

简介

说明

云屋会议API是为使用云屋音视频服务的合作方开发者接入并访问云屋会议资源提供的一套工具。合作伙伴可以通过云屋会议API进行二次开发，例如查询会议、创建会议等。

使用API

访问[云屋视频会议官网](#)，点击[登录](#)。

访问云屋会议管理后台，使用开通的账号登录后台。

登录成功后，在**企业中心** -> **WEB-API**，可以查看API调用方的身份凭证。身份凭证包括 secretId 和 secretKey。

- secretId 是企业的唯一标识，用于表明API调用者的身份。
- secretKey 是密钥，用于生成签名。

鉴权方式

公共参数

| 参数名称 | 类型 | 是否必传 | 说明 |
|----------------|--------|------|--|
| X-CR-SecretId | String | 是 | 每个企业的唯一标识 secretId |
| X-CR-Timestamp | Long | 是 | 当前请求的时间戳，单位为秒 如果与服务器时间相差超过5分钟，会引起签名过期错误。 |
| X-CR-Signature | String | 是 | 签名。加密算法: SHA256，值为 sha256(secretId + '&' + Timestamp + '&' + secretKey) |
| X-CR-RequestId | String | 是 | 每次请求API的ID，可用UUID |

说明：

- 以上参数统一放到 HTTP Header 请求头部中
- secretId 用于唯一标识，指明是哪个企业；secretKey 密钥，用于生成签名。
- 每次请求都要生成一次签名

使用示例

从会议管理后台获取到如下内容：

| 选项 | 值 |
|-----------|----------------------------------|
| secretId | 6mioUaDlFdf3Bagt3s |
| secretKey | t02npiEZn82WqRl26Nfb0stD89bZkwTT |

程序生成当前时间戳 timestamp = 1622098479,

计算签名 signature = sha256(secretId + '&' + Timestamp + '&' + secretKey)
= edda74671f313bc7c081bf6e11c8da9ebe73224e4aadba96f6ae2c6ba4550f25。

生成签名的代码示例如下：

```
package com.cloudroom.api.core;

import com.google.common.hash.Hashing;

import java.nio.charset.StandardCharsets;
import java.util.Calendar;

public class SignTest {
    public static void main(String[] args) {
        String secretId = "6mioUaDlFdf3Bagt3s";
        String secretKey = "t02npiEZn82WqRl26Nfb0stD89bZkwTT";
        long timestamp = Calendar.getInstance().getTimeInMillis() / 1000;
        System.out.println(timestamp);

        String signature = Hashing.sha256()
            .hashString(secretId + '&' + timestamp + '&' + secretKey, StandardCharsets.UTF_8)
```

```
        .toString();  
        System.out.println(signature);  
    }  
}
```

调用创建会议接口，在HTTP Header 请求头部中填入鉴权信息，如图所示：

在body中填入业务参数，如下图所示：

调用成功后，会获取到刚创建的会议号，用于后续业务处理。

常见问题

签名过期

如果接口返回如图结果

说明HTTP Header 请求头部参数 X-CR-Timestamp 的值与服务器时间相差超过5分钟。

需要重新获取当前时间戳，重新生成签名。

修改参数 X-CR-Timestamp 和 X-CR-Signature 的值。

签名无效

如果接口返回如图结果

说明HTTP Header 请求头部参数 X-CR-Signature 的值不对

请按照上文的签名生成规则，生成正确的值。

其他鉴权的错误信息，请参考[鉴权错误码](#)

使用流程

调用[创建会议](#)接口，获得会议号。根据是否使用云屋入会页面，有以下两种情况：

- 调用[查询会议](#)接口，获得会议链接。在浏览器输入会议链接，打开入会页面，会调起会议客户端，从而进入会议。
- 调用[加入会议](#)接口，获得CRMT串。在浏览器输入CRMT串，会调起会议客户端，从而进入会议。

创建会议

- url

https://www.cloudroom.com/CLOUDROOM-SERVER/v2/conference

- 请求方式 POST
- 请求参数

| 参数名 | 必填 | 类型 | 描述 |
|----------------|----|-------------|-------------------------|
| subject | 是 | String | 会议主题 |
| hostId | 否 | Long | 主持人id（默认使用企业创建者的账号为主持人） |
| inviteIds | 否 | Long[] | 参会人id |
| assistantInfos | 否 | Long[] | 联席主持人id |
| password | 否 | String | 会议密码（4-6位数字） |
| emailNotice | 否 | Integer | 是否邮件通知（0：否；1：是，默认值为0） |
| smsNotice | 否 | Integer | 是否短信通知（0：否；1：是，默认值为0） |
| orderConfig | 否 | OrderConfig | 预约配置 |
| attributes | 否 | Attributes | 会议属性 |
| permission | 否 | Permission | 参会者权限 |
| votelds | 否 | String[] | 投票id |

- 请求参数示例

```
{ "subject": "创建会议", "order": 1, "orderConfig": { "periodType": 0 }, "attributes": { "waterMark": 1 } }
```

- 返回参数

| 参数名 | 类型 | 描述 |
|--------|------|-----|
| confId | Long | 会议号 |

- 返回参数示例

```
{ "confId": 12345678 }
```

修改会议

- url

https://www.cloudroom.com/CLOUDROOM-SERVER/v2/conference/{confId}

- 请求方式 PUT
- 请求参数

| 参数名 | 必填 | 类型 | 描述 |
|----------------|----|-------------|-------------------------|
| subject | 否 | String | 会议主题 |
| hostId | 否 | Long | 主持人id（默认使用企业创建者的账号为主持人） |
| inviteeIds | 否 | Long[] | 参会人id |
| assistantInfos | 否 | Long[] | 联席主持人id |
| password | 否 | String | 会议密码（4-6位数字） |
| orderConfig | 否 | OrderConfig | 预约配置 |
| attributes | 否 | Attributes | 会议属性 |
| permission | 否 | Permission | 参会者权限 |

- 请求示例

https://www.cloudroom.com/CLOUDROOM-SERVER/v2/conference/12345678 json

- 请求参数示例

{"subject": "修改会议", "orderConfig": {"periodType": 0}, "attributes": {"waterMark": 1}} json

