1 API Definitions

1.1 Request Data

- POST:
    - Request Header: content-type: application/x-www-form-urlencoded.

1.2 Request Return

- Return Data: If the returned HTTP status code is 200, return json data. Otherwise, it means request error. Get the error details according to the API error code.

1.3 Public Return Code

Return Code	Return Information	Description
200	Operating succeeded	Request succeeded
10007	Call times reached the limit	
10029	Call frequency reached the limit	API call frequency reached the limit

1.4 Parameter Remarks

accessToken is the token for login. Please get it via background and ensure its confidentiality.

1、API List

This section contains the API used for the admin user to get the accessToken via appKey and secret.

See the following list:

No.	Function	Description
1	Get accessToken via appKey and secret	Used for the administrator to get the accessToken

1.1 Get accessToken via appKey and secret

- API Function

This API is used for the admin user to get the accessToken。

- Request Address

https://open.ezvizlife.com/api/lapp/token/get

- Request Type

POST

- Request Parameters

Parameter Name	Type	Description	Required
appKey	String	appKey	Y
appSecret	String	appSecret	Y

- HTTP Request Message

POST /api/lapp/token/get HTTP/1.1

Host: open.ezvizlife.com

Content-Type: application/x-www-form-urlencoded

appKey=9mqitppidgce4y8n54ranvyqc9fjtsrl&appSecret=096e76501644989b63ba0016ec5776

- Return Data

{

"data": {

"accessToken": "at.7jrcjmna8qnqg8d3dgnzs87m4v2dme3l-32enpqgusd-1jvdfe4-uxo15ik0s",

"expireTime": 1470810222045,

"areaDomain": "https://iusopen.ezvizlife.com"

},

"code": "200",

"msg": "Operating succeeded!"

}

- Return Field

Field Name	Type	Description
accessToken	String	Obtained accessToken
expireTime	long	Accurate to millisecond
areaDomain	String	the open api domain name of the user's region, the accessToken is valid only in this region

- Return Code

Return Code	Return Information	Description
200	Operating succeeded.	Request succeeded.
10001	The parameter is empty or incorrect format.	The parameter is empty or incorrect format.
10005	appKey exception	appKey is frozen.
10017	appKey does not exist.	Check the correctness of appKey
10030	appkey and appSecret mismatched.	
49999	Data exception.	API call exception.

1、Play Address Interface

This section includes related interfaces of play address.

The interfaces are listed as follows:

Serial No	Interface Function	Description
1	Obtain play address	Obtain play address of the device
2	Invalidate play address	Invalidate play address of the device

1.1Obtain Play Address

- Function

This interface is used to obtain play address of the single device through device serial number and channel number.

- Request Address

{areaDomain}/api/lapp/live/address/get

- Request Method

POST

- Request Parameters

Parameters	Type	Description	Required
accessToken	String	Access token obtained in authorization process	Y
deviceSerial	String	The serial Number of the device, e.g. 427734222, all in English notation, limited to a maximum of 50 characters.	Y
channelNo	Integer	Channel number, not required, the default is 1	N
protocol	Integer	Stream protocol, 1-ezopen, 2-hls, 3-rtmp, 4-flv, the default is 1	N
code	String	Video encryption password of device in the ezopen protocol address	N
expireTime	Integer	Expiration rime, the unit is second; set validity period for hls/rtmp/flv: 30s-720d	N

Parameters	Type	Description	Required
type	String	Address type, 1-preview, 2- local recording playback, 3-CloudPlay recording playback, not required, the default is 1	N
quality	Integer	Video quality, 1-HD (main bitrate), 2-Fluent (sub-bitrate), the default is 1.	N
startTime	String	The playback start time of the local recording or CloudPlay recording, e.g. 2019-12-01 00:00:00When type != 1 and Protocol != 1: (1) startTime and stopTime are empty, the stopTime is the current time and the startTime is the previous day of the current time; (2) the stopTime is empty and the startTime is not empty, the stopTime is the day after the startTime; (3) the startTime is empty and the stopTime is not empty, the startTime is the previous day of the stopTime.	N
stopTime	String	The playback stop time of the local recordings or CloudPlay recordings, e.g. 2019-12-01 00:00:00	N

- HTTP Request Post

POST /api/lapp/live/address/get HTTP/1.1

Host: iusopen.ezvizlife.com

Content-Type: application/x-www-form-urlencoded

Content-Length: 110

accessToken=at.bju93z4w2iifhu1zbxl7phrz8852juxg-99skn9j3kf-05iffrm-ugesv5l9h&deviceSerial=F00497273&protocol=2

- Return Data

{

"msg": "Operation succeeded",

"code": "200",

```
  "data": {
    "id": "512628410958159872",
    "url":
"https://iusopen.ezvizlife.com/v3/openlive/F00497273_1_1.m3u8?expire=1668578
537&id=512628410958159872&t=56528d84c512aad8b4cfdabb43323e2ce692de0
22cdd9b0dce595a7d8677bd61&ev=100",
    "expireTime": "2022-11-16 06:02:17"
  }
}
```


- Return Field：

Filed Name	Type	Description
code	String	Status code, refers to the return code below. The error code is preferred to be judged, return 200 indicates success
msg	String	Status description
id	String	Live address ID
url	String	Live address
expireTime	String	Expiration time

- Return Code：

Return Code	Return Message	Note
200	Operation successfully, the live address within validity period is obtained	Request successfully

1.2Invalidate Play Address

- Function

This interface is used to invalidate the obtained playing address

- Request Address

`{areaDomain}/api/lapp/live/address/disable

- Request Method

POST

- Request Parameters

Parameters	Type	Description	Required
accessToken	String	Access token obtained in authorization process	Y
deviceSerial	String	The serial Number of the device, e.g. 427734222, all in English notation, limited to a maximum of 50 characters.	Y
channelNo	Integer	Channel number, not required, the default is 1	N
urlId	String	Live address ID	N

- HTTP Request Report

POST /api/lapp/live/address/disable HTTP/1.1

Host: iusopen.ezvizlife.com

Content-Type: application/x-www-form-urlencoded

Content-Length: 136

accessToken=at.bju93z4w2iifhu1zbxl7phrz8852juxg-99skn9j3kf-05iffrm-ugesv5l9h&deviceSerial=F00497273&channelNo=1&urlId=512638098828922880

- Return Data


```
{
  "msg": "Operation succeeded",
  "code": "200"
}
```

- Return Code：

Return Code	Return Message	Note
200	Operation successfully, the live address within validity period is obtained	Request successfully

Query local video record（GET）

query local video record

URL

{areaDomain}/api/v3/das/device/local/video/query

Request

Header

Parameter	Type	Required	Description	Example
accessToken	string	Y	accessToken	
deviceSerial	string	Y	device serial	
channelNo	string	N	channel number	

query

Parameter	Type	Required	Description	Example
recordType	int	N	record type 1 timing record 2 event record 3 intelligent-car 4 intelligent-human The default value is all types	
startTime	string	N	Start time, format is 2022-08-22 13:59:13, optional, the default value is 0 o'clock on the day	
endTime	string	N	End time, format i is 2022-08-22 13:59:13, optional, the default value is the current time	

Response

return parameter

Parameter	Type	Description	Example
startTime	string	Start time	
endTime	string	End time	
deviceSerial	string	Device serial	

Parameter	Type	Description	Example
cameraNo	string	Channel number	
type	string	Record type	
size	string	Record size	

return data


```
{
        "meta": {
                "code": 200,
                "message": "操作成功",
                "moreInfo": null
        },
        "data": [
                {
                        "startTime": "2022-11-10T11:44:15",
                        "endTime": "2022-11-10T11:51:21",
                        "deviceSerial": "J67757598",
                        "cameraNo": "1",
                        "type": "TIMING",
                        "size": ""
                }
        ]
}
```


Error code

Return Code	Error code	Description	Solution
200	200	Operation succeeded	
200	2003	Device offline	

Return Code	Error code	Description	Solution
200	2007	Wrong device number	
200	2009	The device response times out	
200	2030	Device not support	
401	10002	accessToken exception or expired.	
200	20015	Device not support this function	

Device capability set description

The device capability set is a set of features used to characterize the capabilities of the device. Based on the device capability set, you can clearly know which functions the device supports.

