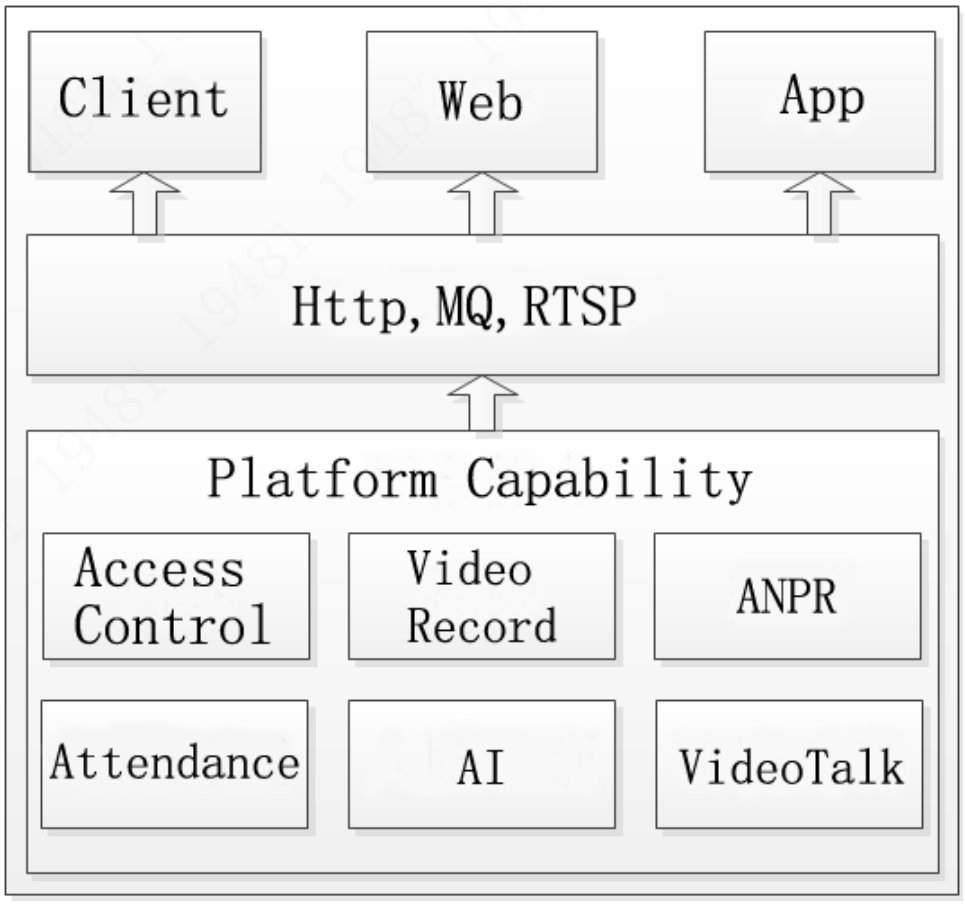
Catalog

1、Development Guide

1.1、General Introduction

Overall structure

The platform provides a function set including authentication, device tree, video, access, transportation, attendance management, face recognition and visible intercom business, and interacts with HTTP, MQ and RTSP protocols. HTTP protocol is used for accessing REST interface provided on platform. Operation control and platform resource access can be required by REST interface. MQ protocol is used for subscribing and receiving message on the platform. RTSP protocol is used for accessing audio and video data on the platform. Overall structure chart:



Development preparation

- Get platform installation package and install the platform according to the manual. After successful installation, configure an account and import License file.
- Add front-end device to the platform to ensure there are accessible front-end coding and decoding resources on the platform.
- Select HTTP and HTTPS: The platform provides HTTP (plaintext) and HTTPS (encrypted) access modes. Default access ports of HTTP and HTTPS are port 80 and port 443 respectively. HTTP is recommended at the development and commissioning stage to position question. HTTPS is recommended in actual operation to ensure safe access to the platform.
- Select proper development language to realize interaction with HTTP, MQ and RTSP protocols. Take player component into consideration to realize function of video playing.

Basic concept

- Session: A session is generated after a client is authenticated on the platform. In the session, the client could request permitted resources and receive MQ push messages on the platform. The platform deletes the session in case of keep-alive timeout or authentication being canceled initiatively.
- Department: The platform has a root department. Departments and devices can be mounted under the root department. Departments and devices also can be carried under an department. The department structure can be defined according the actual hierarchical relation on the site.
- Device: A real physical device. The platform supports encoder, decoder and other devices of Intelbras. The platform gets audio and video resources, alarm detection and other resources via access service and streaming media service.
- Unit: Divide a device into multiple units according to capability provided. Common unit types include coding unit, decoding unit, alarm input unit and alarm output unit. A device includes multiple units.
- Channel: There are multiple channels under a unit. A channel is a smallest unit to provide data. For example, a coding channel can output video streaming data and an alarm channel can detect and output alarm information. A unit has multiple channels, while the channel type and the unit type are consistent.

1.2、Development Guide

Authentication keep-alive

Before accessing platform resources, call the authentication interface (/admin/API/accounts/authorize) to get the authentication of the platform. After successful authentication, save the returned token value and userId. Use the token and userId for subsequent MQ subscription and RTSP interface calling. When calling other REST interfaces, put token value in the X-Subject-Token field at HTTP head, otherwise verification might fail, causing invocation interface return failed. Regularly keep alive for calling keep-alive interfaces (/admin/API/accounts/updateToken). If no keep-alive actions take place within a period, the platform detects keep-alive timeout and the corresponding token value becomes invalid.