- 返回参数
- 返回参数示例

{ } json

查询会议

- url

https://www.cloudroom.com/CLOUDROOM-SERVER/v2/conference/{confId}

- 请求方式 GET
- 请求参数示例

```
https://www.cloudroom.com/CLOUDROOM-SERVER/v2/conference/12345678
```

json

- 返回参数

| 参数名 | 类型 | 描述 |
|------------|--------|------|
| confDetail | Object | 会议详情 |

会议详情

| 参数名 | 类型 | 描述 |
|----------------|------------------|-------------------------|
| confId | Long | 会议Id |
| subject | String | 会议主题 |
| url | String | 主持人入会链接 |
| publicUrl | String | 公共入会链接 |
| password | String | 会议密码 |
| hostId | Long | 主持人id |
| hostName | String | 主持人 |
| inviteeInfos | InviteeInfo[] | 参会人信息 |
| assistantInfos | AssistantInfos[] | 联席主持人信息 |
| status | Integer | 会议状态（0：未开始；1：进行中；2：已结束） |
| pstnNum | String | 电话参会号码 |
| pstnPwd | String | 电话参会密码 |
| emailNotice | Integer | 是否邮件通知（0：否；1：是，默认值为0） |
| smsNotice | Integer | 是否短信通知（0：否；1：是，默认值为0） |
| orderConfig | OrderConfig | 预约配置 |
| attributes | Attributes | 会议属性 |
| permission | Permission | 参会者权限 |

- 返回参数示例

```
{
  "confDetail": {
    "confId": 12345678,
    "subject": "会议主题",
    "url": "http://www.cloudroom.com/abcdef",
```

json

```
"publicUrl": "http://www.cloudroom.com/abc123",
"hostId": 1000141441,
"hostName": "云屋科技",
"status": 0,
"pstnPwd": "123456",
"pstnNum": "008612345678901",
"orderConfig": {
  "scheduleTime": "2021-04-14 15:07:31",
  "scheduleEndTime": "2021-04-14 17:07:31",
  "periodType": 0,
  "periodEndTime": ""
},
"attributes": {
  "autoOpenCamera": 0,
  "autoOpenMicrophone": 0,
  "micUseLocal": 0,
  "cameraUseLocal": 0,
  "loginIvr": 0,
  "onlyInvitee": 0,
  "onlyComp": 0,
  "canJoinAdv": 1,
  "virtualMembers": 0,
  "waterMark": 0,
  "vwalldo": 0
},
"permission": {
  "createWB": 0,
  "saveWB": 0,
  "publicIM": 0,
  "privateIM": 0,
  "recordConf": 0,
  "uploadDoc": 0,
  "openMicrophone": 0,
  "openCamera": 0,
  "shareOrUpload": 0,
  "srvRecord": 0,
  "pstnCall": 0,
  "allowModifyName": 0,
  "imForbidPic": 0,
  "imForbidUrl": 0,
  "allowShare": 0
}
}
}
```


加入会议

- url

https://www.cloudroom.com/CLOUDROOM-SERVER/v2/conference/{confId}/join?nickName={nickName}&userType={userType}&password={password}
- 请求方式 GET
- 请求参数

| 参数名 | 必填 | 类型 | 描述 |
|----------|----|---------|---------------------------|
| nickName | 是 | String | 入会昵称 |
| userType | 否 | Integer | 参会者类型（1：主持人；4：普通参会者，默认是4） |
| password | 否 | String | 会议密码 |

- 请求示例

https://www.cloudroom.com/CLOUDROOM-SERVER/v2/conference/12345678/join?nickName=roomjson

- 返回参数

| 参数名 | 类型 | 描述 |
|------|--------|----------|
| crmt | String | 入会的crmt串 |

- 返回参数示例

```
{ "crmt": "CRMT:LU1FRVRUWVBFPUIFRVRNR1IKLU1HUKFDTlQ9MTU40TA5MDgwODA7ZTEwYWRjMzk0OWJhNTlhYmJlNTZlMDU3ZjIwZjg4M2UKLU1H" }
```

json

CRMT串使用示例

浏览器调用

在浏览器输入CRMT串：

```
CRMT:LU1FRVRUWVBFPUIFRVRNR1IKLU1HUKFDTlQ9MTU40TA5MDgwODA7ZTEwYWRjMzk0OWJhNTlhYmJlNTZlMDU3ZjIwZjg4M2UKLU1H
```

浏览器会弹出相应提示框

点击打开，会调起本地安装的会议客户端，加入该会议

windows程序调用

例如，从接口获取到CRMT串如下：

```
CRMT:LU1FRVRUWVBFPUIFRVRNR1IKLU1HUKFDT1Q9MTU40TA5MDgwODA7ZTEwYWRjMzk0OWJhNT1hYmJlNTZlMDU3ZjIwZjg4M2UKLU1H
```

那么，使用如下代码，可以调用起会议客户端。

```
ShellExecuteA(NULL, "open", "CRMT:XXX", NULL, NULL, SW_SHOW);
```

示例如下：

```
ShellExecuteA(NULL, "open", "CRMT:LU1FRVRUWVBFPUIFRVRNR1IKLU1HUKFDT1Q9MTU40TA5MDgwODA7ZTEwYWRjMzk0OWJhNT1
```

具体CRMT串按照实际值进行修改。

结束会议

- url

```
https://www.cloudroom.com/CLOUDROOM-SERVER/v2/conference/{confId}/end
```

- 请求方式 POST

- 请求参数

- 请求示例

```
https://www.cloudroom.com/CLOUDROOM-SERVER/v2/conference/12345678/end
```

json

- 返回参数

- 返回参数示例

```
{}
```

json

删除会议

- url

```
https://www.cloudroom.com/CLOUDROOM-SERVER/v2/conference/{confId}
```

- 请求方式 DELETE
- 请求参数
- 请求示例

```
https://www.cloudroom.com/CLOUDROOM-SERVER/v2/conference/12345678
```

json

- 返回参数
- 返回参数示例

```
{}
```

json

查询会议列表

说明：

访问限制：300次/min
如果有更高的访问需求，请联系对接的商务接口人。

• url

https://www.cloudroom.com/CLOUDROOM-SERVER/v2/conference/list?searchUserId={searchUserId}&startTime={startTime}&endTime={endTime}¤t={current}&size={size}

