用需要通知的URL。

通知url必须为直接可访问的url，不能携带参数。

示例：https://xxx.test.com/staff/startNotify.do

#### 通知参数

| 字段名 | 变量名 | 必填 | 类型 | 示例值 | 描述 |
| --- | --- | --- | --- | --- | --- |
| 返回状态码 | returnCode | 是 | String(16) | SUCCESS | SUCCESS/FAIL  此字段是接口通信情况标识，非交易成功与否的标识 |
| 返回信息 | returnMsg | 是 | String(128) | 处理成功 | 当return\_code为FAIL时返回信息为错误原因 ，例如  签名失败  参数格式校验错误 |

以下字段在returnCode为SUCCESS的时候有返回：

| 字段名 | 变量名 | 必填 | 类型 | 示例值 | 描述 |
| --- | --- | --- | --- | --- | --- |
| 应用ID | appId | 是 | String(32) | ctgd678efh567hg6787 | 本平台分配的应用ID |
| 租户ID | tenantId | 是 | String(32) | ctg\_hn | 本平台分配的租户ID |
| 流水号 | transactionId | 是 | String(32) | 1009660380201506130728806387 | API流水，唯一表示一笔业务请求 |
| 请求时间戳 | timestamp | 是 | String(17) | 202002181459 | 时间戳,年月日时分秒 |
| 随机字符串 | nonceStr | 是 | String(32) | C380BEC2BFD727A4B6845133519F3AD6 | 随机字符串，不长于32位。推荐**[随机数生成算法](https://pay.weixin.qq.com/wiki/doc/api/micropay.php?chapter=4_3" \t "/home/svenaugustus/Documents\\x/_blank)** |
| 签名 | sign | 是 | String(32) | 5K8264ILTKCH16CQ2502SI8ZNMTM67VS | 通过HMAC-SHA256签名算法计算得出的签名值，[详见](https://pay.weixin.qq.com/wiki/doc/api/micropay.php?chapter=4_3" \t "/home/svenaugustus/Documents\\x/_blank)**[签名生成算法](https://pay.weixin.qq.com/wiki/doc/api/micropay.php?chapter=4_3" \t "/home/svenaugustus/Documents\\x/_blank)** |
| 员工工号 | staffId | 是 | String(32) | 10098 | 工号ID |
| 服务状态 | status | 是 | String(32) | online | 开始服务 online，  结束服务 offline |

举例如下：

{

"appId": "ctgd678efh567hg6787",

"tenantId": "ctg\_hn",

"nonceStr": "FNhPEX3UtEbYAIBoofup34eipLJAec69",

"transactionId": "547299962019631104",

"timestamp": "20200219141439",

"sign": "a6bb0ed30367ae3b8bcb613cf6f386da438b0239aeef083822b6f83d049d4c86",

"staffId": "100098",

"status": "online"

}

#### 返回参数

| 字段名 | 变量名 | 必填 | 类型 | 示例值 | 描述 |
| --- | --- | --- | --- | --- | --- |
| 返回状态码 | returnCode | 是 | String(16) | SUCCESS | SUCCESS/FAIL  此字段是接口通信情况标识，非交易成功与否的标识 |
| 返回信息 | returnMsg | 是 | String(128) | 处理成功 | 当return\_code为FAIL时返回信息为错误原因 ，例如  签名失败  参数格式校验错误 |

举例如下：

{

"returnCode": "SUCCESS",

"returnMsg": "处理成功"

}