MQ notice

Get IP and port connecting MQ by getting MQ configuration (admin/API/BRM/Config/GetMqConfig) interface. Connect MQ service via activeMQ. At present, there are 3 topics to be subscribed on the platform:

Theme	Subscription format
Alarm topic	mq.alarm.msg.topic.{UserId}
Public topic	mq.common.msg.topic
Event topic	mq.event.msg.topic.{UserId}

UserId is authentication interface return. If UserId is 1, subscribe this topic:

Theme	Subscription format
Alarm topic	mq.alarm.msg.topic.1
Public topic	mq.common.msg.topic
Event topic	mq.event.msg.topic.1

Message pushed by MQ is in the format of Json. Refer to MQ message for detail format.

Department tree

Call device department tree interface (/admin/API/tree/deviceOrg) and device list interface (/admin/API/tree/devices) to get department structure and device information. Every device and channel has a unique code by which data could be assembled and a device tree generated. Subsequent operations on device and channel are achieved by requesting to the platform via device code and channel code on the device tree.

Real-time video

Use code channel on the device tree to request real-time video. Get stream data by RTSP protocol after getting the real-time video URL. The platform supports private stream and standard stream. Request private stream and standard stream by assembling different RTSP formats. Intelbras player can play private stream. You can access Intelbras player from Intelbras official website. It adopts C interface and the dynamic library (dll) for loading. Intelbras player also can play standard stream. Or you can select a open source library to decode and play it.

Record playback

Platform record: Configure platform storage. The device can be offline for platform record Search and playback. Device record: Device record should be stored in device. The device must be online for device record search and playback. The platform supports private stream and standard stream. Request private stream and standard stream by assembling different RTSP formats. Intelbras player can play private stream. You can access Intelbras player from Intelbras official website. It adopts C interface and the dynamic library (dll) for loading. Intelbras player also can play standard stream. Or you can select a open source library to decode and play it.

1.3、FAQ

Interface returns 401

Token value of X-Subject-Toekn at http head of request message is invalid. Interface returns 401 error code. Packet capture shown as below:

```
POST /admin/API/MTS/Video/StartVideo HTTP/1.1
Content-type: application/json
X-Subject-Token:
Content-Length: 256
Host: 10.35.93.19:80
Connection: Keep-Alive
User-Agent: Apache-HttpClient/4.5.3 (Java/1.8.0_65)
Accept-Encoding: gzip,deflate
```

{"clientType":"WINPC","clientMac":"","clientPushId":"","project":"PSDK","method":"MTS.Video.StartVideo","data":{"streamType":"1","optional":"/admin/API/MTS/Video/StartVideo?","trackId":"","extend":"","channelId":"1000001$1$0$0","planId":"","dataType":"2"}}

```
HTTP/1.1 401 Unauthorized
Server: Apache-Coyote/1.1
Set-Cookie: JSESSIONID=CAF98CE25740B7867A4FFC9078CD10FA; Path=/admin; HttpOnly
Content-Type: application/json;charset=utf-8
Content-Length: 66
Date: Wed, 23 Jan 2019 03:36:22 GMT
```

{"realm":"DSS","randomKey":"d14a801f91594873","encryptType":"MD5"}

Interface returns 415

Content-type field at http head is not set as application/json. Interface returns 415 error code. Packet capture as shown below:

```
POST /admin/API/MTS/Video/StartVideo HTTP/1.1
X-Subject-Token: 0574987f9cf54f5a8a41a8ae22ae8e57
Content-Length: 288
Content-Type: text/plain; charset=ISO-8859-1
Host: 10.35.93.19:80
Connection: Keep-Alive
User-Agent: Apache-HttpClient/4.5.3 (Java/1.8.0_65)
Accept-Encoding: gzip,deflate
```

{"clientType":"WINPC","clientMac":"","clientPushId":"","project":"PSDK","method":"MTS.Video.StartVideo","data":{"streamType":"1","optional":"/admin/API/MTS/Video/StartVideo?0574987f9cf54f5a8a41a8ae22ae8e57","trackId":"","extend":"","channelId":"1000001$1$0$0","planId":"","dataType":"2"}}

```
HTTP/1.1 415 Unsupported Media Type
Server: Apache-Coyote/1.1
Set-Cookie: JSESSIONID=3FEB4F516B136F900B7A8CED64B81484; Path=/admin; HttpOnly
Content-Type: text/html;charset=utf-8
Content-Language: en
Content-Length: 1048
Date: Wed, 23 Jan 2019 03:46:04 GMT
```

HTTP Status 415 -

---

type Status report

message

description The server refused this request because the request entity is in a format not supported by the requested resource for the requested method.

Apache Tomcat/7.0.85

1.4、Error code

Error code	Error Message
409	No permission
413	Request entity too large
500	Internal error
520	Server error
1004	The parameter is illegal.
1005	The name already exist.
1006	more than the max limit num.
1007	the data not exist
1008	the task was not finish
1101	The nonce is illegal.
1103	The user is not authorized.
1104	No permission to access.
2001	Incorrect username or password.
2002	The user has been locked
2003	The user has been disabled
2004	The user has logged in.