• 请求方式 GET

• 请求参数

| 参数名 | 必填 | 类型 | 描述 |
|--------------|----|---------|-------------------------------|
| searchUserId | 否 | Long | 查询用户会议的用户id |
| startTime | 否 | Long | 开始时间（时间戳，单位：秒） |
| endTime | 否 | Long | 结束时间（时间戳，单位：秒，时间间隔最大一年） |
| current | 否 | Integer | 当前页码（默认为1，超出范围使用默认值） |
| size | 否 | Integer | 每页数量（默认为20，最大值为100，超出范围使用默认值） |

• 请求示例

https://www.cloudroom.com/CLOUDROOM-SERVER/v2/conference/list?current=1&size=20

json

• 返回参数

| 参数名 | 类型 | 描述 |
|-----------|----------|---------------------|
| total | Integer | 总数 |
| remaining | Integer | 是否还有未拉取的数据（0：否；1：是） |
| records | Object[] | 会议列表 |

◦ 会议列表详情

| 参数名 | 类型 | 描述 |
|-----------------|---------|---|
| confId | Long | 会议号 |
| subject | String | 会议主题 |
| confType | Integer | 会议类型（0：非预约会议；1：预约会议） |
| scheduleTime | Long | 预约开始时间（时间戳，单位：秒） |
| scheduleEndTime | Long | 预约结束时间（时间戳，单位：秒） |
| hostName | String | 主持人姓名 |
| members | Integer | 在线人数 |
| periodType | Integer | 周期类型（0：非周期会议；1：每天；2：每个工作日；3：每周；4：每两周；5：每月；6：每季度；7：每年） |

| 参数名 | 类型 | 描述 |
|----------|---------|-------------------|
| status | Integer | 会议状态（0：未开始；1：进行中） |
| password | String | 会议密码 |
| ownerId | Long | 拥有者id |

- 返回参数示例

```
{
  "total": 1,
  "records": [
    {
      "confId": 12345678,
      "subject": "会议主题",
      "confType": 1,
      "scheduleTime": 1616655277,
      "scheduleEndTime": 0,
      "startTime": 1616655277,
      "endTime": 0,
      "hostName": "云屋科技",
      "members": 0,
      "confCode": "abcdef",
      "periodType": 0
    }
  ]
}
```

json

查询历史会议列表

说明：

访问限制：300次/min

如果有更高的访问需求，请联系对接的商务接口人。

- url

https://www.cloudroom.com/CLOUDROOM-SERVER/v2/conference/history/list?searchUserId={searchUserId}&startTime={startTime}&endTime={endTime}¤t={current}&size={size}

- 请求方式 GET
- 请求参数

| 参数名 | 必填 | 类型 | 描述 |
|--------------|----|---------|-------------------------------|
| searchUserId | 否 | Long | 查询用户会议的用户id |
| startTime | 否 | Long | 开始时间（时间戳，单位：秒） |
| endTime | 否 | Long | 结束时间（时间戳，单位：秒，时间间隔最大一年） |
| current | 否 | Integer | 当前页码（默认为1，超出范围使用默认值） |
| size | 否 | Integer | 每页数量（默认为20，最大值为100，超出范围使用默认值） |

- 请求示例

https://www.cloudroom.com/CLOUDROOM-SERVER/v2/conference/history/list?current=1&size=20

json

- 返回参数

| 参数名 | 类型 | 描述 |
|-----------|----------|---------------------|
| total | Integer | 总数 |
| remaining | Integer | 是否还有未拉取的数据（0：否；1：是） |
| records | Object[] | 历史会议列表 |

- 会议列表详情

| 参数名 | 类型 | 描述 |
|------------|---------|--|
| confId | Long | 会议号 |
| subject | String | 会议主题 |
| confType | Integer | 会议类型（0：非预约会议；1：预约会议） |
| startTime | Long | 开始时间（时间戳，单位：秒） |
| endTime | Long | 结束时间（时间戳，单位：秒） |
| hostName | String | 主持人姓名 |
| members | Integer | 参会人数 |
| periodType | Integer | 周期类型（0：非周期会议；1：每天；2：每个工作日；3：每周；4：每两周；5：每 |

| 参数名 | 类型 | 描述 |
|-----|----|----|
|-----|----|----|

- 返回参数示例

```
{
  "total": 1,
  "records": [
    {
      "confId": 12345678,
      "subject": "会议主题",
      "confType": 1,
      "scheduleTime": 1616655277,
      "scheduleEndTime": 0,
      "startTime": 1616655277,
      "endTime": 0,
      "hostName": "云屋科技",
      "members": 0,
      "confCode": "abcdef",
      "periodType": 0
    }
  ]
}
```

json

其他参数

OrderConfig

| 参数名 | 必填 | 类型 | 描述 |
|-----------------|----|---------|---|
| scheduleTime | 是 | Long | 预约开始时间（时间戳，单位：秒） |
| scheduleEndTime | 是 | Long | 预约结束时间（时间戳，单位：秒） |
| periodType | 否 | Integer | 周期类型（0：非周期会议；1：每天；2：每个工作日；3：每周；4：每两周；5：每月，默认为0） |
| periodEndTime | 否 | Long | 周期结束时间（时间戳，单位：秒；-1：无结束时间；如果周期类型不为0，则为必填） |

