

华为云会议

开发指南

文档版本 03
发布日期 2025-09-04



版权所有 © 华为技术有限公司 2025。保留一切权利。

非经本公司书面许可，任何单位和个人不得擅自摘抄、复制本文档内容的部分或全部，并不得以任何形式传播。

商标声明



HUAWEI和其他华为商标均为华为技术有限公司的商标。

本文档提及的其他所有商标或注册商标，由各自的所有人拥有。

注意

您购买的产品、服务或特性等应受华为公司商业合同和条款的约束，本文档中描述的全部或部分产品、服务或特性可能不在您的购买或使用范围之内。除非合同另有约定，华为公司对本文档内容不做任何明示或暗示的声明或保证。

由于产品版本升级或其他原因，本文档内容会不定期进行更新。除非另有约定，本文档仅作为使用指导，本文档中的所有陈述、信息和建议不构成任何明示或暗示的担保。

目录

| | |
|--------------------|----|
| 1 开放性简介..... | 1 |
| 2 集成华为云会议账号..... | 6 |
| 3 App ID 鉴权介绍..... | 9 |
| 4 开发流程..... | 17 |
| 5 最佳实践..... | 22 |
| 6 注意事项..... | 27 |
| 7 提交问题工单..... | 28 |

1 开放性简介

华为云会议的开放能力

华为云会议提供服务端开放集成和客户端开放集成能力，开发者可以根据业务的需要使用对应的接口集成开发。

- 服务端集成：华为云会议提供服务端REST接口开放，包括企业管理、用户管理、通讯录管理、会议管理、会议控制等能力。第三方业务系统（包括业务后台或者终端App）可以调用相关接口实现自己的业务逻辑。
- 客户端SDK集成：华为云会议提供了客户端SDK开放，开发者可以在自己开发的App上集成华为云会议的视频会议能力，实现创会、入会、会控等功能。SDK中已经集成了入会后的UI，第三方App只需调用少量接口就可以在自己的App中集成华为云会议的视频能力。
- 客户端Scheme应用集成：华为云会议客户端提供URL Protocol定义好的参数，第三方应用（包括Web页面或者终端App）可以直接拉起华为云会议客户端并且入会。

图 1-1 华为云会议开放性介绍

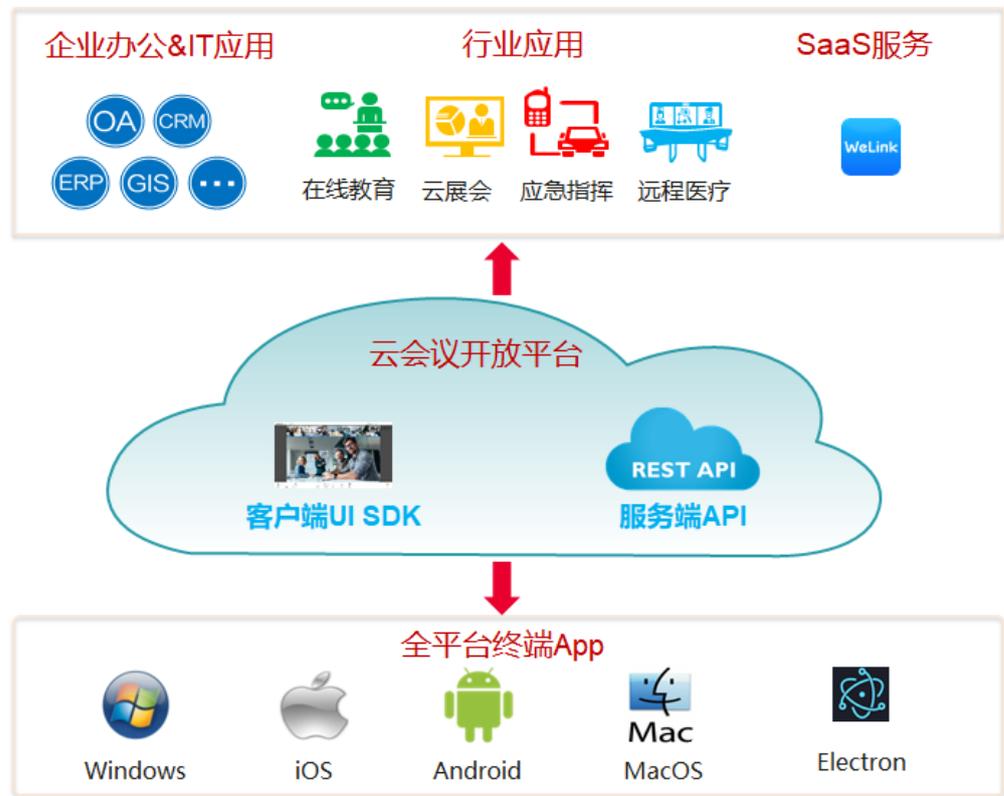


表 1-1 华为云会议开放性能力

| 开放能力分类 | 开放能力 | 开放能力详情 |
|---------|--------|---|
| 服务端开放能力 | 企业管理 | <ul style="list-style-type: none"> • SP管理员增、删、改、查企业 • SP管理员管理企业资源 |
| | 用户管理 | <ul style="list-style-type: none"> • 部门增、删、改、查 • 用户增、删、改、查 • 用户密码管理 • 企业通讯录查询 |
| | 云会议室管理 | <ul style="list-style-type: none"> • 云会议室分配、修改、回收、查询 |
| | 硬终端管理 | <ul style="list-style-type: none"> • 硬终端激活、查询、删除 |
| | 会议管理 | <ul style="list-style-type: none"> • 会议/周期性会议创建、修改、查询 • 历史会议查询 |
| | 会议控制 | <ul style="list-style-type: none"> • 会中对与会人邀请、挂断、静音等 • 会中对多画面设置 |
| | | |

| 开放能力分类 | 开放能力 | 开放能力详情 |
|---------------|-----------------|---|
| | 会议信息推送 | 单会议状态推送： <ul style="list-style-type: none"> • 与会者状态推送 • 最大发言方推送 • 会议质量推送 • 实时字幕推送 • 会议状态推送 企业级会议状态推送： <ul style="list-style-type: none"> • 会议开始状态 • 会议结束状态 • 会议关闭状态 |
| | 仪表盘 | <ul style="list-style-type: none"> • 会议QoS查询 • 会议统计 |
| | 信息窗管理 | <ul style="list-style-type: none"> • 信息窗发布管理 • 信息窗节目管理 • 信息窗素材管理 |
| 客户端UI SDK开放能力 | Android UI SDK | <ul style="list-style-type: none"> • 创建会议、加入会议、匿名入会 • 预约会议、编辑会议、取消会议、获取会议详情、会议VMR列表 • 点对点呼叫 |
| | iOS UI SDK | <ul style="list-style-type: none"> • 创建会议、加入会议、匿名入会 • 预约会议 • 点对点呼叫 |
| | Windows UI SDK | <ul style="list-style-type: none"> • 创建会议、加入会议、匿名入会 • 预约会议、编辑会议、取消会议、获取会议详情、会议VMR列表 • 点对点呼叫 |
| | Mac UI SDK | <ul style="list-style-type: none"> • 创建会议、加入会议、匿名入会 • 预约会议、编辑会议、取消会议、获取会议详情、会议VMR列表 • 点对点呼叫 |
| | Electron UI SDK | <ul style="list-style-type: none"> • 创建会议、加入会议、匿名入会 • 预约会议、编辑会议、取消会议、获取会议详情、会议VMR列表 • 点对点呼叫 |
| | | |

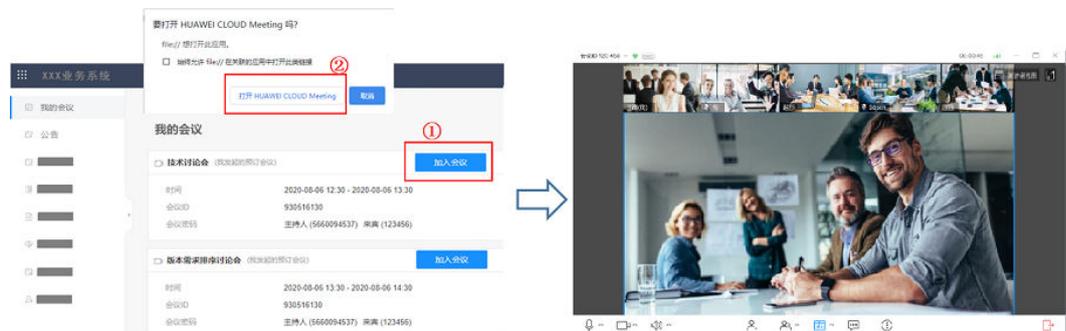
| 开放能力分类 | 开放能力 | 开放能力详情 |
|-------------------|----------------------|---|
| 客户端Scheme 开放能力 | 华为云会议 Android App | <ul style="list-style-type: none"> 拉起App 拉起App并登录 加入会议 拉起App并加入会议 |
| | 华为云会议iOS App | <ul style="list-style-type: none"> 拉起App 拉起App并登录 加入会议 拉起App并加入会议 |
| | 华为云会议 Windows客户端 | <ul style="list-style-type: none"> 拉起App 拉起App并登录 加入会议 拉起App并加入会议 |
| | 华为云会议Mac客 户端 | <ul style="list-style-type: none"> 拉起App 拉起App并登录 加入会议 拉起App并加入会议 |