Error code	Error Message
2005	The user is forbidden to login at the current date
2006	The user is forbidden to login at the current time
2007	The file name can not be null
2008	The old password can not be null
2009	The new password can not be null
2010	The user does not exist
2011	The password can not be same with default password
2012	The subject can not be null
2013	The user(s) can not be null
2014	The video(s) can not be null
2015	The device code can not be null
2016	The device does not exist
2017	The channel(s) can not be null
2018	The channel id is not valid
2019	The tv wall id must be a unsigned long integer
2020	The task id must be a unsigned long integer
2021	The task name can not be null
2022	The task does not exist
2023	The tv wall does not exist
2024	The tag id must be a unsigned long integer
2025	The record source must be a unsigned integer
2026	The subject can not be null
2027	The tag time must be a unsigned long integer
2028	The start time must be a unsigned long integer
2029	The end time must be a unsigned long integer
2030	The cruise id must be a unsigned long integer
2031	The cruise data can not be null
2032	The cruise does not exist
2033	The caller can not be null
2034	The callee can not be null
2035	The duration must be a unsigned long integer
2036	The status must be a unsigned integer
2037	The call time must be a unsigned long integer
2038	The operate type must be a unsigned integer
2039	The operate time must be a unsigned long integer
2040	The service type must be a unsigned integer
2041	The device or service does not exist
2042	The talk type must be a unsigned integer
2043	The talk type does not support
2044	The channel seq must be a unsigned integer
2045	The broadcast channels can not be null

Error code	Error Message
2046	The audio type must be a unsigned integer
2047	The talk mode must be a unsigned integer
2048	The audio bit must be a unsigned integer
2049	The sample rate must be a unsigned integer
2050	The record type must be a unsigned integer
2051	The audio session can not be null
2052	The device is at talking
2053	You are requesting talking, please wait for last response
2054	The data type must be a unsigned integer
2055	The data type does not support
2056	The stream type must be a unsigned integer
2057	The stream type does not support
2058	The disk id can not be null
2059	The stream id must be a unsigned long integer
2060	The stream id does not exist
2061	The alarm id must be a unsigned long integer
2062	The alarm status must be a unsigned integer
2063	The record month can not be null
2064	The record duration must be a unsigned long integer
2065	The alarm code can not be null
2066	The record source does not support
2067	The record type does not support
2068	The alarm does not exist
2069	The user id must be a unsigned long integer
2070	The channel(s) can not be null
2071	The id(s) can not be null
2072	The id must be a unsigned long integer
2073	The record lock info does not exist
2074	The operate type does not support
2075	The lock time must be a unsigned long integer
2076	The command must be a unsigned integer
2077	The direct must be a unsigned integer
2078	The point x must be a integer
2079	The point y must be a integer
2080	The point z must be a integer
2081	The step x must be a unsigned integer
2082	The step y must be a unsigned integer
2083	The handle user can not be null
2084	The user has no authorization.
2085	The PTZ zoom must be a float
2086	The PTZ focus must be a float

Error code	Error Message
2087	The point code can not be null
2088	The point name can not be null
2089	The value of page info can not be null
2090	The value of page size can not be null
2091	The value of order info can not be null
2092	The alarm status must be a unsigned integer
2093	The alarm type must be a unsigned integer
2094	The alarm grade must be a unsigned integer
2095	The handle status must be a unsigned integer
2096	The alarm start time must be a unsigned long integer
2097	The alarm end time must be a unsigned long integer
2098	The handle start time must be a unsigned long integer
2099	The handle end time must be a unsigned long integer
2100	The page number must be a unsigned integer
2101	The page size must be a unsigned integer
2102	The order type must be a unsigned integer
2103	The handle status can not be null
2104	The matrix id can not be null
2105	The operate type can not be null
2106	The tv index can not be null
2107	The tv index must be a unsigned integer
2108	The screen index can not be null
2109	The screen index must be a unsigned integer
2110	The subTv index can not be null
2111	The subTv index must be a unsigned integer
2112	The split num can not be null
2113	The split num must be a unsigned integer
2114	The tvWall id can not be null
2115	The tvWall id must be a unsigned integer
2116	The value of position info can not be null
2117	The left can not be null
2118	The top can not be null
2119	The width can not be null
2120	The height can not be null
2121	The tv type can not be null
2122	The tv type must be a unsigned integer
2123	The alarm level can not be null
2124	The alarm level must be a unsigned integer
2125	The order can not be null
2126	The order must be a unsigned integer
2127	The child window can not be null

Error code	Error Message
2128	The child window must be a integer
2129	The email is not valid
2130	The command does not support
2131	The alarm process does not exist
2132	The user role does not exist
2133	The user does not have permissions of the channel
2134	The user does not have permissions of the device
2135	The mail recipient can not be null
2136	The mail message can not be null
2137	The number of messages has reached an upper limit
2138	The target must be a unsigned integer
2139	The alarm source can not be null
2140	The duration must be a positive integer
2141	The azimuth must be a digit
2142	The data type can not be null
2143	The data name can not be null
2144	The data does not exist
2145	The window info does not exist
2146	The screen type can not be null
2147	The screen type must be a unsigned integer
2148	The left must be a unsigned float
2149	The top must be a unsigned float
2150	The width must be a unsigned float
2151	The height must be a unsigned float
2152	The window id can not be null
2153	The window id must be a unsigned Integer
2154	The input data can not be null
2155	The mac address is forbidden
2156	The domain authentication center is forbidden
2157	The value of handle message data too long
2158	The session can not be null.
2159	The type must be a unsigned integer.
2160	The type does not support.
2161	The ivs face device is offline.
2162	The image data can not be null.
2163	The person name can not be null.
2164	The person type must be a positive integer.
2165	The person gender must be a positive integer.
2166	The ss id must not be null.
2168	Only default password
2169	The system undo init.