Attributes

| 参数名 | 必填 | 类型 | 描述 |
|--------------------|----|---------|--|
| autoOpenCamera | 否 | Integer | 入会摄像头状态（-1：使用参会者本地配置；0：自动关闭；1：自动打开，默认为-1） |
| autoOpenMicrophone | 否 | Integer | 入会麦克风状态（-1：使用参会者本地配置；0：自动关闭；1：自动打开，默认为-1） |
| canJoinAdv | 否 | Integer | 是否允许在主持人加入之前入会（0：否，1：是，默认为1） |
| loginIvr | 否 | Integer | 是否播放进入退出提示音（0：否；1：是，默认为1） |
| waterMark | 否 | Integer | 是否开启共享水印（0：否；1：是，默认为0） |
| virtualMembers | 否 | Integer | 虚拟人数（默认为0，最大值为2000） |
| vwallo | 否 | Integer | 默认视频布局：平铺：2000；互看：0；其他分屏：对应数字（默认为2000） |
| waitHall | 否 | Integer | 是否开启等候室（0：否；1：是，默认为0） |
| allowJoin | 否 | Integer | 允许参会权限（0：允许所有用户参会；1：允许企业内用户参会；2：允许邀请对象参会，默认为0） |

Permission

| 参数名 | 必填 | 类型 | 描述 |
|--------------|----|---------|-----------------------------|
| saveWB | 否 | Integer | 是否允许保存白板和演示文档（0：否；1：是，默认为1） |
| publicIM | 否 | Integer | 是否允许群聊（0：否；1：是，默认为1） |
| privateIM | 否 | Integer | 是否允许普通参会者间私聊（0：否；1：是，默认为1） |
| allowPicture | 否 | Integer | 是否允许在聊天窗口发送图片（0：否；1：是，默认为1） |
| allowUrl | 否 | Integer | 是否允许在聊天窗口发送网址（0：否；1：是，默认为1） |
| recordConf | 否 | Integer | 是否允许本地录制（0：否；1：是，默认为0） |

| 参数名 | 必填 | 类型 | 描述 |
|-----------------|----|---------|---------------------------|
| openMicrophone | 否 | Integer | 是否允许打开麦克风（0：否；1：是，默认为1） |
| openCamera | 否 | Integer | 是否允许打开摄像头（0：否；1：是，默认为1） |
| allowShare | 否 | Integer | 是否允许发起共享（0：否；1：是，默认为1） |
| allowModifyName | 否 | Integer | 是否允许改名（0：否；1：是，默认为1） |
| pstnCall | 否 | Integer | 是否允许切换到电话接听（0：否；1：是，默认为1） |

InviteeInfo

| 参数名 | 类型 | 描述 |
|-------------|---------|------------------------------|
| inviteId | Long | 邀请人id |
| inviteeName | String | 邀请人姓名 |
| inviteeType | Integer | 邀请人类型（0：普通用户；1：终端用户；2：sip用户） |

查询录像

- url

https://www.cloudroom.com/CLOUDROOM-SERVER/v2/video/{confId}?current={current}&size={size}

- 请求方式 GET
- 请求参数

| 参数名 | 必填 | 类型 | 描述 |
|---------|----|---------|-------------------------------|
| confId | 是 | Long | 会议号 |
| current | 否 | Integer | 当前页码（默认为1，超出范围使用默认值） |
| size | 否 | Integer | 每页数量（默认为20，最大值为100，超出范围使用默认值） |

- 请求示例

https://www.cloudroom.com/CLOUDROOM-SERVER/v2/video/12345678

json

- 返回参数

| 参数名 | 类型 | 描述 |
|-----------|----------|---------------------|
| total | Integer | 总数 |
| records | Object[] | 录像列表 |
| remaining | Integer | 是否还有未拉取的数据（0：否；1：是） |

- 录像列表详情

| 参数名 | 类型 | 描述 |
|------------|--------|----------------|
| fileId | String | 录像文件id |
| title | String | 录像文件名 |
| confId | Long | 会议号 |
| createTime | Long | 创建时间（时间戳，单位：秒） |
| duration | Long | 录像时长（单位：秒） |
| size | Long | 录像大小（单位：B） |
| vodUrl | String | 播放地址 |
| url | String | 下载地址 |

- 返回参数示例

```
{
  "total": 1,
  "records": [
    {
      "title": "云屋科技的会议",
      "fileId": "a1b27fe0212f42458c38aa86858939ee",
      "confId": 12345678,
```

json

```
    "createTime": 1649840637,  
    "duration": 461,  
    "size": 6950912,  
    "url": "http://www.cloudroom.com/lv//vod/a1b27fe0212f42458c38aa86858939ee/f0.mp4?download",  
    "vodUrl": "http://www.cloudroom.com/vod/a1b27fe0212f42458c38aa86858939ee"  
  },  
],  
  "remaining": 0  
}
```

删除录像

- url

https://www.cloudroom.com/CLOUDROOM-SERVER/v2/video/{fileId}

- 请求方式 DELETE

- 请求参数

| 参数名 | 必填 | 类型 | 描述 |
|--------|----|--------|--------|
| fileId | 是 | String | 录像文件id |

- 请求示例

```
https://www.cloudroom.com/CLOUDROOM-SERVER/v2/video/387702296195782359
```

json

- 返回参数

- 返回参数示例

```
{}
```

json

创建企业

企业管理的相关接口，需要使用平台全局的 `secretId` 和 `secretKey` 进行鉴权

- url

`https://www.cloudroom.com/CLOUDROOM-SERVER/v2/company`

- 请求方式 `POST`
- 请求参数

| 参数名 | 必填 | 类型 | 描述 |
|-----------|----|--------|--------|
| compGUID | 是 | String | 企业唯一标识 |
| compName | 是 | String | 企业名称 |
| admin | 是 | String | 管理员账号 |
| adminName | 是 | String | 管理员昵称 |
| adminPwd | 是 | String | 管理员密码 |

- 请求参数示例

```
{
  "compGUID": "mytestGUID",
  "compName": "测试CompName",
  "admin": "mytest@cloudroom.com",
  "adminName": "测试01",
  "adminPwd": "123456"
}
```

- 返回参数

| 参数名 | 类型 | 描述 |
|-----------|--------|--------|
| compGUID | String | 企业唯一标识 |
| compName | String | 企业名称 |
| admin | String | 管理员账号 |
| adminName | String | 管理员昵称 |