集成场景 1：第三方系统预约会议，并拉起华为云会议客户端加入会议

第三方应用可以调用华为云会议服务端的REST API预约会议，并且在页面上加入跳转到拉起华为云会议客户端的URL，实现单击入会的效果。

该场景适用于第三方需要在自己的业务系统中管理和调度会议，并且允许用户使用华为云会议客户端参加会议。

图 1-2 终端 Scheme 应用集成



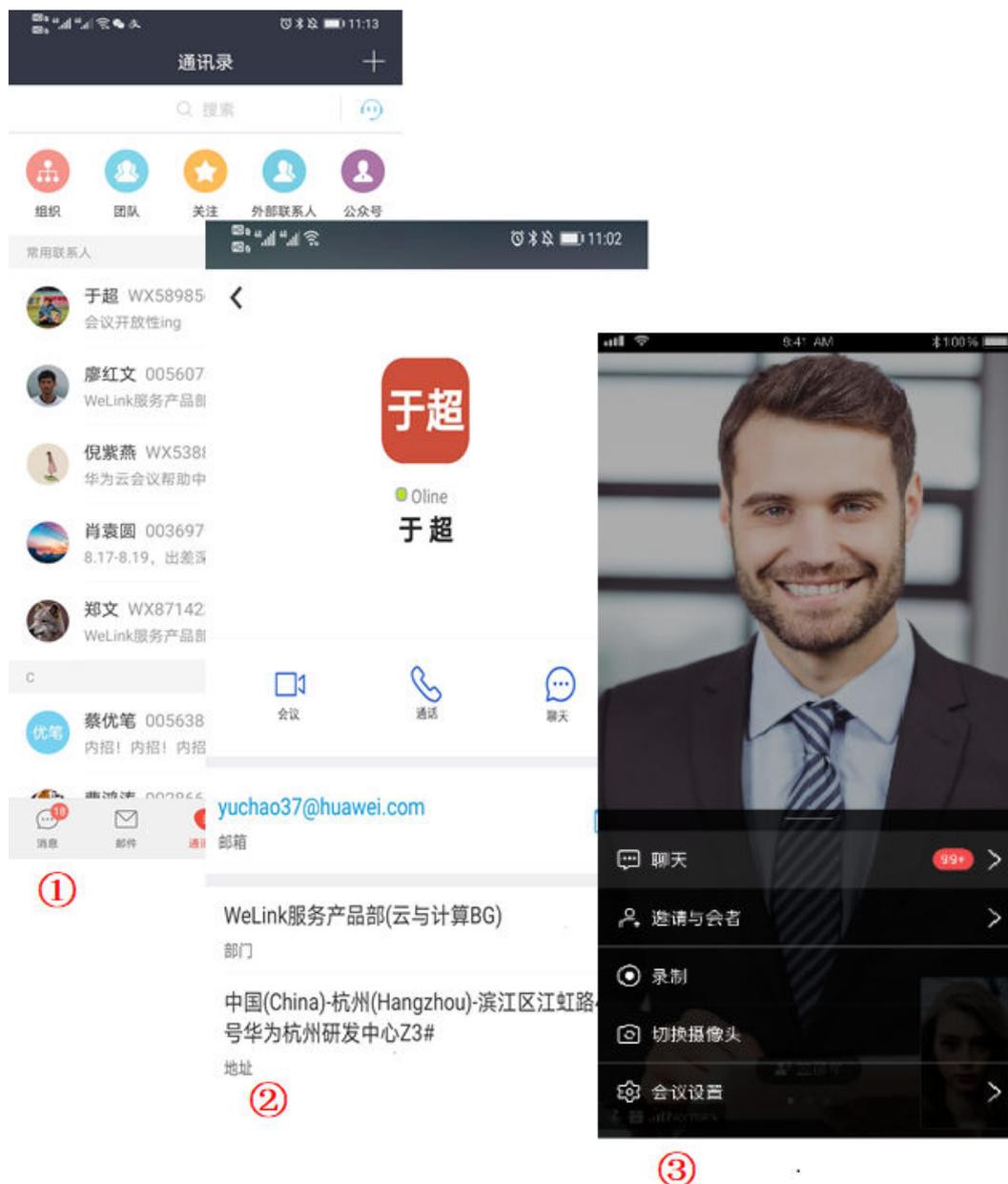
REST API的详细介绍参考《[服务端API参考](#)》。拉起会议客户端的URL的详细介绍参考《[终端Scheme应用集成](#)》。

集成场景 2：第三方 APP 通过 UI SDK 集成，快速集成会议客户端能力

第三方应用有自己的App，通过在App上调用会议客户端UI SDK接口，快速集成会议功能。只需要在客户的App上开发对应的入口并调用对应的接口预约会议、创建会议、加入会议，入会后的会控操作都是SDK提供的界面，无需新开发。

该场景适用于第三方需要将会议能力集成到自己的App内，并且UI SDK提供的会中界面满足使用要求。

图 1-3 第三方 App 基于终端 UI SDK 集成场景



上图中①②③的界面都在同一个App中，其中①②是第三方自己的App界面，③是SDK提供的界面。SDK的详细介绍参考《[客户端SDK参考](#)》。

2 集成华为云会议账号

华为云会议系统涉及两类账号：华为云账号和华为云会议账号。华为云会议账号按角色分类又可以分成三种：SP管理员账号，企业管理员账号和企业普通用户账号。

- 华为云账号：用于购买会议服务；登录华为云控制台；申请App ID；登录API Explorer进行调试。
- 华为云会议账号：
 - SP管理员账号：用于企业管理、企业资源管理。
 - 企业管理员账号：用于企业内部部门管理、用户管理、VMR管理、会管、会控。
 - 企业普通用户账号：用于用户本身的会管、会控。

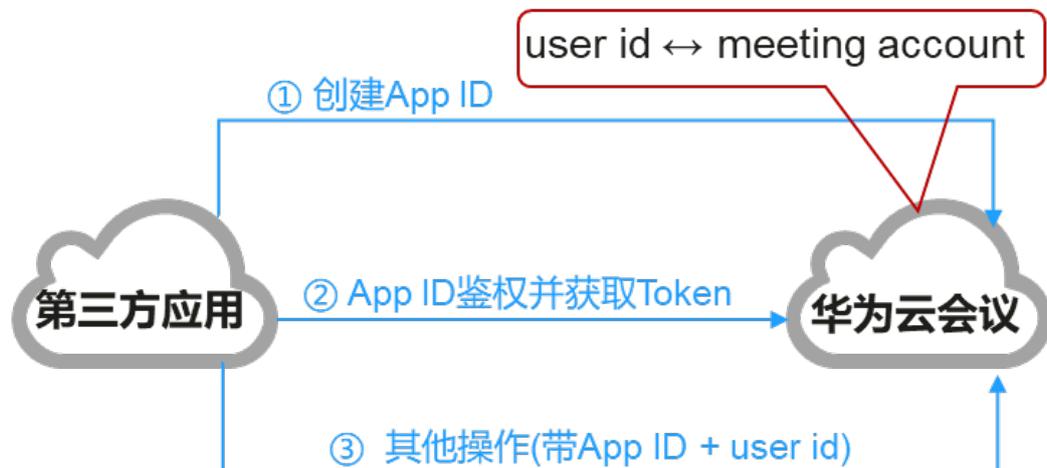
华为云会议提供两种账号集成方式与第三方应用集成：App ID鉴权方式；账号/密码鉴权方式。

📖 说明

- 申请企业管理员账号请参考“[开发前准备](#)”。App ID的申请请参考“[App ID的申请](#)”。

App ID 鉴权方式（推荐）

图 2-1 App ID 鉴权方式集成



- 步骤1** 预先创建好App ID，一个应用只需创建一次，请参考“[App ID的申请](#)”。
- 步骤2** 使用App ID和第三方User ID，调用鉴权接口获取接入token。
- 步骤3** 调用其他需指定用户的管理接口（比如邀请与会者）时带上接入Token，以及第三方User ID。

