

奇虎 360 公司

360 智能摄像机

开放平台接入

V3.1.0

2016/8/4

详细阐述了 360 智能摄像机开放平台的接入流程和开发指导

目录

1. 名词解释	2
2. 开发说明	2
2.1. 关于 sn 号入库	2
2.2. 关于开放平台和 360 智能摄像机 APP	2
2.3. 业务逻辑	3
3. 服务端接口说明	3
3.1. 计算请求签名	3
3.1.1. 计算规则	3
3.1.2. 计算举例	4
3.2. 计算 SN-Token	5
3.2.1. 计算规则	5
3.2.2. 计算举例	5
3.3. 登陆接口	6
3.4. 获取 sn 信息接口	6
3.5. 修改摄像机信息接口	7
3.6. 固件升级接口	7
3.7. 信令接口	8
4. Android SDK	8
5. IOS SDK	8
6. 附录	8
6.1. 开发者常见问题	8
6.2. 错误码说明	9
6.3. 开发者常见咨询	10
6.4. 版本修订说明	10

1. 名词解释

IPC: 360 智能摄像机设备端

app_id: 开放平台分配给开发者的业务 id

app_server_key: 开放平台分配给开发者的服务端请求密钥, 和 app_id 进行匹配

app_sdk_key: 开放平台分配给开发者的 SDK 请求密钥, 和 app_id 进行匹配

uid: 开发者的应用中的用户 id, 开放平台用于标识最终用户

sn: 摄像机的设备序列号, 可以在摄像机的底座或包装盒的背面找到

usid: uid 的身份认证信息, 和 app_id, uid 有关, 和 sn 无关。通过调用开放平台服务端 API 获取, SDK 需要

sn_token: 用户和 sn 在开放平台处的标识, 用来验证用户对 sn 的权限, 和 app_id, uid, sn 都有关系, 由开发者生成, SDK 需要;

计算公式: aes 加密, 密钥是 app_server_key, 加密内容是 expire+', '+app_id+', '+uid+', '+sn

usid, sn_token, 开发方可以缓存。过期后: a) usid 过期, 调用开放平台服务端 API 更新数据; b) sn_token, 调用开发者服务端 API 更新数据