Due to the continuous update of the ability assembly, if you find a field that does not exist in this description, please promptly feedback open-team@ezvizlife.com

The user can obtain the device capability set by calling the [Query Device Capability Set] (#api_capacity) interface

Serial Number	Field	Capability Set Field Description
1	support_defence	Whether to support arming and disarming, activity detection switch
2	support_talk	Whether to support intercom: 0-not supported, 1-full duplex, 3-half duplex
3	support_defenceplan	Whether to support arming and disarming plan 0-not supported, 1-supported, 2-supported new device plan agreement
4	support_disk	Whether to support storage format 0-not supported, 1-supported
5	support_privacy	Whether to support privacy protection 0-not supported, 1-supported
6	support_message	Whether to support the message 0-not supported, 1-supported
7	support_alarm_voice	Whether to support alarm sound configuration 0-not supported, 1-supported
8	support_auto_offline	Whether to support the device to automatically go online and offline 0-not supported, 1-supported
9	supprot_encrypt	Whether to support video image encryption 0-not supported, 1-supported

Serial Number	Field	Capability Set Field Description
10	support_upgrade	Whether to support device upgrade 0-not supported, 1-supported
11	support_cloud	Whether the device model supports cloud storage 0-not supported, 1-supported
12	support_cloud_version	Whether the device version supports cloud storage 0-not supported, 1-support needs to be combined with support_cloud: support_cloud = 1, support_cloud_version = 1 only supports cloud storage support_cloud = 1, support_cloud_version = 0, this type of device supports cloud storage, but the current firmware version does not support cloud storage. Support_cloud = 0 This type of device does not support cloud storage.
13	support_wifi	Whether to support WI-FI: 0-not supported, 1-support netsdk configuration WI-FI, 2-support new WI-FI configuration with userId, 3-support one Key configuration WI-FI
14	support_capture	Whether to support cover capture: 0-not supported, 1-supported
15	support_modify_pwd	Whether to support changing device encryption password: 0-not supported, 1-supported
16	support_resolution	Video playback ratio 16-9 means 16: 9 resolution, default 16-9
17	support_multi_screen	Whether to support multi-screen playback 0-not supported, 1-supported (use by client, regardless of device)

Serial Number	Field	Capability Set Field Description
18	support_upload_cloud_file	Whether to support mobile phone camera upload to cloud storage `0-not supported, 1-supported
19	support_add_del_detector	Whether to support app to remotely add and remove peripherals (detectors): 0-not supported, 1-supported
20	support_ipc_link	Whether to support IPC and A1 linkage relationship settings: 0-not supported, 1-supported
21	support_modify_detectorname	Whether to support modifying peripheral (detector) name: 0-not supported, 1`-supported
22	support_safe_mode_plan	Whether to support switching safety function mode regularly: 0-not supported, 1-supported
23	support_modify_detectorguard	Does the A1 device support separate arming and disarming: This field does not exist: Not supported This field exists: each peripheral is separated by a comma, as listed in table order, each value is 32 bits The value indicates whether each mode can be set in each mode, if a certain detector can be set to enable this parameter, the position is 1,for example "support_modify_guard": "0,0,7,7, 7,0,7,0,0,0 "is the following ability description

detector type	outgoing mode (bit0)	Sleep mode (bit1)	Home mode (bit2)
Smoke feeling	0	0	0

Serial Number	Field	Capability Set Field Description			
		emergency button	0	0	0
		Door magnet	1	1	1
		Infrared	1	1	1
		Curtain	1	1	1
		Emergency button	0	0	0
		Single door magnet	1	1	1
		Sirens	0	0	0
		Gas detector	0	0	0
		Flood detector	0	0	0
24	support_weixin	Whether the detector type supports WeChat interconnection: 0-not supported, 1-supported			
25	support_ssl	Whether to support sound source localization: 0-not supported, 1-supported			
26	support_related_device	Whether to support the associated device 0-no associated device, 1-associated monitoring point or N1, 2-associated detector or A1, 3-associated monitoring point detector or R1, 4associated multi-channel device			

Serial Number	Field	Capability Set Field Description
27	support_related_storage	Does NVR / R1 support related IPC storage: 0-not supported, 1-supported
28	support_remote_auth_randcode	Whether to support device remote authorization to obtain password, 0-not supported, 1-supported
29	support_sdk_transport	Whether to support the ability level of device cross-network configuration: 0-not supported, 1-supported
30	ptz_top_bottom	Whether to support PTZ up and down rotation 0-not supported, 1-supported
31	ptz_left_right	Whether to support pan / tilt rotation 0-not supported, 1-supported
32	ptz_45	Whether to support pan / tilt rotation at 45 degrees 0-not supported, 1-supported
33	ptz_zoom	Whether to support PTZ zoom control 0-not supported, 1-supported
34	support_ptz	Whether to support PTZ control 0-not supported, 1-supported, Note: The capability set of the new device is split into four capabilities of 30-33
35	ptz_preset	Whether to support PTZ preset point 0-not supported, 1-supported
36	ptz_common_cruise	Whether to support normal cruise 0-not supported, 1-supported
37	ptz_figure_cruise	Whether to support pattern cruise 0-not supported, 1-supported
38	ptz_center_mirror	Whether to support center mirroring 0-not supported, 1-supported

Serial Number	Field	Capability Set Field Description
39	ptz_left_right_mirror	Whether to support left and right mirroring 0-not supported, 1-supported
40	ptz_top_bottom_mirror	Whether to support up and down mirroring 0-not supported, 1-supported
41	ptz_close_scene	Whether to support close lens 0-not supported, 1-supported
42	support_wifi_2.4G	Whether to support 2.4G wireless frequency band 0-not supported, 1-supported
43	support_wifi_5G	Whether to support 5G wireless frequency band 0-not supported, 1-supported
44	support_wifi_portal	Whether to support marketing wifi, only take effect when support_wifi_2.4G = 1: 1-support but cannot set marketing page (X1), 2-support and can set marketing page,0-Not supported
45	support_unbind	Whether to support the user to unbind the device 0-not supported, 1-support reset button to unbind, 2-support interface click OK button to unbind
46	support_auto_adjust	Whether to support adaptive stream 0-not supported, 1-supported
47	support_timezone	Whether to support time zone configuration 0-not supported, 1-supported
48	support_language	Supported language types: ENGLISH, SIMPCN, ...
49	support_close_infrared_light	Whether to support infrared switch 0-not supported, 1-supported