----结束

这种鉴权模式下第三方应用不感知华为云会议账号。第三方User ID和华为云会议账号的绑定关系在华为云会议系统中维护。

账号/密码鉴权方式（不推荐）

图 2-2 账号/密码鉴权方式集成

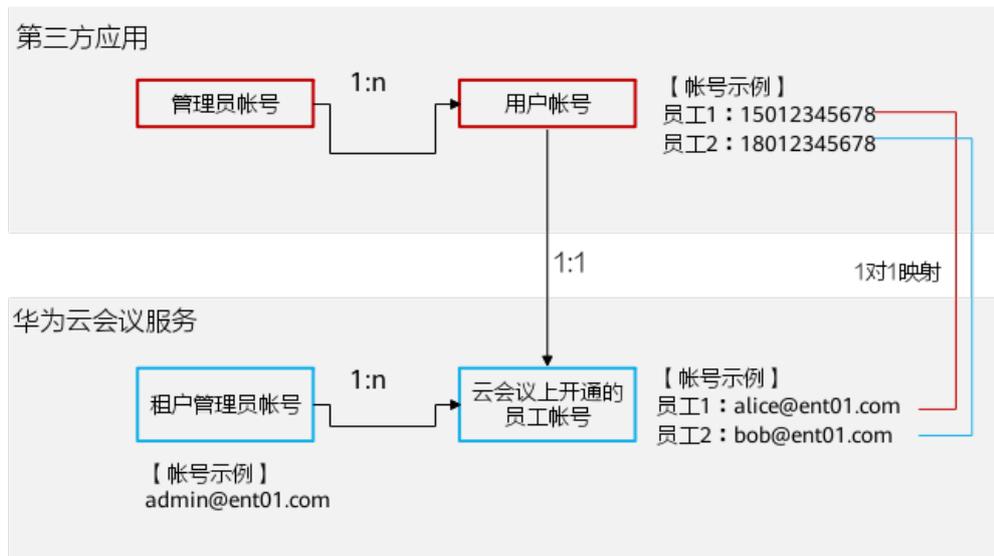


- 步骤1** 使用账号/密码鉴权方式与华为云会议账号集成时，需要调用服务端的REST API创建用户账号，并且在第三方系统中维护第三方的user id和华为云会议账号的绑定关系。
- 步骤2** 使用已经发放好的华为云会议账号和密码，调用鉴权接口获取接入Token。
- 步骤3** 调用其他需指定用户的管理接口（比如邀请与会者）时带上接入Token，以及华为云会议账号（account）。

----结束

这种鉴权模式下，第三方系统需要为第三方User ID绑定华为云会议用户账号，第三方User ID和华为云会议用户账号是1:1映射的。

图 2-3 第三方账号与华为云会议账号的绑定关系



📖 说明

第三方User ID和华为云会议用户账号也可以是n:1，但是需要第三方系统自己负责账号的绑定和解绑定。

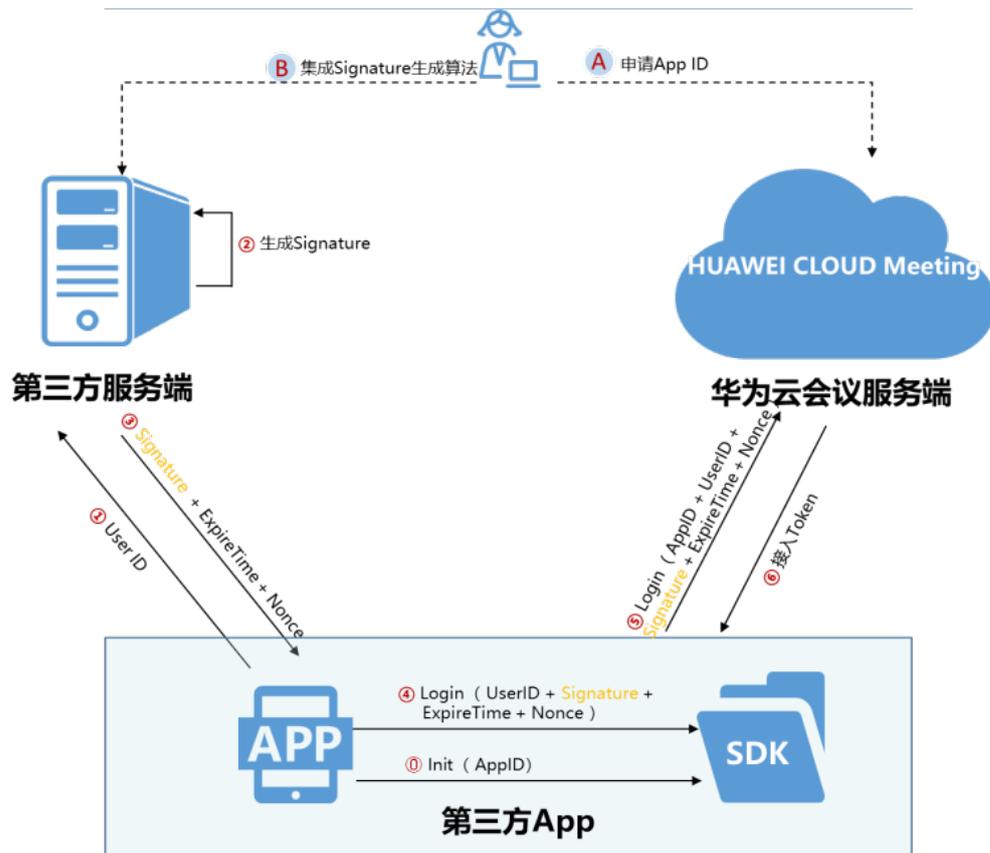
3 App ID 鉴权介绍

为了降低第三方应用集成华为云会议难度，并且提升开放接口的安全性，华为云会议开放能力中支持基于App ID的鉴权方式。App ID是一个应用的标识，同一个App ID可以同时也在第三方的桌面终端、移动终端、Web应用上使用。

App ID 鉴权原理

1. 第三方客户端App ID鉴权原理

图 3-1 第三方客户端 App ID 鉴权流程



鉴权前提:

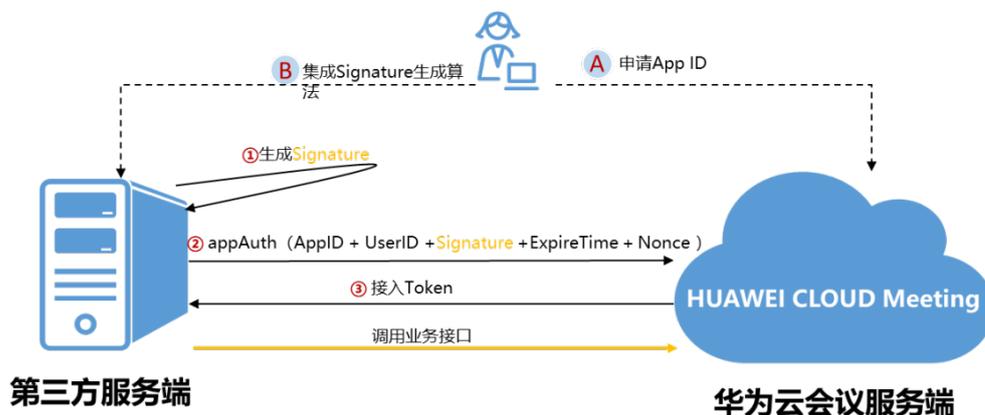
- a. 开发者在[华为云会议控制台](#)上为自己的应用申请App ID，并获取App ID和App Key。
- b. 开发者在自己的服务端集成Signature生成算法，请参考“[第三方服务集成Signature生成算法](#)”。
- c. SDK初始化时传入App ID，请参考《[客户端SDK参考](#)》中的“初始化”部分。

鉴权过程：

- a. 第三方客户端将User ID（第三方User ID）发送给第三方服务器。
- b. 第三方服务器根据App ID、User ID、App Key等，生成鉴权用的签名材料Signature。
- c. 第三方服务端返回Signature，以及ExpireTime和Nonce值。
- d. 第三方客户端调用客户端SDK的Login接口，参数为用户ID、Signature、ExpireTime和Nonce。
- e. 客户端SDK向华为云会议服务端发起鉴权。
- f. 鉴权通过后SDK获取接入Token。该Token对第三方客户端不可见，由SDK负责维护和刷新。

2. 第三方服务App ID鉴权原理

图 3-2 第三方服务端 App ID 鉴权流程



鉴权前提：

- a. 开发者在[华为云会议控制台](#)上为自己的应用申请App ID，并获取App ID和App Key。
- b. 开发者在自己的服务端集成Signature生成算法，请参考“[第三方服务集成Signature生成算法](#)”。

鉴权过程：

- a. 第三方服务端根据调用的业务接口所需的User ID（第三方User ID），加上App ID、App Key、ExpireTime和Nonce，生成鉴权用的签名材料Signature。
- b. 第三方服务端调用华为云会议服务端的App ID鉴权接口（REST接口）。
- c. 华为云会议服务端鉴权通过后，返回接入Token。
- d. 第三方服务端使用接入Token调用其他业务接口。