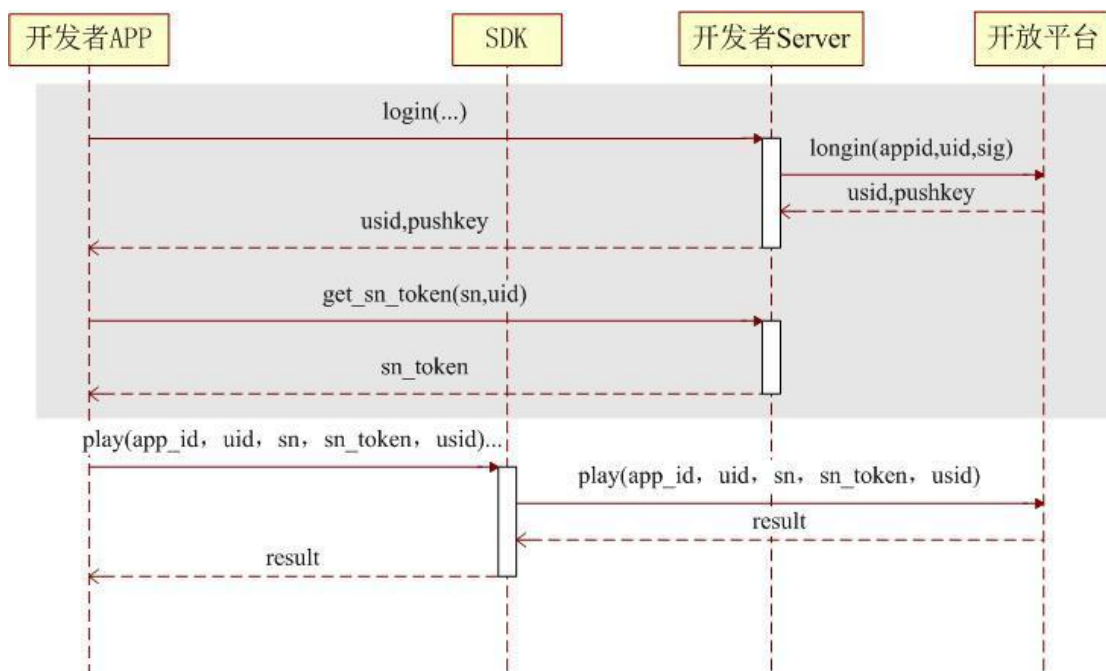
2. 开发说明

2.1. 关于 sn 号入库

所有要使用的设备, sn 号都必须进行入库, 前期的开发测试使用的, 可以联系对接人进行入库, 当正式运营时, 发货后我们会自动入库。

2.2. 关于开放平台和 360 智能摄像机 APP

摄像机在开放平台和 360 智能摄像机 APP 两个系统中是互相冲突的, 入库的设备, 只能在开放平台下以 SDK 的方式使用, 不能在 360 智能摄像机的 app 中使用。没有入库的设备只能在 360 智能摄像机 APP 中使用, 不能在开放平台中使用



2.3. 业务逻辑

开放平台不负责维护终端用户和摄像机的权限关系，对开放平台来说所有的摄像机都归属 app_id，并拥有最高权限。

需要开发者 Server 的参与是出于安全的考虑，用户的身份以及对摄像机的权限需要经过开发者 Server 确认后才能使用 SDK 的功能。

3. 服务端接口说明

注：sig 不做特殊说明均参考登录接口。

接口的字符编码均为 utf-8

新域名：<https://api.dev.jia.360.cn>

旧域名：<https://open.jia.360.cn> (仅支持旧接口)

3.1. 计算请求签名

3.1.1. 计算规则

请求签名的计算公式：md5(request + key)，由两部分组成

其中 request 是 http 请求参数，get 和 post 数据，按照字段名称的自然序 (a-z, 0-9) 排序，这要求 get 和 post 数据不能有字段重复的，并且字段名不能包含 sig；

key 和请求类型相关，如果是开发者 Server 调用，key 为 app_server_key；

如果是开发者 SDK 调用, key 为 app_sdk_key;
计算 sig 值的 php 代码:

```
public function calcSign($params, $signKey)
{
    ksort($params);
    unset($params['sig']);
    $content = "";
    foreach ($params as $k => $v) {
        if (empty($k) || $v === "") {
            continue;
        } else {
            $content .= $k . "=" . $v . "&";
        }
    }
    $content = trim($content, "&");
    return md5($content . $signKey);
}
```

3.1.2. 计算举例

基础数据:

app_id = BCSQOMKSQOMKSQOM
uid = 1000
sn = 36060730406
Server_key = 598c6bca44dc001f2b14d124b24f2da7
Sdk_key = ae2c15ed7242f5f10f52914355470ef9

sig 请求签名计算

计算请求: https://api.dev.jia.360.cn/app/login?uid=1000&app_id=BCSQOMKSQOMKSQOM
的签名

step1: 将参数按照 a-z,0-9 的顺序

排序前:

uid=1000&app_id=BCSQOMKSQOMKSQOM

排序后:

app_id=BCSQOMKSQOMKSQOM&uid=1000

step2: 拼接待加密字符串

params = 'app_id=BCSQOMKSQOMKSQOM&uid=1000';(排序后的参数串)

若请求服务端接口, 则: key = Server_key

若请求 sdk 接口, 则: key = Sdk_key

最终需加密的字符串为(以服务端接口为例):

content = params . key (params 与 key 拼接成字符串)("&"代表拼接, 非字符串中字符)
=app_id=BCSQOMKSQOMKSQOM&uid=1000598c6bca44dc001f2b14d124b24f2da7

step3: 加密

sig = md5(content) = 4f1568b7d3a060206eaa263fbbb72bad

3.2. 计算 SN_Token

3.2.1. 计算规则

计算方法: aes 加密, 秘钥是 app_server_key, 加密内容是 expire+', '+app_id+', '+uid+', '+sn
其中, expire 是 unix 时间戳, 服务端设置为 (当前时间+86400)。
代码如下:

```
public function encryptAES($data, $key)
{
    $blockSize = 16;
    $data = $this->padPKCS5($data, $blockSize);
    $iv = substr($key, 0, $blockSize);
    $encrypted= mcrypt_encrypt(MCRYPT_RIJNDAEL_128,
    $key, $data, MCRYPT_MODE_CBC, $iv);
    return base64_encode($encrypted);
}
public function padPKCS5($data, $blockSize)
{
    $padding = $blockSize - (strlen($data) % $blockSize);
    return $data. str_repeat(chr($padding), $padding);
}
```