Serial Number	Field	Capability Set Field Description
50	support_modify_chan_name	Whether to support channel name configuration to the device (IPC / NVR) 0-not supported, 1-supported
51	support_ptz_model	0-support direct connection + forward PTZ control, 1-support direct connection PTZ control, 2-support forward PTZ control
52	support_talk_type	0-use the microphone above, 1-use the microphone below for intercom
53	support_chan_type	Channel type, 1-digital channel, 2-analog channel
54	support_flow_statistics	Whether to support passenger flow statistics 0-not supported, 1-supported
55	support_more	Whether to support the device setting function 0-not supported, 1-supported Note: "More configuration" is added on the device settings page, this item is implemented according to [device capability level], and more configurations enter H5 Web display
56	support_remote_quiet	Does A1 support remote alarm (silent) function 0-not supported, 1-supported
57	support_customize_rate	Whether to support custom bit rate 0-not supported, 1-supported
58	support_rectify_image	Whether to support deformity correction 0-not supported, 1-supported
59	support_bluetooth	Whether to support Bluetooth 0-not supported, 1-supported
60	support_p2p_mode	The default is 0, which means the old p2p protocol; configuration is 1, which means

Serial Number	Field	Capability Set Field Description
		that this version supports the new p2p protocol
61	support_microscope	Whether to support the microscope function 0-not supported, 1-supported
62	support_sensibility_adjust	Whether to support motion detection sensitivity adjustment 0-not supported, 1-supported
63	support_sleep	Whether to support sleep function 0-not supported, 1-supported
64	support_audio_onoff	Whether to support audio switch setting 0-not supported, 1-supported
65	support_protection_mode	0: No protection mode, there may be activity detection (based on support_denfence (serial number 1)) 1: Only protection mode 2: There is protection mode, there may be activity detection (based on support_denfence (serial number 1)) Example of capability level configuration: support_protection_modesupport_denfenceA111 ordinary IPC01C1S21
66	support_rate_limit	Whether to support HD bit rate limit 0-does not support bit rate limit, 1-supports HD bit rate limit
67	support_userId	Is it supported to associate device via UserID 0-not supported, 1-supported
68	support_music	Whether to support the children's song playback function 0-not supported, 1-supported

Serial Number	Field	Capability Set Field Description
69	support_replay_speed	Whether to support the function of adjusting playback speed 0-not supported, 1-supported (only supported by IPC)
70	support_reverse_direct	Whether to support reverse direct connection function 0-not supported, 1-supported
71	support_channel_offline_notify	Whether to support channel offline notification, after support, channel offline will trigger ideoloss alarm 0-not supported, 1-supported
72	support_fullscreen_ptz	Whether to support the panoramic pan / tilt function 0-not supported, 1-supported (supported by C6B and other pan / tilt cameras). If the capability set support_fullscreen_ptz_12 (serial number 82) exists, please refer to the capability set support_fullscreen_ptz_12
73	support_preset_alarm	Whether to support preset alarm linkage 0-not supported, 1-supported (supported by C6B and other PTZ cameras)
74	support_intelligent_track	Whether to support intelligent tracking 0-not supported, 1-supported (C6B and other PTZ camera support)
75	support_key_focus	Whether to support one-key focus 0-not supported, 1-supported (supported by F1, F2 and other zoom cameras)
76	support_volumn_set	Whether to support volume adjustment 0-not supported, 1-supported
77	support_temperature_alarm	Whether to support temperature and humidity alarm 0-not supported, 1-

Serial Number	Field	Capability Set Field Description
		supported (F2, C1S and other cameras with temperature and humidity sensor support)
78	support_mcvolumn_set	Whether to support microphone volume adjustment: 0-not supported, 1-supported
79	support_unlock	Whether to support unlocking support 0-not supported, 1-supported
80	support_noencript_via_antproxy	Does it support the ability to automatically encrypt the "no video encryption" stream when going through the proxy 0-not supported, 1-supported
81	support_device_offline_notify	Whether to support device offline notification 0-not supported, 1-supported
82	support_fullscreen_ptz_12	Whether to support the panoramic pan / tilt function 0-not supported, 1-supported (C6B and other pan / tilt camera support, 12 panoramic pan / tilt pictures)
83	support_speech_recognition	Whether to support speech recognition 0-not supported, 1-supported
84	support_message_cover	Whether to support message cover 0-not supported, 1-supported
85	support_nat_pass	Whether to support NAT combination with NAT combination of 3-4 (P2PV2.1) 0-not supported, 1-supported
86	support_nvr_whitelist	Whether NVR supports whitelist member management 0-not supported, 1-supported
87	support_voice_alarmclock	Whether to support voice alarm function 0-not supported, 1-supported
88	support_new_talk	Whether to support the new intercom service 0-not supported, 1-supported

Serial Number	Field	Capability Set Field Description
89	support_fullday_record	Whether to support all-day recording configuration switch 0-not supported, 1-supported
90	support_query_play_connections	Whether to support querying current preview, playback link information 0-not supported, 1-supported
91	support_ptz_auto_reset	Whether to support PTZ auto reset 0-not supported, 1-supported
92	support_fisheye_mode	Whether to support fisheye mode 0-not supported, 1-supported
93	support_custom_voice	Whether to support custom voice 0-not supported, 1-supported (voice alarm clock, alarm sound use)
94	support_new_sound_wave	Whether to support sound wave configuration (high frequency version) 0-not supported, 1-supported
95	replay_chan_nums	Number of channels that can be associated with X3 or N1
96	support_horizontal_panoramic	Whether to support horizontal panorama 0-not supported, 1-supported
97	support_active_defense	Whether to support active defense function: 0-not supported, 1-active defense button, 2-active defense button + light reminder switch
98	support_motion_detect_area	Whether to support motion detection area drawing 0-not supported, 1-supported
99	support_chan_defence	Whether to support channel arming and disarming 0-not supported, 1-supported

Serial Number	Field	Capability Set Field Description
100	ptz_focus	Whether to support focal length mode 0-not supported, 1-supported
101	support_pir_detect	Whether to support infrared detection capability 0-not supported, 1-supported (cat eye)
102	support_doorbell_talk	Whether to support the doorbell call ability 0-not supported, 1-supported (cat eye)
103	support_face_detect	Whether to support face detection capability 0-not supported, 1-supported (cat eye)
104	support_restart_time	Device restart time, the configuration unit is seconds, the default is 120s
105	support_human_filter	Whether to support human filtering capability 0-not supported, 1-supported) (C5SI model, the device is supported by smart chip hardware)
106	support_human_service	Whether to support humanoid detection capability 0-not supported, 1-supported (device + platform service is activated to realize humanoid detection service capability device can be supported by updating software version)
107	support_ap_mode	Whether to support adding device to configure WiFi use, 0: not supported, 1: smartconfig + sound wave failure, support AP configuration network, 2: device default AP configuration network
108	support_continuous_cloud	Whether to support continuous cloud storage 0-not supported, 1-supported, note: it has nothing to do with support_cloud (serial number 11)

Serial Number	Field	Capability Set Field Description
109	support_doorbell_sound	Whether to support focal length mode 0-not supported, 1-supported
110	support_associate_detector	Whether to support association detector 0-not supported, 1-supported
111	support_modify_username	Whether to support the modification of the user's note name of the door lock 0-not supported, 1-supported
112	support_transfertype	Preview streaming format transfer type: 0-tcp, 1-udp, default 0 means tcp
113	support_vertical_panoramic	Whether to support vertical panorama (corresponding to support_horizontal_panoramic (serial number 96)) 0-not supported, 1-supported
114	support_alarm_light	Whether to support security lights 0-not supported, 1-supported
115	support_alarm_area	Whether to support security lights 0-not supported, 1-supported
116	support_chime	Whether to support doorbell extension 0-not supported, 1-supported
117	support_video_mode	Whether to support support_video_mode 0-not supported, 1-supported
118	support_relation_camera	Whether to support W2D related camera function 0-not supported, 1-supported
119	support_pir_setting	Whether to support PIR (infrared) area setting 0-not supported, 1-supported
120	support_battery_manage	Whether to support battery management 0-not supported, 1-supported

Push Message System (webhook)

1.Introduction

Webhook: a web custom callback entry that automatically calls the specified URL when some behavior is triggered by the program.