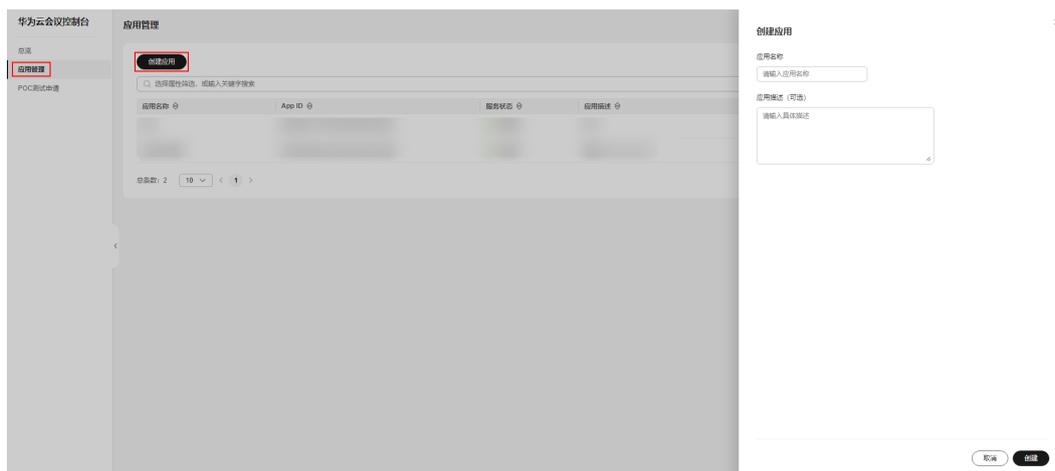
📖 说明

1. 接入Token有效期是12~24小时。
2. 生成Signature和调用appAuth接口时可以不带User ID，如果不带User ID默认是创建企业时的创建者（企业Owner）。
3. User ID需要第三方应用保证在同一个企业内是唯一的。如果第三方开发者是SP，同一个应用在不同企业使用，接口中除了带User ID外，还需要带Corp ID（企业ID）。
4. 如果需要使用华为云会议通讯录的情况下，在登录鉴权过程中还可以携带邮箱、姓名、电话号码等信息，这些信息将被写入华为云会议的通讯录中。

App ID 的申请

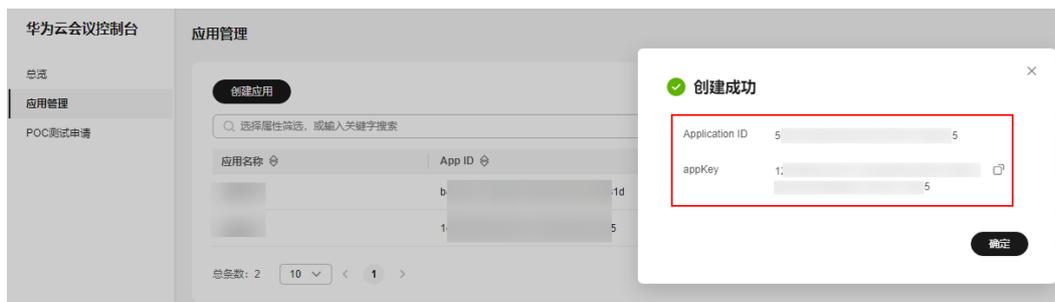
- 步骤1** 使用华为云账号，登录[华为云会议控制台](#)。前提是该华为云账号已经购买了华为云会议服务或者已经绑定了华为云会议企业管理员账号，请参考“[开发流程](#)”章节中的“[开发前准备](#)”。
- 步骤2** 单击“应用管理”，再单击“创建应用”。在弹出的创建应用的输入框中输入第三方应用的名称及描述。

图 3-3 创建应用



- 步骤3** 创建成功
- 单击“创建”后，会生成App ID和App Key。

图 3-4 创建成功



----结束

📖 说明

1. App Key是生成鉴权签名材料Signature的密钥，需要在第三方服务端妥善保存，不要泄露出去，否则会出现被盗用会议资源的风险。
2. App Key只能保存在第三方服务端，不能保存在第三方终端中，否则很容易被反编译获取。
3. App Key需要妥善保存，遗失后只能重置，不能找回。重新生成App Key原来的App Key将会一个月后失效。

第三方服务集成 Signature 生成算法

App只在单个企业内使用的情况下，生成鉴权签名材料Signature的算法如下：

$Signature = HexEncode(HMAC-SHA256((App\ ID + ":" + User\ ID + ":" + ExpireTime + ":" + Nonce), appKey))$

App是SP开发的，需要在多个企业内使用的情况下，生成鉴权签名材料Signature的算法如下：

- 作为企业普通用户，对自己的会议管理：
 $Signature = HexEncode(HMAC-SHA256((App\ ID + ":" + CorpID + ":" + User\ ID + ":" + ExpireTime + ":" + Nonce), appKey))$
- 作为企业管理员，对企业资源进行管理：
 $Signature = HexEncode(HMAC-SHA256((App\ ID + ":" + CorpID + ":" + ExpireTime + ":" + Nonce), appKey))$

📖 说明

作为企业管理员时也可以带User ID，这个User ID必须具有管理员权限。

- 作为SP管理员，对SP资源进行管理（如创建企业或者分配企业资源等）：
 $Signature = HexEncode(HMAC-SHA256((App\ ID + ":" + ExpireTime + ":" + Nonce), appKey))$

算法说明：

1. HMAC-SHA256的输入数据是App ID、CorpID（可选）、User ID（可选）、ExpireTime、Nonce值，中间用“:”分隔。如
d5e17*****489e:alice@ent01:1604020600:EycLQs*****n
INuU1EBpQ

📖 说明

SP模式下即使User ID或者Corp ID和User ID不填的情况下，对应的“:”不能省略。

2. HMAC-SHA256的密钥是App Key。如tZAe*****q32T
3. HMAC-SHA256生成的二进制数需要转换为十六进制字符串（HexEncode），上述的数据和密钥最终生成的Signature如下：
2a8c780c*****5a6c44f3c1b3c2455d
4. ExpireTime：鉴权签名材料Signature的过期时间戳。单位是秒。比如当前的系统的时间戳是1604020000，Signature有效期设置成10分钟，则ExpireTime = 1604020000+10*60 = 1604020600
5. Nonce：随机字符串，每次计算鉴权签名材料Signature时都必须不同。字符串长度为32~64字节

📖 说明

1. 由于鉴权签名材料Signature是有有效期的，所以第三方服务端的系统时间和华为云会议服务端的系统时间误差不能太大（小于ExpireTime中的有效期，如上例中的10分钟）。华为云会议服务端时间已经跟标准NTP时间同步。
2. ExpireTime为0，表示永远不过期。为了防止重放攻击，不建议将ExpireTime设置为0。

Java版本的Signature生成算法源代码

```
import java.io.UnsupportedEncodingException;
import java.security.InvalidKeyException;
import java.security.NoSuchAlgorithmException;
import javax.crypto.Mac;
import javax.crypto.spec.SecretKeySpec;

public class HmacSHA256 {

    //十六进制字符集
    private final static char[] DIGEST_ARRAYS = {
        '0', '1', '2', '3', '4', '5', '6', '7', '8', '9', 'a', 'b', 'c', 'd', 'e', 'f'
    };

    /**
     * 功能：Signature生成算法
     * 输入参数：
     * 1.data: HMAC-SHA256的输入数据
     * 2.key: App Key
     * 输出参数：十六进制字符串编码的HMAC-SHA256值
     */
    public static String encode(String data, String key) {
        byte[] hashByte;
        try {
            Mac sha256HMAC = Mac.getInstance("HmacSHA256");
            SecretKeySpec secretKey = new SecretKeySpec(key.getBytes("UTF-8"), "HmacSHA256");
            sha256HMAC.init(secretKey);

            hashByte = sha256HMAC.doFinal(data.getBytes("UTF-8"));
        } catch (NoSuchAlgorithmException | UnsupportedEncodingException | InvalidKeyException e) {
            return null;
        }

        return bytesToHex(hashByte);
    }

    /**
     * 功能：byte类型数组转换成十六进制字符串
     * 输入参数：
     * 1.bytes:被转换的字节数组
     * 输出参数：十六进制字符串
     */
    private static String bytesToHex(byte[] bytes) {
        StringBuffer hexStr = new StringBuffer();
        for (int i = 0; i < bytes.length; i++) {
            hexStr.append(DIGEST_ARRAYS[bytes[i] >>> 4 & 0X0F]);
            hexStr.append(DIGEST_ARRAYS[bytes[i] & 0X0F]);
        }

        return hexStr.toString();
    }
}
```