- 返回参数示例

```
{
  "compGUID": "mytestGUID",
  "compName": "测试CompName",
  "admin": "mytest@cloudroom.com",
  "adminName": "测试01"
}
```

修改企业

- url

https://www.cloudroom.com/CLOUDROOM-SERVER/v2/company/{compGUID}

- 请求方式 PUT

- 请求参数

| 参数名 | 必填 | 类型 | 描述 |
|----------|----|--------|------|
| compName | 是 | String | 企业名称 |

- 请求参数示例

```
{
  "compName": "my测试CompName"
}
```

json

- 返回参数

- 返回参数示例

```
{}
```

json

删除企业

- url

```
https://www.cloudroom.com/CLOUDROOM-SERVER/v2/company/{compGUID}
```

- 请求方式 DELETE

- 请求方式示例

```
http://www.cloudroom.com/CLOUDROOM-SERVER/v2/company/234234
```

- 返回参数

- 返回参数示例

```
{}
```

json

创建用户

- url

https://www.cloudroom.com/CLOUDROOM-SERVER/v2/user

- 请求方式 POST
- 请求参数

| 参数名 | 必填 | 类型 | 描述 |
|--------------|----|---------|--|
| account | 是 | String | 账号 唯一标识 |
| password | 是 | String | 密码 |
| name | 是 | String | 昵称 |
| mobile | 否 | String | 手机号 |
| departmentId | 否 | Integer | 部门Id (若部门未填写，账号将默认被分配至公司根目录下) |
| roleType | 否 | Integer | 角色： 1：管理员 2：会议安排者 3：普通用户 4：部门管理员 5：财务管理员 默认为 2：会议安排者 |
| email | 否 | String | 邮箱 |

- 请求参数示例

```
{
  "account": "test@cloudroom.com",
  "password": "123456",
  "name": "test",
  "mobile": "18927450001",
  "departmentId": 102,
  "roleType": 2,
  "email": "test@163.com"
}
```

json

- 返回参数

| 参数名 | 类型 | 描述 |
|--------------|---------|------|
| account | String | 账号 |
| name | String | 昵称 |
| mobile | String | 手机号 |
| departmentId | Integer | 部门Id |
| roleType | Integer | 角色 |
| email | String | 邮箱 |

| 参数名 | 类型 | 描述 |
|-----|----|----|
|-----|----|----|

- 返回参数示例

```
{
  "account": "test@cloudroom.com",
  "name": "test",
  "mobile": "18927450001",
  "departmentId": 102,
  "roleType": 2,
  "email": "test@163.com"
}
```

修改用户

- url

https://www.cloudroom.com/CLOUDROOM-SERVER/v2/user/{account}

- 请求方式 PUT

- 请求参数

| 参数名 | 必填 | 类型 | 描述 |
|--------------|----|---------|---|
| password | 否 | String | 密码 |
| name | 否 | String | 昵称 |
| mobile | 否 | String | 手机号 |
| departmentId | 否 | Integer | 部门Id (若部门未填写，账号将默认被分配至公司根目录下) |
| roleType | 否 | Integer | 角色 (1:管理员, 2:会议安排者, 3:普通用户,4:部门管理员,5:财务管理员。默认为2:会议安排者) |
| email | 否 | String | 邮箱 |

- 请求参数示例

```
{
  "password": "12345678",
  "name": "test",
  "mobile": "18927450001",
  "departmentId": 103,
  "roleType": 2,
  "email": "test@163.com"
}
```

- 返回参数

- 返回参数示例

```
{}
```

删除用户

- url

```
https://www.cloudroom.com/CLOUDROOM-SERVER/v2/user/{account}
```

- 请求方式 DELETE

- 请求参数示例

```
https://www.cloudroom.com/CLOUDROOM-SERVER/v2/user/6c1000e2wesd
```

- 返回参数

- 返回参数示例

```
{}
```

json

创建部门

- url

https://www.cloudroom.com/CLOUDROOM-SERVER/v2/department

- 请求方式 POST
- 请求参数

| 参数名 | 必填 | 类型 | 描述 |
|----------|----|--------|----------------------|
| name | 是 | String | 部门名称 |
| parentId | 是 | Long | 上级部门id(根目录的上级部门id是1) |

- 请求参数示例

```
{
  "name": "test3",
  "parentId": 1668
}
```

json

- 返回参数

| 参数名 | 类型 | 描述 |
|--------------|--------|---------|
| departmentId | Long | 唯一标识 |
| name | String | 名称 |
| parentId | Long | 父级id |
| fullPath | String | 完整的名称路径 |

- 返回参数示例

```
{
  "departmentId": 2562,
  "name": "test3",
  "parentId": 1668,
  "fullPath": "公司 > web > test3"
}
```

json

修改部门

- url

https://www.cloudroom.com/CLOUDROOM-SERVER/v2/department/{departmentId}

- 请求方式 PUT
- 请求参数

| 参数名 | 必填 | 类型 | 描述 |
|----------|----|--------|--------|
| name | 是 | String | 部门名称 |
| parentId | 是 | Long | 上级部门id |

- 请求参数示例

```
{
  "name": "test3",
  "parentId": 1668
}
```

- 返回参数

| 参数名 | 类型 | 描述 |
|--------------|--------|---------|
| departmentId | Long | 部门id |
| name | String | 名称 |
| parentId | Long | 父级id |
| fullPath | String | 完整的名称路径 |

- 返回参数示例

```
{
  "departmentId": 2562,
  "name": "test3",
  "parentId": 1668,
  "fullPath": "公司 > web > test3"
}
```

删除部门

删除部门，部门下的成员，会移动到根目录后，角色修改成普通用户

- url

```
https://www.cloudroom.com/CLOUDROOM-SERVER/v2/department/{departmentId}
```