3.2.2. 计算举例

基础数据:

```
app_id   = BCSQOMKSQOMKSQOM
uid       = 1000
sn        = 36060730406
Server_key = 598c6bca44dc001f2b14d124b24f2da7
Sdk key   = ae2c15ed7242f5f10f52914355470ef9
```

sn_token 计算

step1: 拼接需加密的字符串(设当前时间为 1470277968)

expire = 当前时间 + 86400 = 1470277968 + 86400 = 1470364368;

需加密的字符串:

content = expire.', '. app_id.', '. uid.', '. sn ("."仅表示拼接, 非字符串中字符)
= 1470364368,BCSQOMKSQOMKSQOM,1000,36060730406

step2: 对需加密字符串进行 aes 加密

加密 key = Server_key

iv 初始化向量(key 的前 16 位): iv = 598c6bca44dc001f

计算得到 aes 加密结果设为 content

step3: base64 加密

sn_token = base64_encode(content)

= 3AMPRP8BgO0hxNzc21BhYJ7tSrnhHeBxvdTaiw6662IOYwHBdKu7Yz8wC0kDmeF

3.3. 登陆接口

请求 uri: /app/login

请求参数:

参数名	说明	必需	描述
app_id	开发者 id	是	
uid	用户 id	是	
sig	请求签名	是	key 为 app_server_key 计算方法见 3.1

响应参数: json 格式

参数名	说明		
errorCode	错误码		0=成功, 其它异常
errorMsg	错误消息		
usid	用户的认证 id		
pushKey	推送密钥		

3.4. 获取 sn 信息接口

请求 uri: /camera/info

请求参数:

参数名	说明	必需	描述
app_id	开发者 id	是	
uid	用户 id	是	
usid	用户验证 id	是	
sn	sn 号, 可支持多个, 逗号分隔	是	
sig	请求签名	是	key 为 app_server_key 计算方法见 3.1

响应参数:

参数名	说明		
errorCode	错误码		0=成功, 其它异常
errorMsg	错误消息		
snData	array(sn0=>array(), sn1=array())		具体的 sn 信息: sn => sn 号 title => 标题 location => 位置 desc=>描述 thumbnail=>封面图 state=>是否在线 0=离线 1=在

			线 fm_status=>固件状态 0 表示最新, 1 表示有升级版本, 2 表示升级中, 3 表示升级失败, 4 表示强制升级
--	--	--	---

3.5. 修改摄像机信息接口

请求 uri: /camera/update

请求参数:

参数名	说明	必需	描述
app_id	开发者 id	是	
uid	用户 id	是	
usid	用户验证 id	是	
sn	sn 号	是	
sn_token	用户和 sn 验证 id	是	
title	摄像机名字	否	
desc	摄像机描述	否	
sig	请求签名	是	key 为 app_server_key 计算方法见 3.1

响应参数:

参数名	说明		
errorCode	错误码		0=成功, 其它异常
errorMsg	错误消息		

3.6. 固件升级接口

请求 uri: /camera/upgradeNewestFW

请求参数:

参数名	说明	必需	描述
app_id	开发者 id	是	
uid	用户 id	是	
usid	用户验证 id	是	
sn	sn 号	是	
sn_token	用户和 sn 验证 id	是	
sig	请求签名	是	key 为 app_server_key 计算方法见 3.1

响应参数:

参数名	说明		
errorCode	错误码		0=成功, 其它异常

errorMsg	错误消息		
----------	------	--	--

注: 该接口异步执行

3.7. 信令接口

请求 uri: /app/commandServer

请求参数:

参数名	说明	必需	描述
app_id	开发者 id	是	
uid	用户 id	是	
usid	用户验证 id	是	
sn	sn 号	是	
sn_token	用户和 sn 验证 id	是	
command	命令	是	
sig	请求签名	是	key 为 app_server_key 计算方法见 3.1

备注:

指令	描述	所需额外传入参数
poweroff	软开关(关)	无
poweron	软开关(开)	无

响应参数:

参数名	说明		
errorCode	错误码		0=成功, 其它异常
errorMsg	错误消息		

4. Android SDK

5. IOS SDK

6. 附录

6.1. 开发者常见问题

1、usid 过期问题

原因 1: usid 未 urlencode, 导致传到服务端时 “+” 被替换为 “ ” (空格)