Python版本的Signature生成算法源代码

```
import hmac
from hashlib import sha256

class HmacSHA256:
    def __init__(self, sig_data, sig_key):
        self.data = sig_data
        self.key = sig_key

    def encode(self):
        try:
            sig_data = self.data.encode('utf-8')
            secret_key = self.key.encode('utf-8')
            signature = hmac.new(secret_key, sig_data, digestmod=sha256).hexdigest()
        except Exception as e:
```

```
print (e)
raise e
return signature
```

C++版本的Signature生成算法源代码

```
#include <openssl/hmac.h>
#include <string.h>
#include <iostream>
using namespace std;

const int HMAC_ENCODE_SUCCESS = 0;
const int HMAC_ENCODE_FAIL = -1;
const int HMAC_SHA256_STR_LEN = 65;

/**
 * 功能: 字节流转换成十六进制字符串
 * 输入参数:
 *     1.input: 字节流指针
 *     2.len: 字节流字节数
 * 输出参数:
 *     3.output: 输出缓冲区
 *     6.output_length: Signature字符串长度
 * 返回值:
 *     0: 失败
 *     大于0的整数: 转换后的字符串长度
 */
int Byte2HexStr(char * output, unsigned char * input, unsigned int len)
{
    if ((NULL == output) || (NULL == input))
    {
        return 0;
    }

    unsigned int i = 0;
    for (i = 0; i < len; i++)
    {
        sprintf_s(output + 2*i, HMAC_SHA256_STR_LEN-2*i, "%x%x", (input[i] >> 4) & 0x0F, input[i]&0x0F);
    }

    *(output + 2*i) = '\0';

    return 2*i;
}

/**
 * 功能: Signature生成算法
 * 输入参数:
 *     1.key: App Key
 *     2.key_length: App Key的字符串长度
 *     3.input: HMAC-SHA256的输入数据
 *     4.input_length: HMAC-SHA256的输入数据的字符串长度
 * 输出参数:
 *     5.output: Signature输出缓冲区
 *     6.output_length: Signature字符串长度
 * 返回值:
 *     0: 成功
 *     -1: 失败
 */
int HmacEncode(const char * key, unsigned int key_length,
               const char * input, unsigned int input_length,
               char output[HMAC_SHA256_STR_LEN], unsigned int &output_length) {

    //计算HMAC_SHA256的字节流
    const EVP_MD * engine = EVP_sha256();

    unsigned char * byte_output = (unsigned char*)malloc(EVP_MAX_MD_SIZE);
    if (NULL == byte_output)
```

```
{
    output_length = 0;
    return HMAC_ENCODE_FAIL;
}
unsigned int byte_output_length = 0;

HMAC_CTX *ctx;
ctx = HMAC_CTX_new();

HMAC_Init(ctx, key, strlen(key), engine);
HMAC_Update(ctx, (unsigned char*)input, strlen(input));
HMAC_Final(ctx, byte_output, &byte_output_length);

HMAC_CTX_free(ctx);

//HMAC_SHA256的字节流转换成十六进制字符串
int ret = Byte2HexStr(output, byte_output, byte_output_length);
free(byte_output);

if (0 == ret)
{
    output_length = 0;
    return HMAC_ENCODE_FAIL;
}
else
{
    output_length = ret;
    return HMAC_ENCODE_SUCCESS;
}
}
```

📖 说明

C++的Signature生成算法基于openssl 1.1.0以上版本实现。openssl库需要开发者自己编译和安装，请参考[openssl官网](#)。

JavaScript版本的Signature生成算法源代码

```
<script src="https://cdnjs.cloudflare.com/ajax/libs/crypto-js/3.1.9-1/crypto-js.js"> </script>
<script>
    /**
     * 1、重要提醒：生成Signature的算法不要放在前端代码中，否则会导致App Key泄露，从而导致企业资源被
     盗
     * 2、本示例代码仅供调试时使用
     */
    function genSignature(data,appKey){
        var sign = CryptoJS.HmacSHA256(data,appKey).toString();
        return sign
    }
</script>
```

集成 UI SDK 的第三方客户端 App ID 鉴权

第三方客户端初始化SDK时，需要传入事先申请好的App ID。

- Android SDK的初始化接口，请参考《客户端SDK参考》中“Android SDK”的“[初始化](#)”章节。
- iOS SDK的初始化接口，请参考《客户端SDK参考》中“iOS SDK”的“[初始化](#)”章节。
- Windows SDK的初始化接口，请参考《客户端SDK参考》中“Windows SDK”的“[初始化](#)”章节。

- Mac SDK的初始化接口，请参考《客户端SDK参考》中“Mac SDK”的“[初始化](#)”章节。

第三方客户端登录时，从第三方服务端获取Signature、ExpireTime和Nonce后，调用终端SDK的Login接口完成鉴权。

- Android SDK的Login接口，请参考《客户端SDK参考》中“Android SDK”的“[登录](#)”章节。
- iOS SDK的Login接口，请参考《客户端SDK参考》中“iOS SDK”的“[登录](#)”章节。
- Windows SDK的Login接口，请参考《客户端SDK参考》中“Windows SDK”的“[登录](#)”章节。
- Mac SDK的Login接口，请参考《客户端SDK参考》中“Mac SDK”的“[登录](#)”章节。

第三方服务端 App ID 鉴权

第三方服务端通过App ID鉴权接口，完成App ID鉴权，并获取接入Token。请参考《服务端API参考》中的“[执行App ID鉴权](#)”章节。

4 开发流程

图 4-1 开发流程



开发前准备

1. 注册华为账号并开通华为云
登录[华为云官网](#)，单击“注册”，按照界面提示完成华为云的注册。成功注册后请尽快完成“企业实名认证”，认证过程请参考[企业账号如何完成实名认证](#)。
2. 开通华为云会议企业管理员账号
 - **正式商用**
订购华为云会议服务。如何订购，请参考[如何购买会议套餐](#)。
 - **免费试用**
 - a. 用步骤1中申请的华为云账号登录华为云官网后，进入[华为云会议官网](#)页面。单击“控制台”。