Error code	Error Message
2170	The system finished init.
2171	The repository more than maximum number x.
2172	The person more than a repository maximum number x.
2173	The person more than maximum number x.
2174	The person type more than maximum number x.
2175	The person type already exists.
2176	The person type has been used, can not be removed.
2177	There are some person types has been used, can not be removed.
2178	Get current task failed from VMS
2179	The repository not enabled.
2180	The person image size more than maximum limit.
2181	The person image size more than minimum limit.
2182	The request info is null.
2183	The face repository search xo is null.
2184	The face repository id is null or zero.
2185	The face repository xo is null.
2186	The face repository name is empty.
2187	The face repository color is empty.
2188	The face person search xo is null.
2189	The face person id is null or zero.
2190	The face person xo is null.
2191	The face person name is empty
2192	The face person gender is empty
2193	The face person type is empty.
2194	The face person type less than zero or equals to zero
2195	The face image data is empty
2196	The face person type xo is null.
2197	The face person type name is empty.
2198	The face person type id is empty.
2199	The face application data is empty.
2200	The face application search xo is null.
2201	The face control xo is null.
2202	The face control channel info is null.
2203	The face control channel info of id is error.
2204	The face control channel info of similarity is error.
2205	The face repository name already exists.
2206	Mail service has not been opened
2207	The face repository not exist.
2208	The face excel not exist.
2212	The file format must be zip, rar, 7z .
2213	The repository has been used, can not be removed.

Error code	Error Message
2214	There are some repositories has been used, can not be removed.
2215	The face device not support.
2216	The face person id already exists.
2217	The person more than maximum number x.
2218	The user has been locked, because of system update.
2219	The domain ip and port and login user repeat.
2220	The domain name repeat.
2221	The domain is not exist.
2222	The domain can not found.
2223	The domain not enabled.
2224	The face repository is being operated, please try again later.
2225	The domain more than max number.
2226	The domain more than license limit
2227	The user does not have permissions of the slave channel
2228	The master-slave track config is empty
2229	Device delete face repository failed.
2230	the slave channel not exist.
2231	device not support master slave track config.
2232	The page mode must be a unsigned integer.
2233	The last id must be a unsigned integer.
2234	The lens type must be a unsigned integer.
2235	The system is busy. Please try again later.
2236	The face detection interrupt.
3000	The communication has an error or is timeout
3001	The MTS server is not ready.
3002	The SS server is not ready.
3003	The DMS server is not ready.
3004	The VMS server is not ready.
3005	The PTS server is not ready.
3028	other device is upgrade now!
3029	the device is offline
3030	the device is not door, unable to upgrade
3031	the screen have plan,unable to change
3032	illegal char
3099	The Domain server is not ready.
3100	The service work has an error or is false
3101	The service data has an error or is false
3111	The face server is not ready.
3112	The communication not has response.
3113	The communication has an error.
3201	Record Plan is running

Error code	Error Message
3202	Not match audio type.
3203	Not found available disk
3204	Not found capability
3206	The userType not user clientType.
3207	The search user status must be in -1, 0, 1.
5000	The item type can not be null.
6000	Parameter should be unsignedInteger
6001	Parameter can not be null
6002	UserName can not be null
6003	Password can not be null
6004	User name exist
6005	role name exist
6006	Timetemplate name exist
6007	TimeTemplateName null
6008	new password can not be null
6009	old password can not be null
6010	the randomKey's cache is out of time or removed
6011	the randomKey can not be null
6012	LanPath can not be null
6013	backup file does not exists
6014	the new password can not be same with original password
6015	restore file path can not be null
6016	restore file does not exists
6017	email address repeats
6018	can not delete or update system user
6019	can not delete or update user,do not have privilege
6020	export file must be xls or csv or txt
6021	current user can not update another user
6022	user ids can not be null
6023	alarm source channel id can not be null
6024	the answers's length must be three.
6025	The encrypt password can not be bull
6026	The auto backup type can not be bull
6027	The auto backup date can not be bull
6028	The auto backup time can not be bull
6029	The decrypt password can not be bull
6030	The manual restore file content not be bull
6031	The alarm keeping info should be unsignedInteger
6032	The log keeping info should be unsignedInteger
6033	The heatmap keeping info should be unsignedInteger
6034	The timesync starting time can not be null

Error code	Error Message
6035	The send email address is illegal
6036	The recipient email address is illegal
6037	The current timeTemplate is occupied by others
6038	The source channel id can not be null.
6039	The target channel id can not be null.
6040	The database does not exist from channel info
6041	Can not delete online user
6044	current user can not operate system service!
6045	Device ip and port repeat
6046	Device not exists
6047	Device password is incorrect
6048	Device login name is incorrect
6049	Device login timeout
6050	Account already login
6051	Account locked
6052	Account are blacklisted
6053	Resource insufficient
6054	Child connection fail
6055	Main connection fail
6056	Connections exceed
6057	The device password or login name is incorrect
6058	RecordScheduleName null!
6059	RecordScheduleName repeats!
6060	Delete iscsi failed,Check that server task!
6071	Disk group has configured group quota, cannot set picture type!
6072	The IP address is illegal!
6073	The port number is illegal!
6074	The port number has been occupied!
6075	The virtual disk capacity should be between 10 and 16384!
6076	The orgName must not be null
6077	The organization exsit
6078	The organization orgCode is NULL
6079	can not be deleted
6080	The devicegroup forbidden delete
6083	Organization has been changed, please refresh the page.
6089	Incorrect user password
6091	Device discovery cache expired.
6092	Unknown error.
6093	Service type not exists.
6094	Service not register.
6096	Service bad request.