Compared with message pull mode, message push mode enables messages to reach the client system in a more real-time manner, and the complexity of the client system will be reduced (it is no longer necessary to start a scheduled task to call the message pull interface). The disadvantage is that when an exception occurs at the consumer end, the message producer continues to produce messages. Even though we have a retry mechanism, the messages may still be lost due to the limited number of retries.

Because of the characteristics of webhook, when a large number of messages are generated, the message throughput mainly depends on the processing efficiency of the customer webhook message processing service. To ensure the quality of service, the timeout period set for each message push request is 2 seconds. After the processing timeout, it is considered to have failed to send this message. If the retry mechanism is configured, the message will be sent again using the retry mechanism. If the maximum number of retries fails, the message will be discarded and will not be pushed again. It is recommended that developers quickly write the push messages into their own message queue after receiving them to improve the message receiving ability.

Please contact us via email open-team@ezvizlife.com if you'd like receive the message from camera and sensor through message service.

2. Request Protocol Description

Request Method

HTTP POST Content-Type: text/plain

Note: The current http request body is in json format, but the Content-Type of the request header is text/plain.

Request Header

Field	Name	Description
t	Time stamp	Message sending time stamp.

Field	Name	Description
signature	Signature	(Optional) Use hmac+sha1 to sign the data in the message body. See the following security related sections for details.
message_type	Message type	Consistent with the message type in the message subscription, for example: ys.alarm(alarm message), ys.calling(calling message), etc.

Description of header content in the request body

Field	Name	Description
channelNo	Channel number	
deviceId	Device serial number	
messageId	Message id	
messageTime	Message push time	
type	Message type	

Note: The data in the body is the message reported by the transparent transmission device.

Request Body

JSON format, consistent with the message body structure in message subscription results.

Request body example


```
{
  "body": {
    "data": "0",
    "index": 24409
  },
  "header": {
    "channelNo": 1,
    "deviceId": "D98462102",
    "messageId": "5e57f239793f2b007fecb0de",
```

```
    "messageTime": 1582821945396,
    "type": "ys.open.isapi"
  }
}
```


Response

For a webhook push, if the http return code of the customer webhook service is 200 and the returned content contains the push request message messageId, the push is successful.

Return example


```
{
  "messageId": "5e57f239793f2b007fecb0de"
}
```


Security

1. The customer must provide https url as the webhook address;
2. If the signing key is provided when the service is opened, the push request header contains the signature of the message body (the customer verifies the signature after receiving the message); Signature method ： signature=hmac_sha1(HTTP request body+ time stamp, secret configured when the customer enables the message push)

3. Cautions

1. The customer system must design and implement the deployment of the webhook message processing service according to its own device message volume. If the customer webhook message service has insufficient processing capability and causes its own system failure, messages may be lost (a retry mechanism can be configured to retry, but the timeliness will decline, and if the failure lasts too long, the messages may still be lost). It is recommended that the webhook message processing service implemented by the customer directly forwards the push messages to the persistent message queue when receiving them, and then the later business system will consume these messages for business logic processing, so as to improve the efficiency of webhook message pushing and receiving.
2. The timeout period of the push is 2 seconds. If the customer webhook service takes more than 2 seconds to process the push message, the push

will be considered as a failure and the subsequent processing operations for the push failure will be performed.

3. If the customer webhook message processing service needs the whole cluster to be shut down for upgrading (the service is completely unavailable), there is a risk of message loss. Therefore, if the whole cluster is shut down for maintenance, a processing plan should be prepared.

4. Information Provided When Webhook Is Enabled

1. Webhook callback address
2. Push message type (eg： ys.alarm / ys.open.isapi / ys.calling / ys.onoffline / ys.open.ram / ys.iot / ys.auth.update)
3. Signing key (optional)
4. Maximum number of retries for push failure (optional, value range: 1~3)
5. Email address for service downgrade notification (optional)

Appendix 1: Signature sample code (java version)

```
public static String hmacSha1(String key, String data, Charset bytesEncode)
throws NoSuchAlgorithmException, InvalidKeyException {
  SecretKeySpec signingKey = new SecretKeySpec(key.getBytes(bytesEncode),
"HmacSHA1");
  Mac mac = Mac.getInstance("HmacSHA1");
  mac.init(signingKey);
  // General method, just implement it yourself (from byte[] to hex string)
  return byteToHexString(mac.doFinal(data.getBytes(bytesEncode)));
}


...
String secret = ... ;//Signing key
String body = ...; //HTTP request body
String timestamp = ...; //HTTP header，field：t
//Verify signature
signature.equals(hmacSha1(secret, body + timestamp, StandardCharsets.UTF_8))
```

Appendix 2: Callback address receiving template

```
@RequestMapping(value = "/webhook")
public ResponseEntity<String> webhook(@RequestHeader HttpHeaders header,
@RequestBody String body) {
  final List<String> t = header.get("t");
  WebhookMessage receiveMessage = null;
  log.info("Message acquisition time:{}, request header:{},request
body:{}",System.currentTimeMillis(),JSON.toJSONString(header),body);
  System.out.println("Message received:"+body);
  try {
    receiveMessage = JSON.parseObject(body, WebhookMessage.class);
    //todo: Process the received message. It is better to send it to other
middleware or write it to the database, without affecting the processing of the
callback address

  } catch (Exception e) {
    e.printStackTrace();
}
//Must return
  Map<String, String> result = new HashMap<>(1);
  assert receiveMessage != null;
  String messageId = receiveMessage.getHeader().getMessageId();
  result.put("messageId", messageId);
  final ResponseEntity<String> resp =
ResponseEntity.ok(JSON.toJSONString(result));
  log.info("Information returned:{}",JSON.toJSONString(result));
  return resp;
}
```

1. Error code description:

Return Value	Description	Remark
200	Operation completed	
1001	Invalid user name	
1002	The user name is occupied	
1003	Invalid password	
1004	Duplicated password	
1005	No more incorrect password attempts are allowed	
1006	The phone number is registered	
1007	Unregistered phone number	
1008	Invalid phone number	
1009	The user name and phone does not match	
1010	Getting verification code failed	
1011	Incorrect verification code	
1012	Invalid verification code	
1013	The user does not exist	
1014	Incorrect password or appKey	
1015	The user is locked	
1021	Verification parameters exception	

Return Value	Description	Remark
1026	The email is registered	
1031	Unregistered email	
1032	Invalid email	
1041	No more attempts are allowed to get verification code	
1043	No more incorrect verification code attempts are allowed	
2000	The device does not exist	
2001	The camera does not exist	The camera is not registered to Ezviz Cloud. Check the camera network configuration
2003	The device is offline	Refer to Service Center Trouble Shooting Method
2004	Device exception	
2007	Incorrect device serial No.	
2009	The device request timeout	
2030	The device does not support Ezviz Cloud	Check whether the device support Ezviz Cloud. You can also contact our supports: 4007005998
5000	The device is added by yourself	
5001	The device is added by others	
5002	Incorrect device verification code	
7001	The invitation does not exist	