原因 2: 获取 usid 的 app_id 与请求所用的 app_id 不一致 (多为 app_id 未传参)

2、sig 验证不通过问题

原因 1: 加密 key 用错, 服务端接口用 Server_key, sdk 接口用 Sdk_key

原因 2: 参数未 urlencode 导致计算 sig 时的参数与传入服务端时的参数不一致 (多为 usid,sn_token)

原因 3: 计算 sig 时参数格式化错误, 正确应类似 “app_id=IKMOQSKMOQSKMOQS&sn=36000000000” (中间以 “&” 间隔, 最后没有 “&”, 参数数按照 a-z,0-9 的顺序排列)

原因 4: 参数未传值如: "app_id=IKMOQSKMOQSKMOQS&uid=&sn=36000000000"(uid 未传值, 计算 sig 时应将其过滤掉)

3、sn_token 验证失败

原因 1: sn_token 未进行 urlencode, 导致传到服务端时 “+” 被替换为 “ ”, 验证失败

原因 2: sn_token 计算错误

4、sn_token 计算错误

原因 1: 缺少 aes 加密所需的 jar 包, 导致 key.getBytes 错误

解决: AES 加密用到的两个 jar 包, 替换原来安装的 %JAVA_HOME%\jre\lib\security 下的 local_policy.jar 和 US_export_policy.jar

原因 2: 未设置 iv 初始化向量

注意 1: 128 位 AES 加密(CBC 模式)然后对加密结果 Base64 编码得到 sn_token

注意 2: 加密的密钥为 Server_key, iv 初始化向量为密钥的前 16 位

5、绑定不成功, 报 13011, 找不到绑定记录

原因 1: ts 参数未传或与激活时值不一致

原因 2: 激活后立即绑定不成功, 中间有个时间差, 一般绑定轮询 2 分钟

6.2. 错误码说明

100	缺少必要的参数
400	无效的应用 id
440	签名验证失败
441	摄像机已经离线
13001	sn_token 解析失败
13002	usid 过期
13003	sn_token 过期
13004	usid 验证失败
13005	sn_token 验证失败
13006	usid 解析失败
13007	sn 不属于该企业
13009	图片签名验证失败
13011	无绑定记录
13012	添加绑定记录失败
1000326	获得摄像机固件信息失败
1000327	升级摄像机固件失败
1000315	给摄像机发送命令失败
1000319	更新摄像机信息失败

6.3. 开发者常见咨询

1、流畅、高清、超清的像素分别是多少么？

流畅是 960*640，高清和超清都是 1280×720

2、上传带宽最少需要多大

摄像机有 3 种模式，流畅（20KB），高清（60KB），超清（130KB）

1KB = 8Kb

3、如果用 360 摄像头做直播，按 1000 观看人次，我们公司的网络大概需要多大的带宽合适？

对于摄像机这端的网络，和多少人看没有关系，是我们服务端转发的，我们服务端支持海量用户同时观看

4、蓝色灯常亮是代表什么意思？闪烁呢？

蓝灯亮表示摄像机连上网，闪烁表示摄像头正在有人播放

5、云录像是连续录制的吗？

云录像是事件触发式的,不是连续录制的,摄像机里有算法,检测到画面有变化的时候才录,录的时长和画面变化的持续时长有关(一般为 30s 左右)

6、摄像机变蓝灯之后突然接变为绿灯闪烁，然后又变为蓝灯，一直在交叉变动，然后视频不能看的，是摄像机硬件问题么？

如果确认网络没有问题，那就是摄像机硬件有问题，这个现象就是频繁上下线

7、sdk 支持多个摄像头同时实时播放吗？

app 端是不支持的

8、播放视频时候右下角的水印能够去掉么

不能去掉，这是公司要求的，我们的品牌展现

9、你们现在使用的是什么协议获取的视频流

视频走的是私有协议,使用 tcp

6.4. 版本修订说明

v1.0.1 2015.11.30

新建文档

v1.0.3 2015.12.2

明确 app_server_sig 和 app_sdk_sig 的计算公式

明确接口的相关信息（域名，uri，参数，编码）

v1.0.5 2015.12.17

规范文档

计算请求签名微调

v2.0.0 2016.3.24

统一规范名称

V3.0.0 2016.5.5

新增部分接口

接口微小改动

常见问题总结

V3.1.0 2016.8.4

新版域名与接口

新增部分接口

新增计算 demo

常见问题总结

常见咨询总结