Error code	Error Message
6097	Can not connect to P2P server.
6098	P2P server response empty, device may be not register to p2p server.
6099	failed to invoke obms.
6101	Import file parse error
6102	Import file data is illegal
6103	Building enable closed is not support
6106	The device is a member of a device group or management group and is not allowed to delete
6107	The user is a member of a intercom management group and is not allowed to delete
6108	The number of encoder channels exceeds license limit
6109	The number of access control exceeds license limit
6110	The number of access control channels exceeds license limit
6111	The number of VTO exceeds license limit
6112	The number of SIP ID exceeds license limit
6113	The number of alarm host exceeds license limit
6114	The number of defence area exceeds license limit
6115	The number of entrance and exit exceeds license limit
6116	The number of face detect exceeds license limit
6117	The number of Client connections exceeds license limit
6120	Failed to fetch device channel info from SOSO
6121	Door is in use
6123	the user has not the module rights
6124	This record was not found
6126	The alarm source code has a alarm plan
6133	The license has expired and you cannot login
6135	The backup file does not match the current DSS version and cannot be restored
6136	Device type is illegal
6138	The number of Onvif channels exceeds license limit
6139	Disallow building disabled while unit is enabled
6140	The number of attendance terminals exceeds license limit
6141	The device channel has been distributed to the repository
6146	The number of registered VDP APP users exceeds license limit
6153	The residence scenario is not configured
6155	The residence scenario is not supported by platform
6156	The window has been configured link video wall and cannot be delete
6159	The task of extract acs record is running, please try later.
6160	The number of P2P channels exceeds license limit.
7000	Authentication failed
7001	The card person's id can not be null
7002	The card person's id is exists
7003	The anti-passback'name is exists

Error code	Error Message
7004	The inter-lock'name is exists
7006	The door group does not exist
7007	The department has persons,can not be delete
7008	The department is exists
7009	The department has child node,can not be delete
7010	The door group has rights, can not be delete
8000	Device group name has already existed
8001	Manager group name has already existed
8017	AccessControl support one rule, AccessCentralControl support three rule
8018	The person has first card info
8019	The person has anti-passback info
8023	The person group has multicard info
8024	This device group is a member of a relation group and is not allowed to delete
8025	This management group is a member of a relation group and is not allowed to delete
8042	The card num is exist
8043	The person has person group info
8044	The inputing parameter is illegal
8045	The person has card rights,must have card number
8046	The parameter must be integer
8049	Import person info is null
8050	Parameter is null
8053	The shift name already exists.
8056	The period name already exists.
8058	Attendance shift is being used.
8059	Attendance period is being used.
8060	The user has no permissions of video intercom
8061	Search license information error
8064	The householder is exists
8066	The person already exist in person group
8067	The extract task has not finished
10001	The parkingLotName is Empty!
10002	The idleLot is more than totalLot!
10003	The parkingLotName is already exists!
10004	The carNo. is already exists or repeat!
10005	The device is not exists or deleted!
10006	The device is offLine!
10007	The time is not in timeTemplate!
10008	The parkingLot config error!
10009	Illegal alarm info!
10010	The cardNUm is not exists!
20002	edit model exception

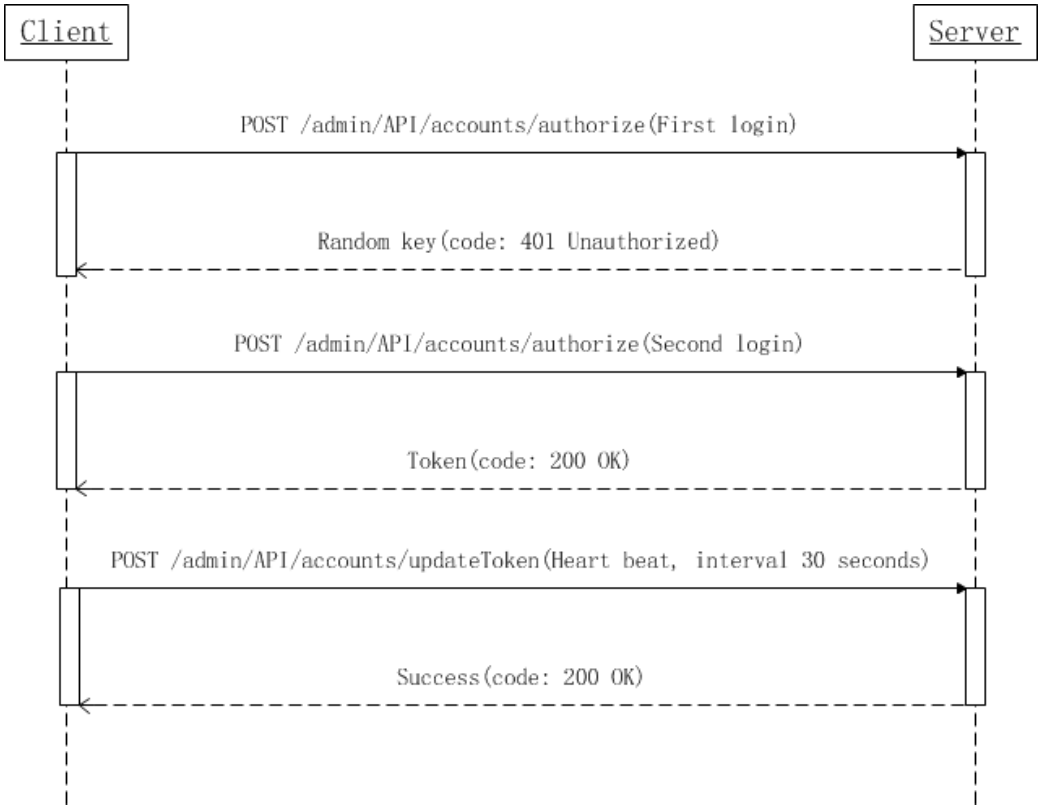
Error code	Error Message
20003	delete model exception
20004	modelName exists
20005	model label exists,delete fail
20007	label index exists
20008	label name exists
20009	save label exception
20010	edit label exception
20011	delete label exception
20013	parameter error
20014	The image size is oversized
20015	turn to point exception

2、Business Process

2.1、Login authentication

Authentication

- Login authentication requires two authentications. The first one takes the username, calculates the signature value according to the encrypted information returned by the platform, and then puts the calculated signature value into the second authentication request. After successful authentication, it is necessary to call the update token interface for keeping alive within the keep-alive duration.