Return Value	Description	Remark
7002	Verifying the invitation failed	
7003	The invited user does not match	
7004	Canceling invitation failed	
7005	Deleting invitation failed	
7006	You cannot invite yourself	
7007	Duplicated invitation	You should call the interface for sharing or deleting the sharing. Troubleshooting: Clear all the sharing data in Ezviz Client and add the device again by calling related interface
10001	Parameters error	Parameter is empty or the format is incorrect
10002	accessToken exception or expired	The accessToken is valid for seven days. It is recommended that you can get the accessToken when the accessToken will be expired or Error Code 10002 appears
10004	The user does not exist	
10005	appKey exception	Return the error code when appKey is incorrect or appKey status is frozen
10006	The IP is limited	
10007	No more calling attempts are allowed	
10009	Signature parameters error	
10012	The third-party account is bound with the Ezviz account	
10013	The APP has no permission to call this interface	

Return Value	Description	Remark
10014	The APPKEY corresponding third-party userId is not bound with the phone	The appKey for getting AccessToken is different from the one set in SDK
10017	appKey does not exist	Fill in the App key applied in the official website
10018	AccessToken does not match with Appkey	Check whether the appKey for getting AccessToken is the same with the one set in SDK.
10019	Password error	
10020	The requesting method is required	
10029	The call frequency exceeds the upper-limit	
10030	appKey and appSecret mismatch.	
10031	The sub-account or the EZVIZ user has no permission	
10032	Sub-account not exist	
10034	Sub-account name already exist	
10035	Getting sub-account AccessToken error	
10036	The sub-account is frozen.	
20001	The channel does not exist	Check whether the camera is added again and the channel parameters are updated
20002	The device does not exist	①The device does not register to Ezviz. Check the network is connected. ②The device serial No. does not exist.

Return Value	Description	Remark
20003	Parameters exception and you need to upgrade the SDK version	
20004	Parameters exception and you need to upgrade the SDK version	
20005	You need to perform SDK security authentication	Security authentication is deleted
20006	Network exception	
20007	The device is offline	Refer to Service Center Check Method
20008	The device response timeout	The device response timeout. Check the network is connected and try again
20009	The device cannot be added to child account	
20010	The device verification code error	The verification code is on the device tag. It contains six upper-cases
20011	Adding device failed.	Check whether the network is connected.
20012	Adding the device failed.	
20013	The device has been added by other users.	
20014	Incorrect device serial No..	
20015	The device does not support the function.	
20016	The current device is formatting.	
20017	The device has been added by yourself.	

Return Value	Description	Remark
20018	The user does not have this device.	Check whether the device belongs to the user.
20019	The device does not support cloud storage service.	
20020	The device is online and is added by yourself.	
20021	The device is online and is not added by the user.	
20022	The device is online and is added by other users.	
20023	The device is offline and is not added by the user.	
20024	The device is offline and is added by the user.	
20025	Duplicated sharing.	Check whether the sharing exists in the account that added the device.
20026	The video does not exist in Video Gallery.	
20029	The device is offline and is added by yourself.	
20030	The user does not have the video in this video gallery.	
20031	The terminal binding enabled, and failed to verify device code.	Disable the terminal binding
20032	The channel does not exist for this user.	

Return Value	Description	Remark
20033	The video shared by yourself cannot be added to favorites.	
20101	Share the video to yourself.	
20102	No corresponding invitation information.	
20103	The friend already exists.	
20104	The friend does not exist.	
20105	The friend status error.	
20106	The corresponding group does not exist.	
20107	You cannot add yourself as friend.	
20108	The current user is not the friend of the added user.	
20109	The corresponding sharing does not exist.	
20110	The friend group does not belong to the current user.	
20111	The friend is not in the status of waiting verification.	
20112	Adding the user in application as friend failed.	
20201	Handling the alarm information failed.	
20202	Handling the leaved message failed.	

Return Value	Description	Remark
20301	The alarm message searched via UUID does not exist.	
20302	The picture searched via UUID does not exist.	
20303	The picture searched via FID does not exist.	
30001	The user doesn't exist	
49999	Data exception.	
50000	The server exception.	
60000	The device does not support PTZ control.	
60001	The user has no PTZ control permission.	
60002	The device PTZ has reached the top limit.	
60003	The device PTZ has reached the bottom limit.	
60004	The device PTZ has reached the left limit.	
60005	The device PTZ has reached the right limit.	
60006	PTZ control failed.	
60007	No more preset can be added.	
60008	The preset number of C6 has reached the limit. You cannot add more preset.	

Return Value	Description	Remark
60009	The preset is calling.	
60010	The preset is the current position.	
60011	The preset does not exist.	
60012	Unknown error.	
60013	The version is the latest one.	
60014	The device is upgrading.	
60015	The device is rebooting.	
60016	The encryption is disabled.	
60017	Capturing failed.	
60018	Upgrading device failed.	
60019	The encryption is enabled.	
60020	The command is not supported.	Check whether the device support the command.
60021	It is current arming/disarming status.	
60022	It is current status.	It is current open or closed status.
60023	Subscription failed.	
60024	Canceling subscription failed.	
60025	Setting people counting failed.	
60026	The device is in privacy mask status.	
60027	The device is mirroring.	

Return Value	Description	Remark
60028	The device is controlling PTZ.	
60029	The device is in two-way audio status.	
60030	No more incorrect card password attempts are allowed. Try again after 24 hours.	
60031	Card password information does not exist.	
60032	Incorrect card password status or the password is expired.	
60033	The card password is not for sale. You can only buy the corresponding device.	
60035	Buying cloud storage server failed.	
60040	The added devices are not in the same LAN with the parent device.	
60041	The added devices are not in the same LAN with the parent device.	
60042	Incorrect password for added device.	
60043	No more devices can be added.	
60044	Network connection for the added device timeout.	

Return Value	Description	Remark
60045	The added device IP conflicts with the one of other channel.	
60046	The added device IP conflicts with the one of parent device.	
60047	The stream type is not supported.	
60048	The bandwidth exceeds the system accessing bandwidth.	
60049	Invalid IP or port.	
60050	The added device is not supported. You should upgrade the device.	
60051	The added device is not supported.	
60052	Incorrect channel No. for added device.	
60053	The resolution of added device is not supported.	
60054	The account for added device is locked.	
60055	Getting stream for the added device error.	
60056	Deleting device failed.	
60057	The deleted device has no linkage.	Check whether there's linkage between IPC and NVR.

、API List

This part includes sub-accounts related API.