- 请求方式 DELETE

- 请求方式示例

```
https://www.cloudroom.com/CLOUDROOM-SERVER/v2/department/2545
```

- 返回参数

- 返回参数示例

```
{}
```

json

错误码

鉴权错误码

鉴权失败，HTTP Status Code是400。

响应体是JSON串，包含错误码code和详细的错误信息msg。

| code | msg | 说明 |
|-------|-----------------------------|------------------------|
| 22001 | header传参X-CR-SecretId不能为空 | |
| 22002 | header传参X-CR-Timestamp不能为空 | |
| 22003 | header传参X-CR-Signature不能为空 | |
| 22004 | 签名过期 | 当前请求的时间戳，与服务器时间相差超过5分钟 |
| 22005 | 签名无效 | X-CR-Signature 值不对 |
| 22006 | 账号不存在 | X-CR-SecretId 值找不到该账号 |
| 22007 | 企业未开放API接口权限 | |
| 22009 | header传参X-CR-Timestamp格式不正确 | |

接口调用超出频率，HTTP Status Code是429。

响应体是JSON串，包含错误码code和详细的错误信息msg。

| code | msg |
|-------|--------|
| 22008 | 接口超出频率 |

业务错误码

业务处理失败，HTTP Status Code是500。

响应体是JSON串，包含错误码code和详细的错误信息msg。

| code | msg |
|-------|--------------|
| 400 | 请求地址或参数错误 |
| 500 | 服务器内部异常 |
| 22201 | 会议不存在 |
| 22202 | 会议密码错误 |
| 22203 | 会议未开始 |
| 22204 | 会议已结束 |
| 22205 | 仅邀请对象可加入会议 |
| 22206 | 仅企业内用户可加入会议 |
| 22207 | 获取小程序token异常 |
| 22208 | 企业暂无小程序配置 |

| | | | |
|-------|-------------|------------------|------------|
| 22209 | code | 会议主题不能为空 | msg |
| 22210 | | 会议密码为4-6位数字 | |
| 22211 | | 周期会议结束时间不能为空 | |
| 22212 | | 会议主题不能超过50个字符 | |
| 22213 | | 周期会议结束时间格式错误 | |
| 22214 | | 预约开始时间不能为空 | |
| 22215 | | 预约开始时间不能小于当前时间 | |
| 22216 | | 预约开始时间格式错误 | |
| 22217 | | 预约结束时间不能为空 | |
| 22218 | | 预约结束时间不能小于开始时间 | |
| 22219 | | 预约结束时间格式错误 | |
| 22220 | | 非法的周期会议类型 | |
| 22221 | | 周期会议结束时间不能小于开始时间 | |
| 22222 | | 预约时间不能超过24小时 | |
| 22223 | | 主持人不存在 | |
| 22224 | | 权限不足 | |
| 22225 | | 会议分屏数错误 | |
| 22226 | | 虚拟人数应在0-2000之间 | |
| 22227 | | 会议已开始或已结束 | |
| 22228 | | 邀请人不存在 | |
| 22229 | | 邀请人不能为空 | |
| 22230 | | 昵称不能为空 | |
| 22231 | | 参会者类型错误 | |
| 22232 | | 昵称不能大于30个字符 | |
| 22233 | | 会议记录为空 | |
| 22234 | | 联席主持人个数不能超过四个 | |
| 22235 | | 联席主持人不存在 | |
| 22236 | | 查询时间间隔不能超过一年 | |
| 21030 | | 录像不存在 | |

开发者工具

为了方便开发者快速集成和接入云屋会议API，这里提供了基于java语言的开发工具包。

[工具下载地址](#)

依赖环境

- JDK 7 以上
- 获得secretId和secretKey

使用方式

在自己的项目引入该jar包，初始化 `CRCClient` 类。

```
CRCClient client = new CRCClient(secretId, secretKey);
```

java

工具类 `CRCClient` 封装了对会议API的调用，提供了相应的封装方法。

封装方法

| 功能 | 方法名 |
|--------|---------------------|
| 创建会议 | createMeeting |
| 修改会议 | updateMeeting |
| 查询会议 | queryMeeting |
| 加入会议 | joinMeeting |
| 结束会议 | endMeeting |
| 删除会议 | deleteMeeting |
| 查询会议列表 | queryMeetingList |
| 查询历史会议 | queryHistoryMeeting |
| 查询录像 | queryVideo |
| 删除录像 | deleteVideo |

使用示例

创建一个简单会议

```
String secretId = "LQrJzTFBsSTB90Stjr";
String secretKey = "HxYxy7baX8jn7swVJaFdEx7HpMu0cRX6";
CRCClient client = new CRCClient(secretId, secretKey);
// 构造传参
CreateMeetingRequestBody requestBody = new CreateMeetingRequestBody();
// 会议主题
requestBody.setSubject("api test");
// 构造请求
CreateMeetingRequest request = new CreateMeetingRequest(requestBody);
// 创建会议
```

java

```
// 创建会议
CreateMeetingResponse response = client.createMeeting(request);
```

说明：

- 创建会议，调用 `CRClient` 类的 `createMeeting` 方法
- `CreateMeetingRequest` 类封装了创建会议接口的请求
- `CreateMeetingRequestBody` 类包含了创建会议接口的传参，详细参数见[创建会议接口的请求参数说明](#)
- `CreateMeetingResponse` 类封装了创建会议接口的返回数据，详细参数见[创建会议接口的返回参数说明](#)

预约会议

```
String secretId = "LQrJzTFBsSTB90Stjr";
String secretKey = "HxYxy7baX8jn7swVJaFdEx7HpMu0cRX6";
CRClient client = new CRClient(secretId, secretKey);
// 构造传参
CreateMeetingRequestBody requestBody = new CreateMeetingRequestBody()
    // 会议主题
    .setSubject("api test")
    .setOrderConfig(new OrderConfig()
        // 预约开始时间（时间戳，单位：秒）
        .setScheduleTime(1622980800L)
        // 预约结束时间（时间戳，单位：秒）
        .setScheduleEndTime(1622988000L));
// 构造请求
CreateMeetingRequest request = new CreateMeetingRequest(requestBody);
// 创建会议
CreateMeetingResponse response = client.createMeeting(request);
```

java

说明：

- 创建会议，调用 `CRClient` 类的 `createMeeting` 方法
- `CreateMeetingRequest` 类封装了创建会议接口的请求
- `CreateMeetingRequestBody` 类包含了创建会议接口的传参，详细参数见[创建会议接口的请求参数说明](#)
- `CreateMeetingResponse` 类封装了创建会议接口的返回数据，详细参数见[创建会议接口的返回参数说明](#)