First authentication

-The username is required for the first authentication request. At this time, the platform returns 401 no-authority error code and carries encrypted information in the message body. The encrypted information includes domain information (realm), random key and encryption type. Currently, the encryption method only supports MD5. -The request process is as follows:

```
POST /admin/API/accounts/authorize HTTP/1.1
Host: 10.35.92.66
Connection: close
Content-Type: application/json;charset=UTF-8
Content-Length: 67

{
    "userName" : "system",
    "ipAddress":    "",
    "clientType":   "WINPC"
}HTTP/1.1 401 Unauthorized
Server: Apache-Coyote/1.1
Set-Cookie: JSESSIONID=7994C0D4B07F194D0757882392DCC2C0; Path=/admi
n; HttpOnly
Content-Type: application/json;charset=utf-8
Content-Length: 66
Date: Wed, 19 Dec 2018 08:41:58 GMT
Connection: close
```

```
{"realm":"DSS","randomKey":"9c2b603650f54bcb","encryptType":"MD5"}
```

Second authentication

- A signature is required for the second authentication request. According to the username, password, domain information (realm), random key and encryption method (MD5), the signature is calculated as follows: Pseudo code examples:

```
temp = md5(password)
temp = md5(userName + temp)
temp = md5(temp)
temp = md5(userName + ":" + realm + ":" + temp)
signature = md5(temp + ":" + randomKey)
```

- As shown in the above pseudo code, MD5 encryption is required for five times, in which the parameters and result values are both strings and the letters of MD5 encrypted result values are all in lowercase. The assumed values are as follows:

Parameters	Value
userName	"system"
password	"admin123"
realm	"DSS"
randomKey	"9c2b603650f54bcb"

-The calculated values for each stage are as follows:

Times of encryptions	Parameters/Results	Value
1	Parameters	"admin123"
1	Result	"0192023a7bbd73250516f069df18b500"
2	Parameters	"system0192023a7bbd73250516f069df18b500"
2	Result	"5a0fdbe44b86807b5e5e127918bbc475"
3	Parameters	"5a0fdbe44b86807b5e5e127918bbc475"
3	Result	"1e27fadce9af09e120ab5142a83a679e"
4	Parameters	"system:DSS:1e27fadce9af09e120ab5142a83a679e"
4	Result	"675ae42820b189caa27d63b4b3264232"
5	Parameters	"675ae42820b189caa27d63b4b3264232:9c2b603650f54bcb"
5	Result	"7b6f728a6fa77e1ea5984afda8008e42"

-The request process is as follows:

```
POST /admin/API/accounts/authorize HTTP/1.1
```

```
Host: 10.35.92.66
Connection: close
Content-Type: application/json;charset=UTF-8
Content-Length: 186

{
    "userName" : "system",
    "randomKey":    "9c2b603650f54bcb",
    "mac":  "",
    "encryptType" : "MD5",
    "ipAddress":    "",
    "signature":    "7b6f728a6fa77e1ea5984afda8008e42",
    "clientType":   "WINPC"
}HTTP/1.1 200 OK
Server: Apache-Coyote/1.1
Set-Cookie: JSESSIONID=042C68CF5399B1898DCFD78B35A90896; Path=/admi
n; HttpOnly
Content-Type: application/json;charset=utf-8
Transfer-Encoding: chunked
Date: Wed, 19 Dec 2018 08:41:58 GMT
Connection: close

{"duration":30,"token":"a0e5844699db4cf8a2f9afaae11becd4","userId":"
1","versionInfo":{"lastVersion":"1074472","updateUrl":"/admin/x86/Ge
neral_DSS-Express_Client-x86_V1.000.0000002.0.R.20181213.exe;/admin/
x64/General_DSS-Express_Client-x64_V1.000.0000002.0.R.20181213.exe
"},"sipNum":"8888881000"}
```

-After the second authentication is successful, the meaning of the field returned by the platform is as follows: duration: Keep-alive duration. It is necessary to call the update token interface (/admin/API/accounts/updateToken) for keeping alive during the keep-alive duration. token: Token. In subsequent requests, the token needs to be placed in the request header and the field name is X-Subject-Token. userId: User ID. VersionInfo: Client version and client download address.