Following is the API list：

No	Function	Description
1	create sub-account	Create a sub-account in B (big account) mode
2	get single subaccount information	Get selected sub-acount information
3	get sub-account information list	Get sub-account information list in different page
4	edit sub-account password	Edit sub-account password
5	edit sub-account Permission strategy	Edit sub-account Permission strategy
6	add sub-account permission	Add sub-account statement in Permission strategy
7	delete sub-account permission	Delete one device’s all statement in sub-account
8	get sub-account AccessToken	Get sub-account AccessToken
9	delete sub-account	delete sub-account

- Common Return Code

Returned Code	Returned Infromation	Description
200	Operation succeed	Request succeed
10001	Parameter error	No parameter or wrong format
10013	Your application has no permission to call the API	
10002	accessToken error or expiration	Get accessToken again

Returned Code	Returned Infromation	Description
10005	appKey error	appKey locked
10031	The sub-account or the EZVIZ user has no permission	
10032	Sub-account not exist	
10034	Sub-account name already exist	
10035	Getting sub-account AccessToken error	
10036	The sub-account is frozen.	
50000	Operation failed	Operation failed

1.1Create Sub-Account

- Function

Create a sub-account in B (big account) mode.(sub-account functions description)

- Request Address

{areaDomain}/api/lapp/ram/account/create

- Request Method

POST

- Request Parameters

Parameters	Type	Description	Required
accessToken	String	The access token got from permission	Y
accountName	String	Sub-account name, 4-40 letters or characters	Y
password	String	Sub-account passwordLowerCase(MD5(AppKey#Passwords plaintext))	Y

LowerCase(MD5(AppKey#Passwords plaintext)):for AppKey use MD5 encryption to # and plaintext, and transfer to lower letters

- HTTP Request Message

POST /api/lapp/ram/account/create HTTP/1.1

Host: isgpopen.ezvizlife.com

Content-Type: application/x-www-form-urlencoded

accessToken=at.9307p5ye4yilog2f9apn82368j9g62g1-2rs1w3h0k0-092yx3m-ysd3i9cg9&accountName=test&password=5305b671da2d66785f7e6dd24c117370

- Return Data


```
{
  "data": {
    "accountId": "b3ad7ba927524b748e557572024d4ac2"
  },
  "code": "200",
  "msg": "Operation succeeded!"
}
```


- Return Filed：

Filed Name	Type	Description
accountId	String	Sub-account id

- Return Code

Common Return Code

1.2 Get Single Sub-Account Information

- Function:

This port is used for access selected sub-account information.(sub-account function discerption)

- Request Address

{areaDomain}/api/lapp/ram/account/get

- Request Method

POST

- Request Parameters

Parameter	Type	Description	Required
accessToken	String	The access token got from permission	Y
accountId	String	Sub-account id	N
accountName	String	Sub-account name	N

If the accessToken parameter is the same type of sub-account's AccessToken, the accountId and accountName's parameter can be none. If not, one of them must not be none, if both of them are not none, port return accountID sub-account information.

- HTTP Request Message

POST /api/lapp/ram/account/get HTTP/1.1

Host: isgpopen.ezvizlife.com

Content-Type: application/x-www-form-urlencoded

accessToken=at.9307p5ye4yilog2f9apn82368j9g62g1-2rs1w3h0k0-092yx3m-ysd3i9cg9&accountId=b3ad7ba927524b748e557572024d4ac2

- Return Data


```
{
  "data": {
    "accountId": "b3ad7ba927524b748e557572024d4ac2",
    "accountName": "test",
    "appKey": "ae1b9af9dcac4caeb88da6dbbf2dd8d5",
    "accountStatus": 1,
    "policy": {
      "Statement": [
        {
          "Permission": "GET,UPDATE,REAL",
          "Resource": [
            "dev:469631729",
```

```
            "dev:519928976",
            "cam:544229080:1"
          ]
        },
        {
          "Permission": "GET",
          "Resource": [
            "dev:470686804"
          ]
        }
      ]
    }
  },
  "code": "200",
  "msg": "Operation succeeded!"
}
```


- Return Filed:

Filed Name	type	description
accountId	String	Sub-account id
accountName	String	Sub-account name
appKey	String	Sub-account belonged app's AppKey
accountStatus	int	Sub-account status. 0 is off, 1 is on
policy	Policy	Sub-account permission strategy

Policy type(Policy grammar structure) :

Filed Name	Type	Description
Statement	Array[Statement]	Statement

Statement type :

Filed Name	Type	Description
Permission	String	Permission list
Resource	Array[String]	Resource list

- Return Code

Common Return Code

1.3 Access Sub-account Information List

- Function:

This port is used for get sub-account information in different pages(sub-account function description)

- Request Address

{areaDomain}/api/lapp/ram/account/list

- Request Method

POST

- Request Parameters

Parameter	Type	Description	Required
accessToken	String	accessToken got from permission	Y
pageStart	int	Page starts at 0	N
pageSize	int	Page size, default 10, max 50	N

- HTTP Request Message

POST /api/lapp/ram/account/get HTTP/1.1

Host: isgpopen.ezvizlife.com

Content-Type: application/x-www-form-urlencoded

accessToken=at.9307p5ye4yilog2f9apn82368j9g62g1-2rs1w3h0k0-092yx3m-ysd3i9cg9&pageStart=0&pageSize=2

- Return Data

{

"page": {

```
    "total": 15,
    "page": 0,
    "size": 2
  },
  "data": [
    {
      "accountId": "b3ad7ba927524b748e557572024d4ac2",
      "accountName": "test",
      "appKey": "ae1b9af9dcac4caeb88da6dbbf2dd8d5",
      "accountStatus": 1,
      "policy": {
        "Statement": [
          {
            "Permission": "GET,UPDATE,REAL",
            "Resource": [
              "dev:469631729",
              "dev:519928976",
              "cam:544229080:1"
            ]
          },
          {
            "Permission": "GET",
            "Resource": [
              "dev:470686804"
            ]
          }
        ]
      }
```

```
    },
    {
      "accountId": "0058a3964698415d8a70a931faa48d78",
      "accountName": "test2",
      "appKey": "ae1b9af9dcac4caeb88da6dbbf2dd8d5",
      "accountStatus": 1,
      "policy": null
    }
  ],
  "code": "200",
  "msg": "Operation succeeded!"
}
```


- Return Filed:

Filed Name	Type	Description
page	Page	Page information
data	Array[Account]	Sub-account information list
code	String	Sub-account information list
msg	String	Operation message

Page type

Filed Name	type	description
total	long	Total record count
page	long	Start page
size	long	Page size

Account type

Filed Name	Type	Description
accountId	String	Sub-account id

Filed Name	Type	Description
accountName	String	Sub-account name
appKey	String	Sub account apps' AppKey
accountStatus	int	Sub-account status, 0 is off, 1 is on
policy	Policy	Sub-account permission strategy

Policy type (Policy grammar structure)：

Filed Name	Type	Description
Statement	Array[Statement]	statement

Statement type：

Filed Name	Type	Description
Permission	String	Permission list
Resource	Array[String]	Resource list

- Return Code

Common Return Code

1.4 Edit Current Sub-account Password

- Function

This port is used for edit current sub-account password

- Request Address

{areaDomain}/api/lapp/ram/account/updatePassword

- Request Method

POST

- Request Parameters

Parameter	Type	Description	Required
accessToken	String	The accessToken got from permission	Y
accountId	String	Sub-account id	Y

Parameter	Type	Description	Required
oldPassword	String	Old password ,LowerCase(MD5(AppKey#plaintext))	Y
newPassword	String	New password LowerCase(MD5(AppKey#plaintext))	Y

LowerCase(MD5(AppKey#plaintext)):for AppKey use MD5 encryption to # and plaintext, and transfer to lower letters

- HTTP Request Message

POST /api/lapp/ram/account/updatePassword HTTP/1.1

Host: isgpopen.ezvizlife.com

Content-Type: application/x-www-form-urlencoded

accessToken=at.9307p5ye4yilog2f9apn82368j9g62g1-2rs1w3h0k0-092yx3m-ysd3i9cg9&accountId=b3ad7ba927524b748e557572024d4ac2&oldPassword=5305b671da2d66785f7e6dd24c117370&newPassword=cc03e747a6afbbcbf8be7668acfebee5

- Return Data


```
{
  "code": "200",
  "msg": "Operation succeeded!"
}
```


- Return Code

Common Return Code

1.5 Edit Sub-account Permission Strategy

- Function:

This port is used for edit mode B sub-account permission strategy

- Request Address

{areaDomain}/api/lapp/ram/policy/set

- Request Method

POST

- Request Parameters

Parameter	Type	Description	Required
accessToken	String	The accessToken got from permission	Y
accountId	String	Sub-account Id	Y
policy	String	Permission strategy,Policy grammar sentence structure	Y

- HTTP Request Message

POST /api/lapp/ram/policy/set HTTP/1.1

Host: isgpopen.ezvizlife.com

Content-Type: application/x-www-form-urlencoded

accessToken=at.9307p5ye4yilog2f9apn82368j9g62g1-2rs1w3h0k0-092yx3m-ysd3i9cg9&policy=%7B%22Statement%22%3A%5B%7B%22Permission%22%3A+%22GET%2CUPDATE%2CREAL%22%2C%22Resource%22%3A%5B%22dev%3A469631729%22%2C%22dev%3A519928976%22%2C%22cam%3A544229080%3A1%22%5D%7D%5D%7D&accountId=b3ad7ba927524b748e557572024d4ac2

- Return Data

```
{
  "code": "200",
  "msg": "Operation succeed!"
}
```

- Return Code

Common Return Code

1.6 Add Sub-account Permission

- Function:

This port is used for add sub-account statement in permission strategy

- Request Address

{areaDomain}/api/lapp/ram/statement/add

- Request Method

POST

- Request Parameters

Parameter	Type	Description	Required
accessToken	String	The accessToken from permission	Y
accountId	String	Sub-account Id	Y
statement	String	Statement,Statement grammar structure,For example:{"Permission": "GET", "Resource": ["dev:469631729"]}	Y

- HTTP Request Message

POST /api/lapp/ram/statement/add HTTP/1.1

Host: isgpopen.ezvizlife.com

Content-Type: application/x-www-form-urlencoded

accessToken=at.9307p5ye4yilog2f9apn82368j9g62g1-2rs1w3h0k0-092yx3m-ysd3i9cg9&statement==%7B%22Permission%22%3A+%22GET%22%2C+%22Resource%22%3A+%5B%22dev%3A547596317%22%5D%7D&accountId=b3ad7ba927524b748e557572024d4ac2

- Return Data

```
{
  "code": "200",
  "msg": "Operation succeed!"
}
```

- Return Code

Common Return Code

1.7 Delete Sub-Account Permission

- Function

This port is used for delete one device's all statement in a sub-account

- Request Address

{areaDomain}/api/lapp/ram/statement/delete

- Request Method

POST

- Request Parameters

Parameter	Type	Description	Required
accessToken	String	The accessToken got from permission	Y
accountId	String	Sub-account Id	Y
deviceSerial	String	Device serial number	Y

- HTTP Request Message

POST /api/lapp/ram/statement/delete HTTP/1.1

Host: isgpopen.ezvizlife.com

Content-Type: application/x-www-form-urlencoded

accessToken=at.9307p5ye4yilog2f9apn82368j9g62g1-2rs1w3h0k0-092yx3m-ysd3i9cg9&deviceSerial==547596317&accountId=b3ad7ba927524b748e557572024d4ac2

- Return Data


```
{
  "code": "200",
  "msg": "Operation succeeded!"
}
```


- Return Code

Common Return Code

1.8 Access Mode B Sub-Account AccessToken

- Function

This port is used for access mode B sub-account accessToken

- Request Address

{areaDomain}/api/lapp/ram/token/get

- Request Method

POST

- Request Parameters

Parameter	Type	Description	Required
accessToken	String	The accessToken got from permission	Y
accountId	String	sub-account Id	Y

- HTTP Request Message

POST /api/lapp/ram/token/get HTTP/1.1

Host: isgpopen.ezvizlife.com

Content-Type: application/x-www-form-urlencoded

accessToken=at.9307p5ye4yilog2f9apn82368j9g62g1-2rs1w3h0k0-092yx3m-ysd3i9cg9&accountId=b3ad7ba927524b748e557572024d4ac2

- Return Data


```
{
  "data": {
    "accessToken": "ra.7jrcjmna8qnqg8d3dgnzs87m4v2dme3l-32enpqgusd-1jvdfe4-uxo15ik0s",
    "expireTime": 1470810222045,
    "areaDomain": "https://iusopen.ezvizlife.com"
  },
  "code": "200",
  "msg": "Operation succeed !"
}
```


- Return Code

Common Return Code

1.9 Delete Sub-Account

- Function

This port is used for delete sub-account

- Request Address

{areaDomain}/api/lapp/ram/account/delete

- Request Method

POST

- Request Parameters

Parameter	Type	Description	Required
accessToken	String	The accessToken got from permission	Y
accountId	String	sub-account Id	Y

- HTTP Request Message

POST /api/lapp/ram/account/delete HTTP/1.1

Host: isgpopen.ezvizlife.com

Content-Type: application/x-www-form-urlencoded

accessToken=at.9307p5ye4yilog2f9apn82368j9g62g1-2rs1w3h0k0-092yx3m-ysd3i9cg9&accountId=b3ad7ba927524b748e557572024d4ac2

- Return Data


```
{
  "code": "200",
  "msg": "Operation succeed !"
}
```

Function Description For The Sub Account Of B (Big Account)

Opening the function of subaccount of platform B (large account) the main is aimed at controlling the B (large account) mode developer account permissions. With this function, the sub account can be created within the permissions of developer account, and different permission is assigned to the corresponding subaccount

Why use sub account

One of the core issues that a sub account needs to address is how to safely authorize other people to access devices without exposing the AccessToken of the B (big account) model developer. Because once the developer's AccessToken is exposed, there is a security risk, and others can have access to all of the device resources of the developer.

However, the sub account function is a long-term effective mechanism of access control, and by separating sub account with different permissions, different permissions will be assigned to different users. In this case, once the sub account leak ,it will not cause global information disclosure.

Basic concepts

The following is a brief explanation of some basic concepts: sub account: It is a sub account created from developer accounts with the EZVIZ open platform. when creating sub account ,the developer can set different permission and passwords .when created, each sub account has a ID.

- Authorization policy: a rule used to define permissions, such as previewing a device.
- Resource: a resource that a user can access, such as a device with a serial number of 469631729, or a first channel of a device with a serial number of 469631729.

Application scenarios

If you buy some EZVIZ devises or other Hikvision devises with EZVIZ protocol and all the devises are used with the account of developer ,the users who have an access to these devises will be exposed to using the AccessToken created from the developer account Here's the problem: you can't control the privileges of the devices that a particular user can access.

Here is the solution: rights control can be achieved through the function of sub account. The developer account is called the primary account while account

recreated is called a sub account. The sub account can only use the operations and equipment resources authorized by the master account.

Application scenarios of sub account

the scene of educational institution video surveillance system is taken for example. a kindergarten has ten classrooms, and each classroom has 2 EZVIZ cameras, while the kindergarten administrator has these 20 EZVIZ equipment ownership. An APP has been developed and integrated OpenSDK in a The kindergarten, and with the help of this APP, these parents will be able to see video in the class. Once the parents register on this APP ,the kindergarten administrators assign corresponding classroom permissions to the parents to see own their children ,and PTZ control, video encryption can not be operated.