图 4-2 进入控制台



b. 单击“开通云会议服务”。

图 4-3 开通云会议服务



c. 单击“POC测试申请”，并填写企业名称、联系方式和使用场景。

图 4-4 申请 POC 测试资源

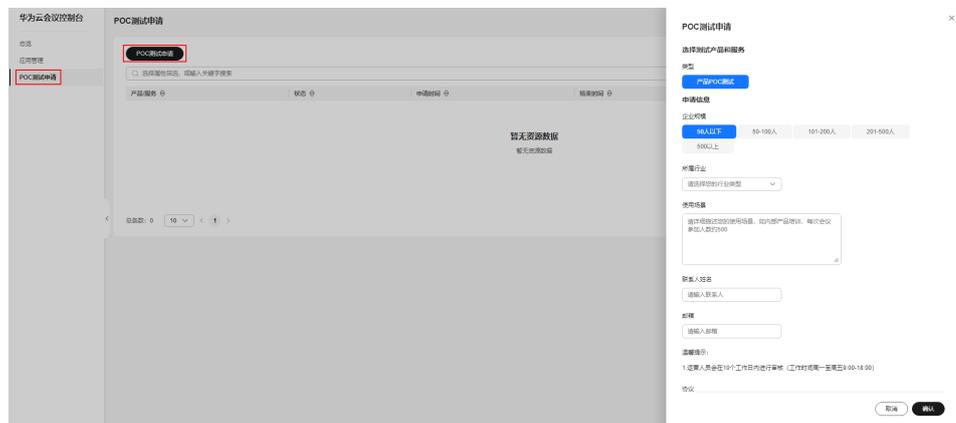


图 4-5 等待审批



说明

- 二次开发的资源审批请联系华为销售人员。

图 4-6 审批通过



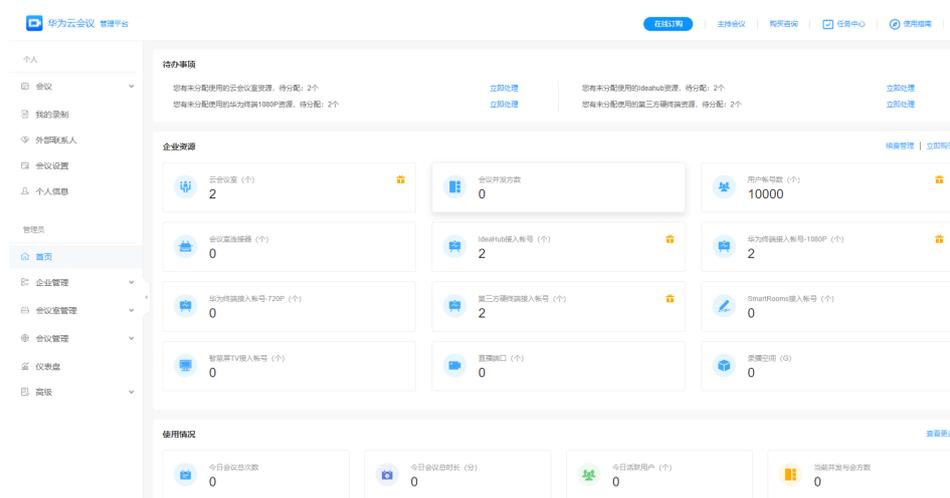
3. 添加其他账号（可选）

在华为云会议管理后台，单击“前往管理平台”进入华为云会议管理平台。

图 4-7 前往管理平台



图 4-8 华为云会议管理平台



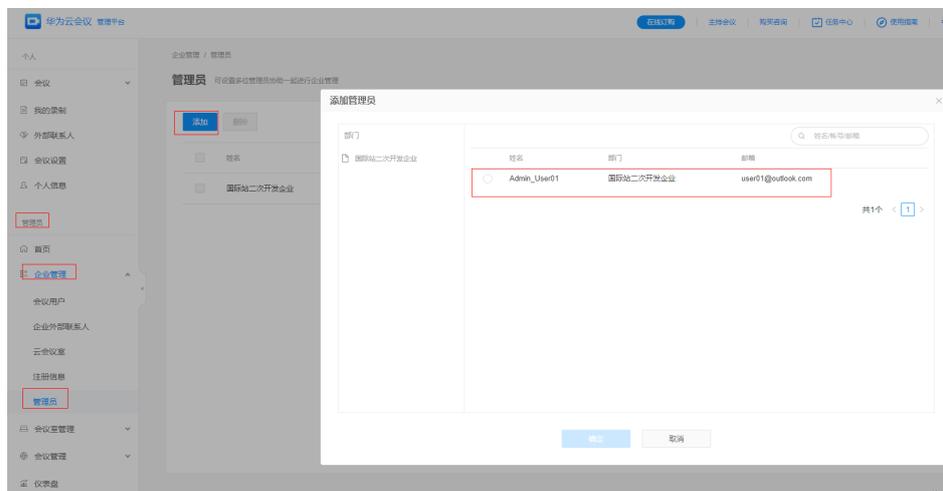
登录后，在左侧菜单栏中选择“管理员>企业管理 > 会议用户”，添加会议用户。

图 4-9 添加会议用户



普通的会议用户可以通过“管理员>企业管理 > 管理员”，添加为该企业的企业管理员。

图 4-10 添加管理员



4. 申请App ID
App ID鉴权的介绍和申请请参考“[App ID鉴权介绍](#)”。

集成服务端 REST API

1. 下载服务端集成开发工具以及二次开发文档。
在集成开发服务端API的过程中，除本开发指南外，开发者还需要参考《[服务端API参考](#)》。
2. 准备开发环境。
订购华为云会议服务，基于业务账号调用服务端API，接入地址是“<https://api-intl.meeting.huaweicloud.com>”。
3. 集成登录鉴权。
第三方系统在直接调用云会议功能前，需要先调用鉴权接口，完成第三方应用账号向会议服务端的鉴权。详细开发方法参见《[服务端API参考](#)》中的[登录鉴权](#)章节。

图 4-11 服务端鉴权集成



4. 调用REST接口集成开发。
在完成鉴权调用后，开发者可以在第三方应用系统上集成华为云会议的服务端功能接口。

详细开发方法参见《[服务端API参考](#)》。

5. 功能调试。

开发完成后，可以接入华为云会议调试环境（<https://api-intl.meeting.huaweicloud.com>）进行功能调试。

华为云会议提供的每个服务端REST API都可以在华为云的[API Explorer](#)上调试。

集成客户端 UI SDK

1. 下载客户端SDK开发包以及二次开发文档。

在集成开发客户端SDK的过程中，除本开发指南外，开发者还需要查看本《[客户端SDK参考](#)》。

请参考《[客户端SDK参考](#)》中的“[资源下载](#)”获取SDK开发包，Demo源代码在SDK包中。

2. 试用Demo。

华为云会议为每个平台SDK提供了Demo安装包，开发者可以下载Demo，快速体验功能的运行和调试。

请参考《[客户端SDK参考](#)》中每个平台SDK下的“[Demo安装包下载](#)”试用Demo。

3. 准备开发环境。

- 开发环境准备

- 不同平台的开发工具及环境要求。详见《[客户端SDK参考](#)》中对应平台下的“快速入门”。

- 调测环境准备

订购华为云会议服务后或者申请免费试用账号后，调用客户端SDK。

4. 调用SDK集成开发。

开发者可以在自己对应要开发的平台上集成华为云会议对应平台的SDK接口。

详细开发方法参见《[客户端SDK参考](#)》。

5. 功能调试。

开发完成后，可以接入华为云会议调试环境进行功能调试。

商用发布

开发者完成应用开发和调测后将应用发布给最终用户使用。

5 最佳实践

开放性应用实践概览

华为云会议提供了服务端API开放和客户端SDK开放，开发者基于自己的应用场景可以灵活地集成华为云会议的开放性接口。本文介绍了几个基于华为云会议集成的最佳实践，帮助开发者了解几种典型场景下的集成方案，以管窥豹，快速掌握基于华为云会议的集成开发能力。

图 5-1 开放性应用实践概览

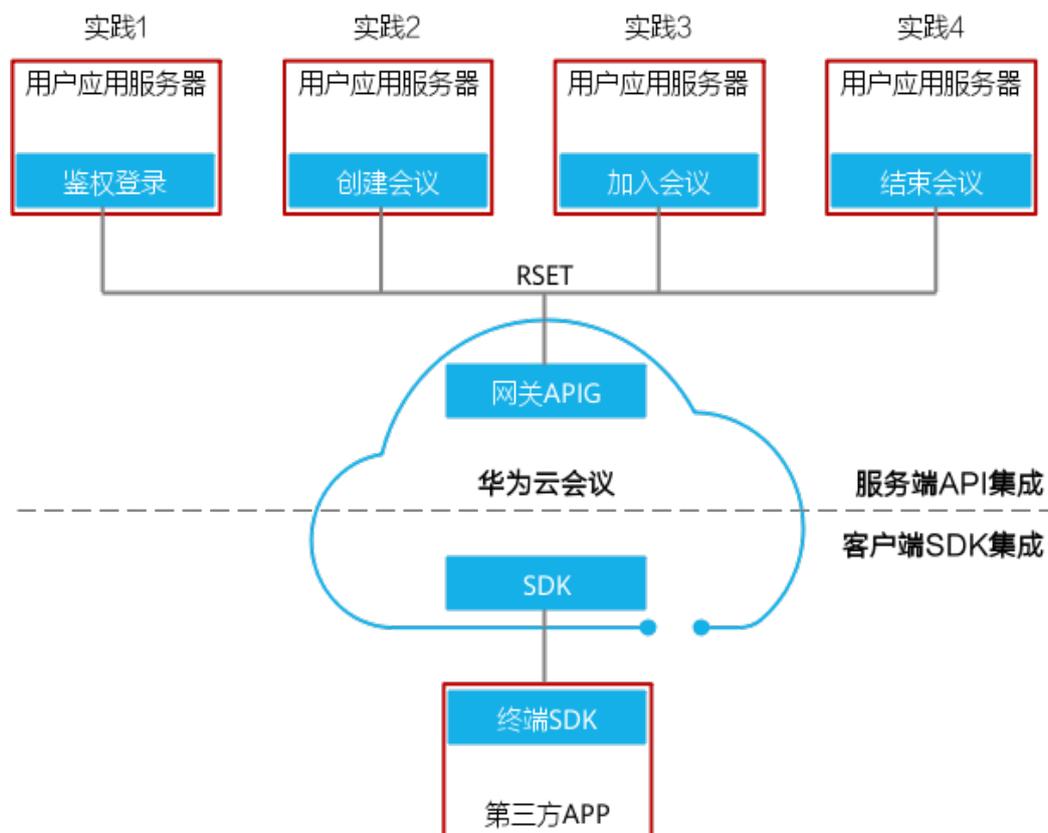


表 5-1 服务端 API 集成实践

| 实践 | 应用场景 |
|-----------|---|
| 实践1: 鉴权登录 | 服务端API调用前需要鉴权登录后获取接入Token, 开放者通过App ID鉴权接口完成鉴权登录, 并获取接入Token。 |
| 实践2: 创建会议 | 服务端创建会议: 用户已有的管理系统可集成华为会议服务的会议管理接口, 可以调用服务端API创建会议。 客户端创建会议: 用户可以在客户端上直接创建立即开始的会议或者预约会议。 |
| 实践3: 加入会议 | 服务端拉起与会者: 调用服务端API, 添加与会者。与会者收到会议来电并加入会议。 客户端主动入会: 用户可以, 通过输入会议ID/密码入会; 单击邮件链接入会; 单击Web链接入会; 在微信/微信小程序中单击链接入会。 |
| 实践4: 离开会议 | 服务端结束会议: 调用服务端API, 结束会议。 客户端离开/结束会议: 用户在会中界面上结束会议和离开会议。 |