Update token

- After successful authentication, it is necessary to call the update token interface for survival within the survival duration. The keep-alive interval is suggested to be 3/4 of the keep-alive duration. In the request to update the token, a signature value needs to be carried in the message body, and the signature value here needs to be MD5 calculated according to the fourth calculation result of the second authentication and the latest token value. The first token value is the token returned when the second authentication is successful, and the

subsequent token value is the one returned by the updatetoken. The pseudo code is as follows:

```
signature = md5(temp + ":" + token)
```

-The assumed values are as follows:

Parameters	Value
temp	"675ae42820b189caa27d63b4b3264232"
token	"a0e5844699db4cf8a2f9afaae11becd4"

-The calculated values are as follows:

Times of encryptions	Parameters/Results	Value
1	Parameters	"675ae42820b189caa27d63b4b3264232:a0e5844699db4cf8a2f9afaae11becd4"
1	Result	"5bce0dc0059363e251a706a8b1b281a9"

-The request process is as follows:

```
POST /admin/API/accounts/updateToken HTTP/1.1
Host: 10.35.92.66
Connection: close
Content-Type: application/json;charset=UTF-8
X-Subject-Token: a0e5844699db4cf8a2f9afaae11becd4
Content-Length: 52

{
    "signature":    "5bce0dc0059363e251a706a8b1b281a9"
}HTTP/1.1 200 OK
Server: Apache-Coyote/1.1
Set-Cookie: JSESSIONID=C778CC5FC4302D4C71DB10F7FF34391B; Path=/admin; HttpOnly
Content-Type: application/json;charset=UTF-8
Transfer-Encoding: chunked
Date: Wed, 19 Dec 2018 08:43:04 GMT
Connection: close

{"code":1000,"desc":"Success","data":{"token":"a0e5844699db4cf8a2f9afaae11becd4","duration":30}}
```

The client needs to update the token value in the returned message to the local token value and use the updated token value for subsequent calculations.

Note:

1. The platform will not check the signature brought in the updateToken request, but there is no guarantee that no check will be added subsequently. Therefore, it is recommended to calculate according to the instructions in the document.
2. The token of the current version is fixed, that is, the token value returned in updatetoken is the same as that returned in the second authentication. However, there is no guarantee that it will not change, so it is recommended to update with the token value returned by updatetoken.

Cancel authentication

When the client signs out, it needs to call the unauthorized interface. The message body of this interface needs to bring the username and token.

-The request process is as follows:

```
POST /admin/API/accounts/unauthorize HTTP/1.1
Host: 10.35.92.66
Connection: close
Content-Type: application/json;charset=UTF-8
X-Subject-Token: a0e5844699db4cf8a2f9afaae11becd4
Content-Length: 71

{
    "userName" : "system",
    "token":    "a0e5844699db4cf8a2f9afaae11becd4"
}HTTP/1.1 200 OK
Server: Apache-Coyote/1.1
Set-Cookie: JSESSIONID=EB3DEE86E1FD0DAAEADF89E19CD2B1C6; Path=/admi
n; HttpOnly
Content-Type: application/json;charset=UTF-8
Transfer-Encoding: chunked
Date: Wed, 19 Dec 2018 08:43:17 GMT
Connection: close

{"code":1000,"desc":"Success"}
```

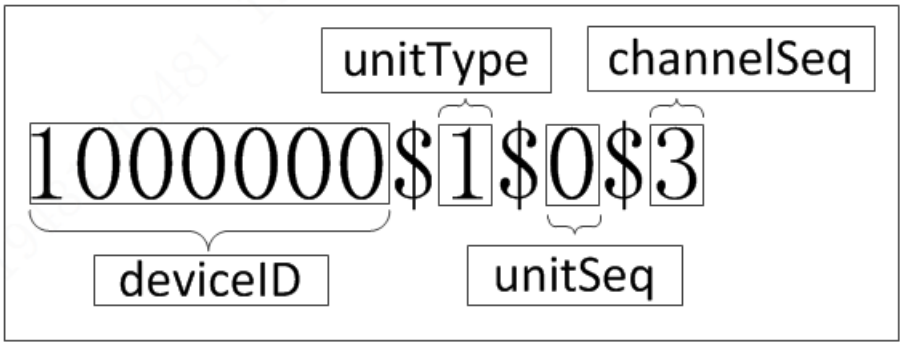
2.2、Device department tree

- The acquisition of the device department tree involves two protocols: The device department tree (/admin/API/tree/deviceorg) and device list (/admin/API/tree/devices)
    1. The device department tree can obtain the relative relationships of all departments, devices and channels.

2. The device list obtains details of devices, units and channels, such as device IP, port, type and other information.

Channel coding rules

A channel ID consists of deviceId, unitType, unitSeq and channelSeq, separated by a $ symbol, as shown in the following figure:



- Where The deviceID is a 7-digit number. Starting from 1,000,000, the codes of the newly added devices in the platform increase in sequence. UnitType is a unit type. Refer to the following table for specific meaning. UnitSeq is the unit serial number, starting from 0. ChannelSeq is the channel number, starting from 0, indicating which channel is under this unit. In the above figure, 1000000$1$0$3 indicates the 4th channel of the 1st unit of the coding unit of No.1,000,000 device.
- Unit type parameters

unitType	Unit type
1	Encoding unit
2	Decoding unit
3	Alarm input unit
4	Alarm output unit
5	Screen input unit
6	Screen output unit
7	Access control unit
8	Voice control unit
9	Transcoding unit
10	Dynamic ring unit
11	POS unit
12	Virtual unit

Device department tree

- Multiple departments and multiple devices can be carried under departments, different types of units under devices, and multiple channels under units. The device department interface supports condition search for channelType and orgCode. The

channel type and department are placed in the parameters of the request url as follows:

```
GET /admin/API/tree/deviceOrg?channelType=1,2,3,4,5,6,7,8,10,11,12,14,15,33&sort=&orgCode= HTTP/1.1
```

- -The meaning of parameters is as follows:

Parameter Name	Type	Description
channelType	string	This is a comma-separated string of numbers, with each number in the string representing a channel type. If the parameter channelType is null, all channel types are requested. The channel type corresponds to the unit type associated with the channel, and the specific meaning can refer to the unit type.
sort	string	The field is not used and is null by default.
orgCode	string	Department code; if orgCode is blank, all departments are searched.