Details of the scheme are described below ：

1. User register on this APP . App users are terminal ones ,and they are user of developer. And they have nothing to do with EZVIZ cloud or its platform account.
2. Create sub accounts. For every valid App users, App Server can define each App user access permissions (described by the Policy grammar); for every App user has access to specify a sub account, if there is no, it need to create a specified access sub accounts. correspondence between App users and sub accounts will be saved in App server.

The sub account can be created with the open platform of EZVIZ via an interface, and terminal user are not aware of the sub account .

In this scenario, a sub account of "" parents in classroom A " will be created and authorized to check theinformation of devices and to live view and playback. However, the sub account just have the access to C6(519928976)、C2C(470686804)

```
{
  "Statement": [
    {
      "Permission": "Get,Real,Replay",
      "Resource": [
        "dev:519928976",
        "dev:470686804"
```

```
        ]
    }
  ]
}
```


3. log in with account
4. The App Server recognize the corresponding sub account of this App user,and request the EZVIZ cloud to offer the platform and gain a Access Token in class A
5. the opening platform of EZVIZ give back a valid voucher with expiration time and Access Token to App Server.
6. App Server returns the access credentials to the Client App. while Client App can buffer this Access Token. When Access Token fails, Client App needs to apply for a new Access Token from App Server.
7. Client App uses Access Token to request EZVIZ opening platform API (including the Open SDK interface and the Http interface). EZVIZ opening platform verifies that the sub account has access to the device in accordance with the Access Token in the request parameter, and only the requests that meet the minimum permissions required by the API interface are allowed. If the user queries the device list, the cloud platform will only return the device information gained by sub account with corresponding access

In this scenario, the user will check device list, and cloud platform only return back the information of equipment C6 (519928976), C2C (470686804); if other equipment or this 2 equipment is operated with the PTZ control, video encryption, cloud platform will return to the error code “no access”

Sub account authorization policy (Policy) configuration

There is a Statement in Policy (one can have multiple Statement in Policy). And the Statement specifies the corresponding Permission, Resource.
The following is an example of an authorization policy (Policy):


```
{
  "Statement": [
    {
      "Permission": "Get,Update,DevCtrl",
```

```
      "Resource": [
        "dev:469631729",
        "cam:544229080:1"
      ]
    },
    {
      "Permission": "Get,Real",
      "Resource": [
        "dev:470686804"
      ]
    }
  ]
}
```


The Policy licensed 469631729, 544229080, 470686804 and other equipment resources under the developer EZVIZ account, and it supported Get, Update, DevCtrl, and Real respectively.

Configuration details

Statement

“Statement” describes the authorization semantics, and each contains descriptions of Permission and Resource. Each request system will check matches one by one, and if the matches are successful, the request can pass the authentication. If none matches successfully, the request will be denied .

Permission

Includes the following permissions:

Permission	Resource type	Description
Update	dev、cam	Modify resources, such as modifying device names and channel names
Get	dev、cam	Check resource information, including configuration information, etc.

Permission	Resource type	Description
DevCtrl	dev、cam	Complete control of the equipment, including Real, Replay, Alarm, Capture, Video, Ptz, Upgrade, Format, Pipe, Config and other defined device operating rights and defined device operating rights later
Real	dev、cam	live view
Replay	dev、cam	Video playback (including local video playback and cloud storage playback)
Alarm	dev	Get the device alarm information and subscribe to the alert message
Capture	dev、cam	Capture
Video	dev、cam	Recording
Ptz	dev、cam	PTZ control
Upgrade	dev	Upgrading
Format	dev	Format the disk
Pipe	dev	Using open platform transparent channel function
Config	dev、cam	Configurate the device, such as video encryption, disarm association between NVR and IPC

Comment: in this version, the sub-account does not have add/delete device permission; sub-account can send request of add/delete device to developer, developer add/delete device.

Resource

Resource usually means the operation object, like device, channel, we use the following format to name the resource, {resourceType}:{resourceId}；

- resourceType: the type of resource, there are only two types, dev(device), cam(channel)；
- resourceId：the ID of resource, resourceId of dev type is serial number, resourceId of cam is serial number: channel number.

Subordinate relations

Cam resource belongs to dev resource, the sub-account's permission to dev resource can pass to cam resource. For instant, there are two resources, dev: 544229080 and cam: 544229080:1, cam: 544229080:1 belongs to dev: 544229080; If the developer give the real permission of dev: 544229080 resource to sub-account, then this sub-account has the real permission for cam: 544229080:1.

Policy Structure Introduction

Policy structure includes authorization sentence list. Every authorization sentence includes permission and resource.

Support JSON format description

For now, it only supports JSON format description. When you create or update policy, the cloud check if JSON format is correct or not. About JSON grammar, please see RFC 7159. Users can also use some online JSON format checker or editor to test the JSON text's validity

Policy grammar

Grammar description symbol instruction :

1. the JSON symbol of Policy :

{ } [ ] " , :

2. special symbol of grammar:

= < > ( ) |

3. when an element has multiple values, use comma and ellipsis:

[<resource_string>, <resource_string>, ...]

4. the elements between double quotation marks is string :

<permission_block> = "Permission" : "<permission_string>, <permission_string>, ..."

grammar description :

policy = {

<statement_block>

}

<statement_block> = "Statement" : [<statement>, <statement>, ...]

<statement> = {

<permission_block>,

<resource_block>

}

<permission_block> = "Permission" : "<permission_string>, <permission_string>, ..."

<resource_block> = "Resource" : [<resource_string>, <resource_string>, ...]

Policy grammar description：

- A policy can has many statement。
- in one statement, permission is a string support multiple authorizations, resource is a list which support multiple objects。

Permission

permission supports multiple values, the values must be accepted at ezviz platform：

Permission	**Resource type**	**description**
Update	dev、cam	Edit resource, for example, edit device's name, channel name
Get	dev、cam	Search resource information, includes configuration
DevCtrl	dev、cam	Device control all, includes, Real、Replay、Alarm、Capture、Video、Ptz、Upgrade、Format、Pipe、Config and other defined device operation permissions and the device operation permissions which will be defined in the future.
Real	dev、cam	live view
Replay	dev、cam	Play back (includes local, and cloud)
Alarm	dev	Access device alarm information, subscribe alarm information
Capture	dev、cam	capture

Permission	Resource type	description
Video	dev、cam	recording
Ptz	dev、cam	cloud control
Upgrade	dev	upgrade
Format	dev	Format the disk
Pipe	dev	Using EZVIZ transparent channel function
Config	dev、cam	Configure device, like encrypt video, disarm, link NVR and IPC etc.

Comment: in this version, the sub-account does not have add/delete device permission; sub-account can send request of add/delete device to developer, developer add/delete device.

Example： "Permission": "Get,Update,DevCtrl"

Resource

Resource usually means the operation object, like device, channel, we use the following format to name the resource {resourceType}:{resourceId}

Format Description:

- • resourceType: the type of resource, there are only two types, dev(device), cam(channel);
- • resourceId: the ID of resource, resourceId of dev type is serial number, resourceId of cam is serial number: channel number.

Example： "Resource": ["dev:469631729","cam:544229080:1"].

Policy example

The following Policy includes two statements: first statement allows device 469631729 and 544229080:1 channels get resource information, Update, Real, Replay permissions. The second statement allows to get device 470686804 information and real view.

```
{
  "Statement": [
    {
```

```
        "Permission": "Get,Update,Real,Replay",
        "Resource": [
          "dev:469631729",
          "cam:544229080:1"
        ]
      },
      {
        "Permission": "Get,Real",
        "Resource": [
          "dev:470686804"
        ]
      }
    ]
}
```