最佳实践 1: 鉴权登录

开发者集成华为云会议服务端API时, 推荐使用App ID鉴权方式, 鉴权通过后可以获取接入Token。第一次鉴权通过后, 华为云会议后台会自动为第三方的用户应用分配一个华为云会议账号。

图 5-2 APP ID 鉴权



登录成功后, 华为云会议会自动给第三方应用分配华为云会议账号。这个绑定关系由华为云会议维护, 第三方开发者不用关注华为云会议账号。

表 5-2 第三方账号和华为云会议账号的映射关系

| 第三方账号 | 华为云会议账号 |
|-------------|---------------------------------------|
| 15012345678 | Auto-94b91e94ce3f4ddab9ecfe7625418e60 |
| 18012345678 | Auto-7700ebc37cf580a680cdfde4b34b41d0 |

最佳实践 2：创建会议

- 用户的应用服务器通过调用服务器API接口，实现预定会议或创建即时会议。
- 用户的APP通过调用客户端SDK接口，实现预定会议或创建即时会议。

图 5-3 服务端创建会议集成方案

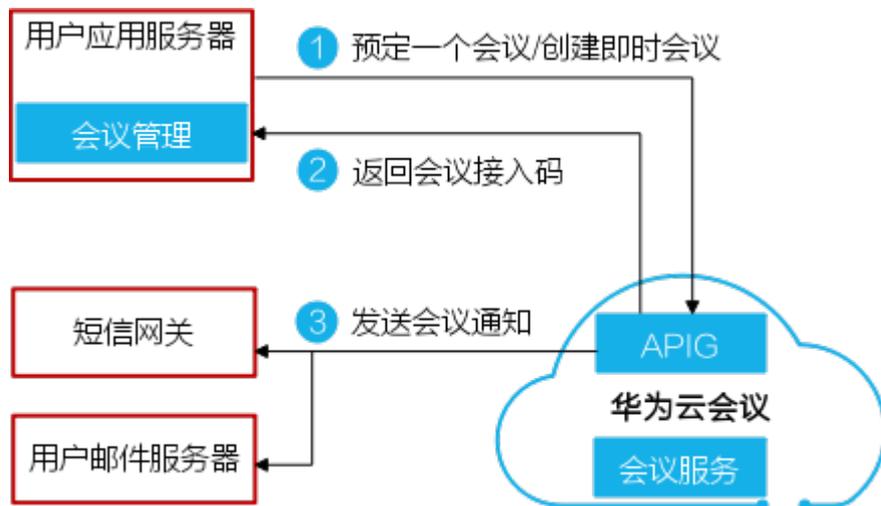
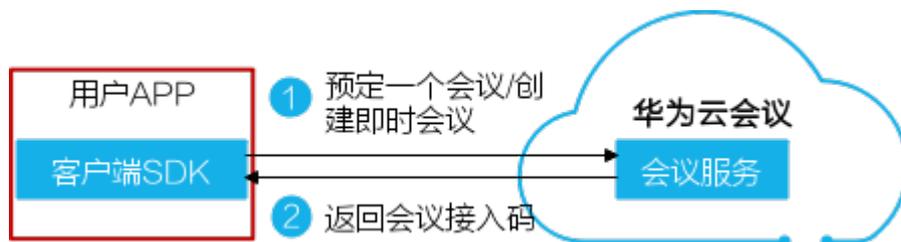