创建一个周期会议

```
String secretId = "LQrJzTFBsSTB90Stjr";
String secretKey = "HxYxy7baX8jn7swVJaFdEx7HpMu0cRX6";
CRClient client = new CRClient(secretId, secretKey);
// 构造传参
CreateMeetingRequestBody requestBody = new CreateMeetingRequestBody()
    // 会议主题
    .setSubject("周期会议")
    .setOrderConfig(new OrderConfig()
        // 预约开始时间（时间戳，单位：秒）
        .setScheduleTime(1622980800L)
        // 预约结束时间（时间戳，单位：秒）
        .setScheduleEndTime(1622988000L)
        // 周期类型（0：非周期会议；1：每天；2：每个工作日；3：每周；4：每两周；5：每月，默认为0）
        // 3 每周
        .setPeriodType(3)
        // 周期结束时间（时间戳，单位：秒；-1：无结束时间；如果周期类型不为0，则为必填）
        .setPeriodEndTime(1622988000L)
    );
// 构造请求
CreateMeetingRequest request = new CreateMeetingRequest(requestBody);
// 创建会议
CreateMeetingResponse response = client.createMeeting(request);
```

java

说明：

- 创建会议，调用 `CRClient` 类的 `createMeeting` 方法
- `CreateMeetingRequest` 类封装了创建会议接口的请求
- `CreateMeetingRequestBody` 类包含了创建会议接口的传参，详细参数见[创建会议接口的请求参数说明](#)
- `CreateMeetingResponse` 类封装了创建会议接口的返回数据，详细参数见[创建会议接口的返回参数说明](#)

修改预约的会议

```
String secretId = "LQrJzTFBsSTB90Stjr";
String secretKey = "HxYxy7baX8jn7swVJaFdEx7HpMu0cRX6";
CRClient client = new CRClient(secretId, secretKey);
// 构造传参
UpdateMeetingRequestBody requestBody = new UpdateMeetingRequestBody()
    // 会议主题
    .setSubject("修改会议")
    .setOrderConfig(new OrderConfig()
        // 预约开始时间（时间戳，单位：秒）
        .setScheduleTime(1623067200L)
        // 预约结束时间（时间戳，单位：秒）
        .setScheduleEndTime(1623074400L));
// 构造请求
UpdateMeetingRequest request = new UpdateMeetingRequest(72235176L, requestBody);
// 修改会议 72235176
UpdateMeetingResponse response = client.updateMeeting(request);
```

说明：

- 修改会议，调用 `CRClient` 类的 `updateMeeting` 方法
- `UpdateMeetingRequest` 类封装了修改会议接口的请求
- `UpdateMeetingRequestBody` 类包含了修改会议接口的传参，详细参数见[修改会议接口的请求参数说明](#)
- `UpdateMeetingResponse` 类封装了修改会议接口的返回数据，详细参数见[修改会议接口的返回参数说明](#)

修改周期会议

```
String secretId = "LQrJzTFBsSTB90Stjr";
String secretKey = "HxYxy7baX8jn7swVJaFdEx7HpMu0cRX6";
CRClient client = new CRClient(secretId, secretKey);
// 构造传参
UpdateMeetingRequestBody requestBody = new UpdateMeetingRequestBody()
    // 会议主题
    .setSubject("修改周期会议")
    .setOrderConfig(new OrderConfig()
        // 预约开始时间（时间戳，单位：秒）
        .setScheduleTime(1623067200L)
        // 预约结束时间（时间戳，单位：秒）
        .setScheduleEndTime(1623074400L)
        // 周期类型（0：非周期会议；1：每天；2：每个工作日；3：每周；4：每两周；5：每月，默认为0）
        // 1：每天
        .setPeriodType(1)
        // 周期结束时间（时间戳，单位：秒；-1：无结束时间；如果周期类型不为0，则为必填）
        .setPeriodEndTime(1623074400L));
// 构造请求
UpdateMeetingRequest request = new UpdateMeetingRequest(15310396L, requestBody);
// 修改会议 15310396
UpdateMeetingResponse response = client.updateMeeting(request);
```

说明：

- 修改会议，调用 `CRCClient` 类的 `updateMeeting` 方法
- `UpdateMeetingRequest` 类封装了修改会议接口的请求
- `UpdateMeetingRequestBody` 类包含了修改会议接口的传参，详细参数见[修改会议接口的请求参数说明](#)
- `UpdateMeetingResponse` 类封装了修改会议接口的返回数据，详细参数见[修改会议接口的返回参数说明](#)

查询会议

```
String secretId = "LQrJzTFBsSTB90Stjr";
String secretKey = "HxYxy7baX8jn7swVJaFdEx7HpMu0cRX6";
CRCClient client = new CRCClient(secretId, secretKey);
// 构造请求
QueryMeetingRequest request = new QueryMeetingRequest(15310396L);
// 查询会议 15310396
QueryMeetingResponse response = client.queryMeeting(request);
```

java

说明：

- 查询会议，调用 `CRCClient` 类的 `queryMeeting` 方法
- `QueryMeetingRequest` 类封装了查询会议接口的请求，详细参数见[查询会议接口的请求参数说明](#)
- `QueryMeetingResponse` 类封装了查询会议接口的返回数据，详细参数见[查询会议接口的返回参数说明](#)

加入会议

```
String secretId = "LQrJzTFBsSTB90Stjr";
String secretKey = "HxYxy7baX8jn7swVJaFdEx7HpMu0cRX6";
CRCClient client = new CRCClient(secretId, secretKey);
// 构造请求
JoinMeetingRequest request = new JoinMeetingRequest(15310396L, "test");
// 加入会议 15310396
JoinMeetingResponse response = client.joinMeeting(request);
```

java

说明：

- 加入会议，调用 `CRCClient` 类的 `joinMeeting` 方法
- `JoinMeetingRequest` 类封装了加入会议接口的请求，详细参数见[加入会议接口的请求参数说明](#)
- `JoinMeetingResponse` 类封装了加入会议接口的返回数据，详细参数见[加入会议接口的返回参数说明](#)

结束会议

```
String secretId = "LQrJzTFBsSTB90Stjr";
String secretKey = "HxYxy7baX8jn7swVJaFdEx7HpMu0cRX6";
CRCClient client = new CRCClient(secretId, secretKey);
// 构造请求
EndMeetingRequest request = new EndMeetingRequest(15310396L);
// 结束会议 15310396
EndMeetingResponse response = client.endMeeting(request);
```

java

说明：

- 结束会议，调用 `CRCClient` 类的 `endMeeting` 方法
- `EndMeetingRequest` 类封装了结束会议接口的请求，详细参数见[结束会议接口的请求参数说明](#)
- `EndMeetingResponse` 类封装了结束会议接口的返回数据，详细参数见[结束会议接口的返回参数说明](#)

删除会议

java

```
String secretId = "LQrJzTFBsSTB90Stjr";
String secretKey = "HxYxy7baX8jn7swVJaFdEx7HpMu0cRX6";
CRCClient client = new CRCClient(secretId, secretKey);
// 构造请求
DeleteMeetingRequest request = new DeleteMeetingRequest(77574744L);
// 删除会议 77574744
DeleteMeetingResponse response = client.deleteMeeting(request);
```

说明：

- 删除会议，调用 `CRCClient` 类的 `deleteMeeting` 方法
- `DeleteMeetingRequest` 类封装了删除会议接口的请求，详细参数见[删除会议接口的请求参数说明](#)
- `DeleteMeetingResponse` 类封装了删除会议接口的返回数据，详细参数见[删除会议接口的返回参数说明](#)

查询会议列表

java

```
String secretId = "LQrJzTFBsSTB90Stjr";
String secretKey = "HxYxy7baX8jn7swVJaFdEx7HpMu0cRX6";
CRCClient client = new CRCClient(secretId, secretKey);
// 构造请求
QueryMeetingListRequest request = new QueryMeetingListRequest(new QueryMeetingParams());
// 查询会议列表
QueryMeetingListResponse response = client.queryMeetingList(request);
```

说明：

- 查询会议列表，调用 `CRCClient` 类的 `queryMeetingList` 方法
- `QueryMeetingListRequest` 类封装了查询会议列表接口的请求
- `QueryMeetingParams` 类包含了查询会议列表接口的传参，详细参数见[查询会议列表接口的请求参数说明](#)
- `QueryMeetingListResponse` 类封装了查询会议列表接口的返回数据，详细参数见[查询会议列表接口的返回参数说明](#)

查询历史会议列表

java

```
String secretId = "LQrJzTFBsSTB90Stjr";
String secretKey = "HxYxy7baX8jn7swVJaFdEx7HpMu0cRX6";
CRCClient client = new CRCClient(secretId, secretKey);
// 构造请求
QueryHistoryMeetingRequest request = new QueryHistoryMeetingRequest(new QueryMeetingParams());
// 查询历史会议列表
QueryHistoryMeetingResponse response = client.queryHistoryMeeting(request);
```

说明：

- 查询历史会议列表，调用 `CRCClient` 类的 `queryHistoryMeeting` 方法
- `QueryHistoryMeetingRequest` 类封装了查询历史会议列表接口的请求
- `QueryMeetingParams` 类包含了查询历史会议列表接口的传参，详细参数见[查询历史会议列表接口的请求参数说明](#)
- `QueryHistoryMeetingResponse` 类封装了查询历史会议列表接口的返回数据，详细参数见[查询历史会议列表接口的返回参数说明](#)

查询录像

java

```
String secretId = "LQrJzTFBsSTB90Stjr";
String secretKey = "HxYxy7baX8jn7swVJaFdEx7HpMu0cRX6";
CRCClient client = new CRCClient(secretId, secretKey);
// 会议号
```

```

Long confId = 35475015L;
// 构造请求
QueryVideoListRequest request = new QueryVideoListRequest(new QueryVideoParams().setConfId(confId));
// 查询录像
QueryVideoListResponse response = client.queryVideo(request);

```

说明：

- 查询录像，调用 `CRClient` 类的 `queryVideo` 方法
- `QueryVideoListRequest` 类封装了查询录像接口的请求
- `QueryVideoParams` 类包含了查询录像接口的传参，详细参数见[查询录像接口的请求参数说明](#)
- `QueryVideoListResponse` 类封装了查询录像接口的返回数据，详细参数见[查询录像接口的返回参数说明](#)

删除录像

```

String secretId = "LQrJzTFBsSTB90Stjr";
String secretKey = "HxYxy7baX8jn7swVJaFdEx7HpMu0cRX6";
CRClient client = new CRClient(secretId, secretKey);
// 文件id
String fileId = "387702296233417094";
// 构造请求
DeleteVideoRequest request = new DeleteVideoRequest(fileId);
// 删除录像
DeleteVideoResponse response = client.deleteVideo(request);

```

java

说明：

- 删除录像，调用 `CRClient` 类的 `deleteVideo` 方法
- `DeleteVideoRequest` 类封装了删除录像接口的请求，详细参数见[删除录像接口的请求参数说明](#)
- `DeleteVideoResponse` 类封装了删除录像接口的返回数据，详细参数见[删除录像接口的返回参数说明](#)

版本历史及Demo下载

Demo下载

[点此下载](#)

版本变更记录

2022-04-20 V1.0.1

[spring-cloudroom-demo-1.0.1.jar](#)

新增录像管理接口：

- 1.查询录像
- 2.删除录像

2021-06-07 V1.0.0

[spring-cloudroom-demo-1.0.0.jar](#)

新增会议管理接口：

- 1.创建会议
- 2.修改会议
- 3.查询会议
- 4.加入会议
- 5.结束会议
- 6.删除会议
- 7.查询会议列表
- 8.查询历史会议列表