图 5-4 客户端创建会议集成方案

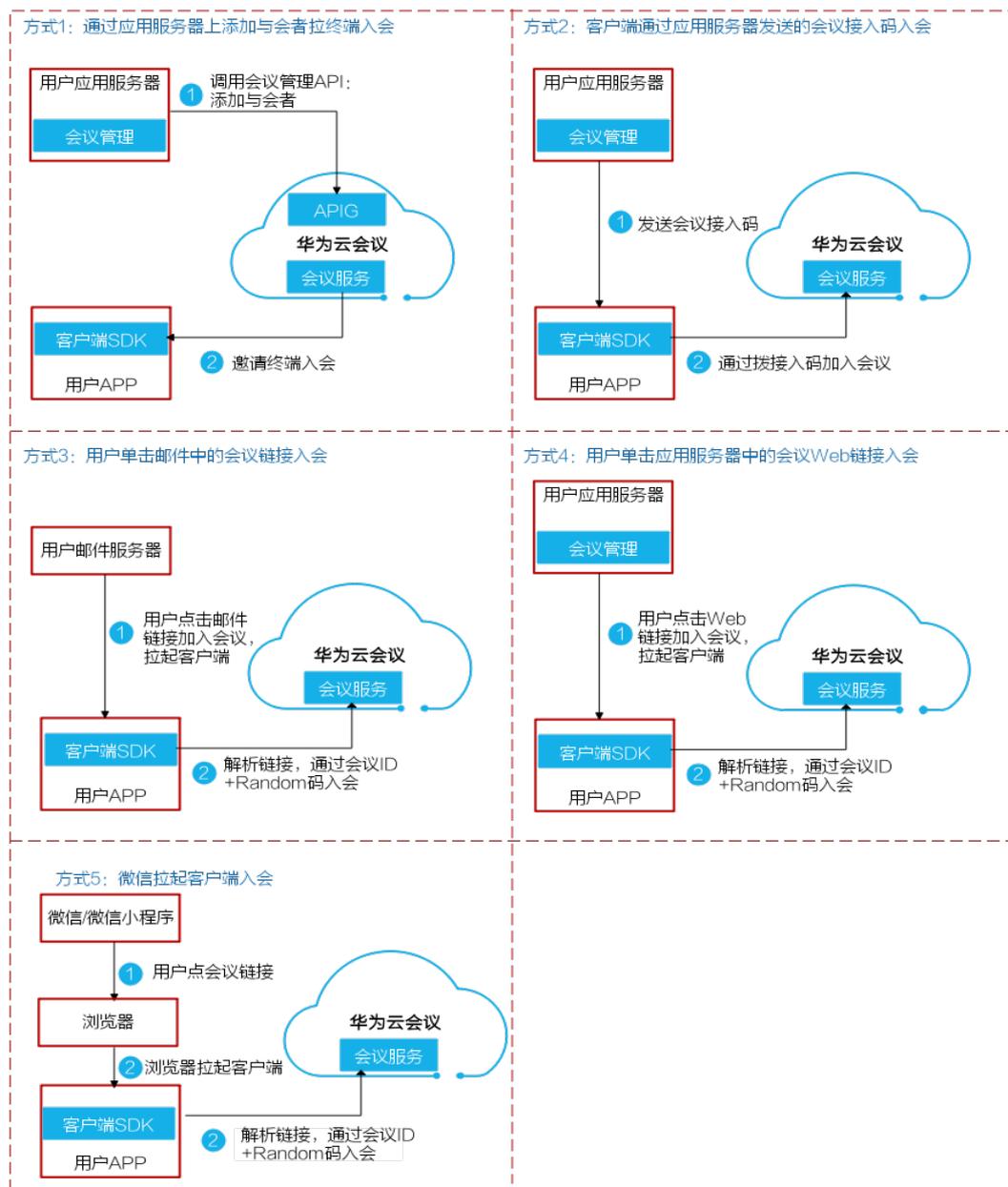


最佳实践 3：加入会议

华为云会议提供多种入会方式，用户的应用服务器或者APP通过集成华为云会议的入会接口，可以通过多种方式加入会议。

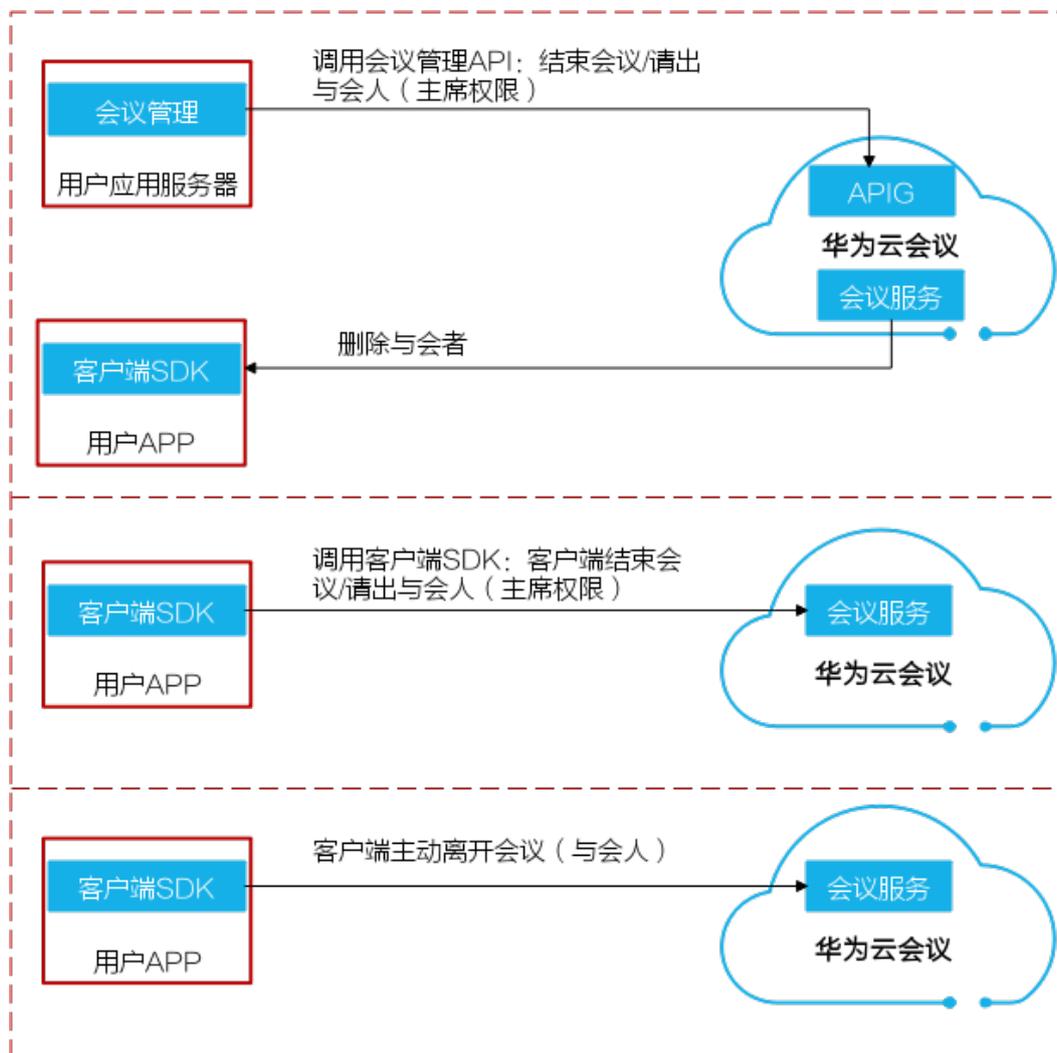
- 方式1：终端被邀入会
- 方式2：终端通过会议ID和密码主动入会
- 方式3：邮件链接拉起终端入会
- 方式4：Web链接拉起终端入会
- 方式5：微信/微信小程序拉起终端入会

图 5-5 加入会议集成方案



最佳实践 4：离开/结束会议

图 5-6 离开会议集成方案



6 注意事项

- 传输安全
为了传输安全，REST API必须通过HTTPS传输，不支持HTTP传输。
- HTTP编码
请求及返回结果都使用UTF-8字符集进行编码。

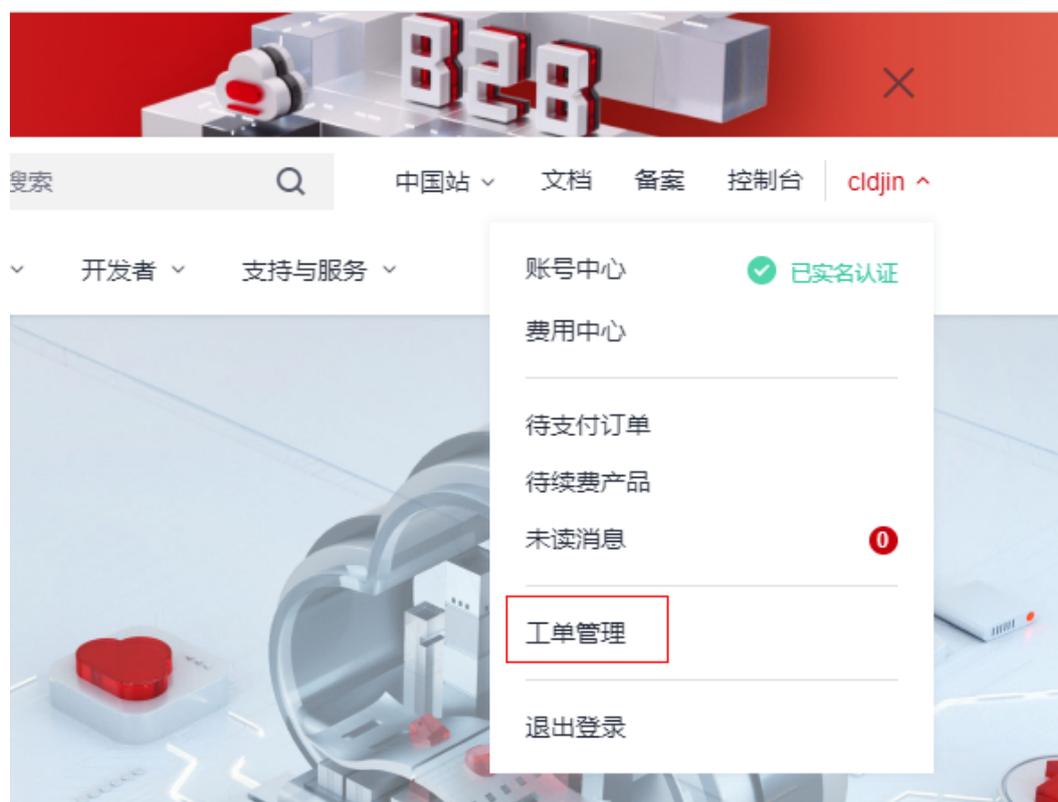
7 提交问题工单

在集成云会议产品的过程中，如有任何问题，推荐您使用华为云工单系统提交问题，技术支持工程师收到工单后会尽快处理您的问题。

步骤1 登录[华为云官网](#)。

步骤2 选择“账号名 > 工单管理”。

图 7-1 选择工单管理



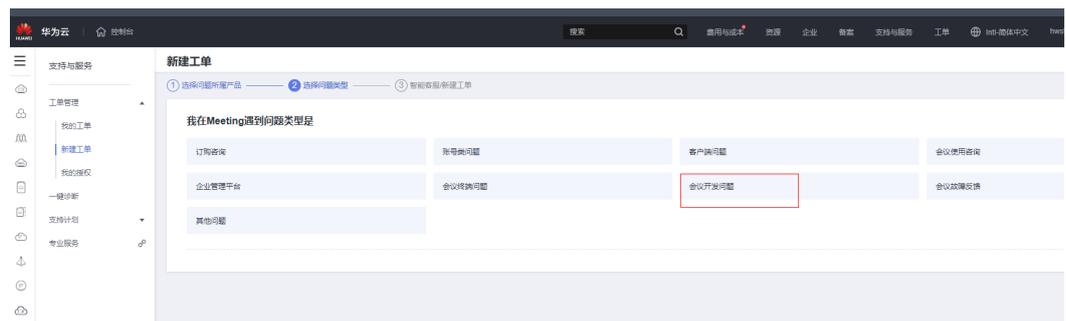
步骤3 搜索“会议”。

图 7-2 搜索“会议”



步骤4 选择问题类型为“会议云服务-开放集成类”，并单击“新建工单”。

图 7-3 选择问题类型



步骤5 根据界面提示，完成问题和信息填写，然后提交工单。

请尽量详细描述您的问题现象、问题所属类型（服务端集成或客户端集成），并上传问题相关日志，以便支持人员能尽快定位问题。

📖 说明

问题相关日志请只保留当前问题发生时的日志，删除以前的日志。建议这样获取日志：清空以前的日志文件-->复现问题-->重新获取日志文件。

图 7-4 填写工单信息

新建工单

选择问题所属产品 选择问题类型

会议 > 会议云服务-开放集成类 > 新建工单

* 案例严重性

您的案例严重性选项根据您所购买的支持计划等级权益而定。 [查看支持计划](#)

* 区域

* 问题描述

0/1200

请勿在问题描述中输入您的用户名、密码、银行账户等机密信息。

机密信息 暂不输入 立即输入

联系方式 工单留言 手机 邮箱

* 手机号码 -

手机号用来接收工单进展提醒；必要时，华为云工程师会主动与您电话联系。

联系时间 任何时间 指定时间

* 邮箱

邮箱仅用来接受工单进展提醒。

上传附件

只能上传3个大小不超过4M的附件，支持格式: JPG, JPEG, BMP, PNG, GIF, TXT, DOC, DOCX, RAR, ZIP, PDF, XLS,)

我已经阅读并同意 [《用户授权书》](#) 和 [《隐私政策声明》](#)

咨询·反馈

----结束