- Platform reply data is Json string, departments is array type, each array element is an department, each department has attribute field, device list and channel list, and the returned data format is as follows:

```
{
"code": 1000,
"desc": "Success",
"data": {
    "departments": [
        {
            "code": "001",
            "parentCode": "",
            "name":"root",
            "orgType":"1",
            "modifyTime": "1483528546",
            "deparmentsCount": "1",
            "domainId": "0",
            "device": [
                {
                    "id": "1000004",
                    "sort": "0"
                },
                {
                    "id": "1000738",
                    "sort": "0"
                }
            ],
            "channel": [
```

```
                {
                    "id": "1000004$7$0$0",
                    "sort": "0"
                },
                {
                    "id": "1000004$7$0$1",
                    "sort": "0"
                }
            ]
        },
        {
            "code": "001001",
            "parentCode": "001",
            "name": "Access Control",
            "orgType":"1",
            "modifyTime": "1544749099",
            "deparmentsCount": "0",
            "domainId": "",
            "device": [],
            "channel": []
        }
    ]
}
}
```


- The attribute fields of the department have the following meanings:

Parameter Name	Type	Description
code	string	The code of each department is unique, and the code of the root node is "001".
parentCode	string	Each department has a parent node, and its parent node parentCode is an empty string. The department finds its corresponding parent department through parentCode.
name	string	Department name
orgType	string	The department type is 1 by default, indicating the basic department.
modifyTime	string	Department modification time, timestamp type
deparmentsCount	string	Number of sub-departments
domainId	string	Domain ID

- Devices under the department have the following meanings in Device:

Parameter Name	Type	Description
id	string	Device ID
sort	string	Sort value

-The channels under the group are in channel, and the meanings of each field are as follows:

Parameter Name	Type	Description
id	string	Channel ID
sort	string	Sort value

- The data returned by request can form an department tree which has devices and channels, but without their details. The details should be obtained through the device list.

Device list

- Different types of units are carried under the device, and specific passages under the units.
- When requesting a list of devices, the optional parameters are composed of orgCode, deviceCodes and categories. These parameters are combined into Json strings and placed in the message body of the request. Three condition parameters are of relation “and”, as shown below:

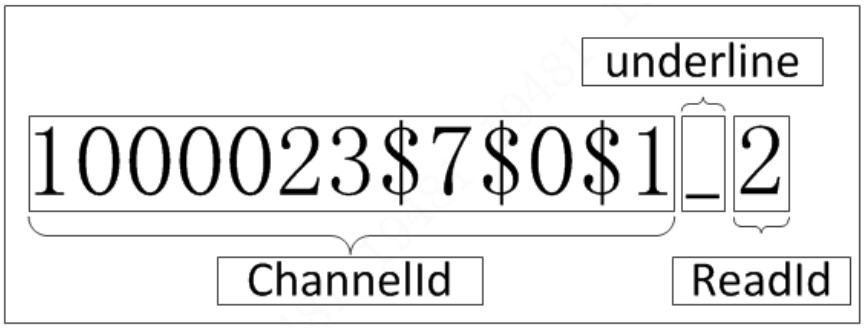
```
{
"orgCode":"",
"deviceCodes":  [],
"categories":   []
}
```

Parameter Name	Type	Description
orgCode	string	If not used, it is null by default.
deviceCodes	string	Device code: The specific code of the device to be queried; if it is blank, it means obtaining all devices.
categories	string	Device category: Filter the type of device; if blank, all device types are queried.

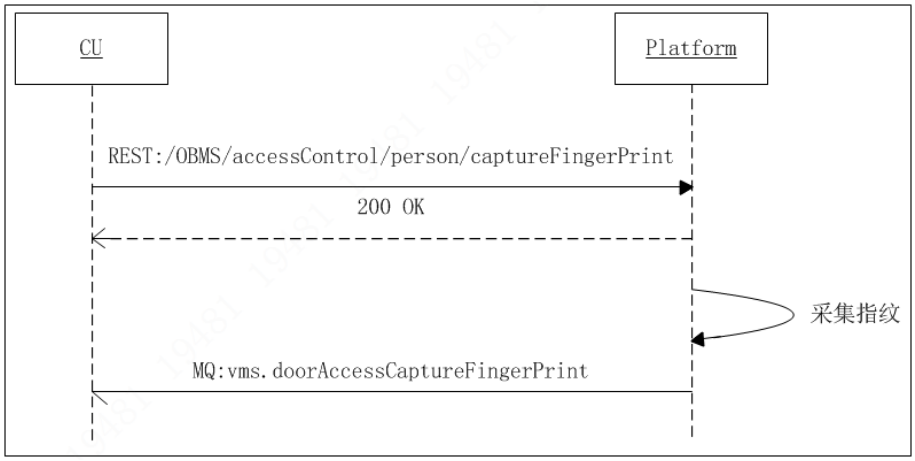
- Platform reply data is Json string, devices node in Json is of the array type, with each array element being a device, and each device having device attributes and units. Units node is of the array type, with each array element being a unit, and each unit having unit attributes and channels. Channels node is of the array type, with each array element being a channel, and the attributes of the channel being under the channel.
- Device attribute field For the meaning of each field, see Device List Protocol.
- Unit attribute field For the meaning of each field, see Device List Protocol.
- Channel attribute field For the meaning of each field, see Device List Protocol.

2.3、Personnel Management

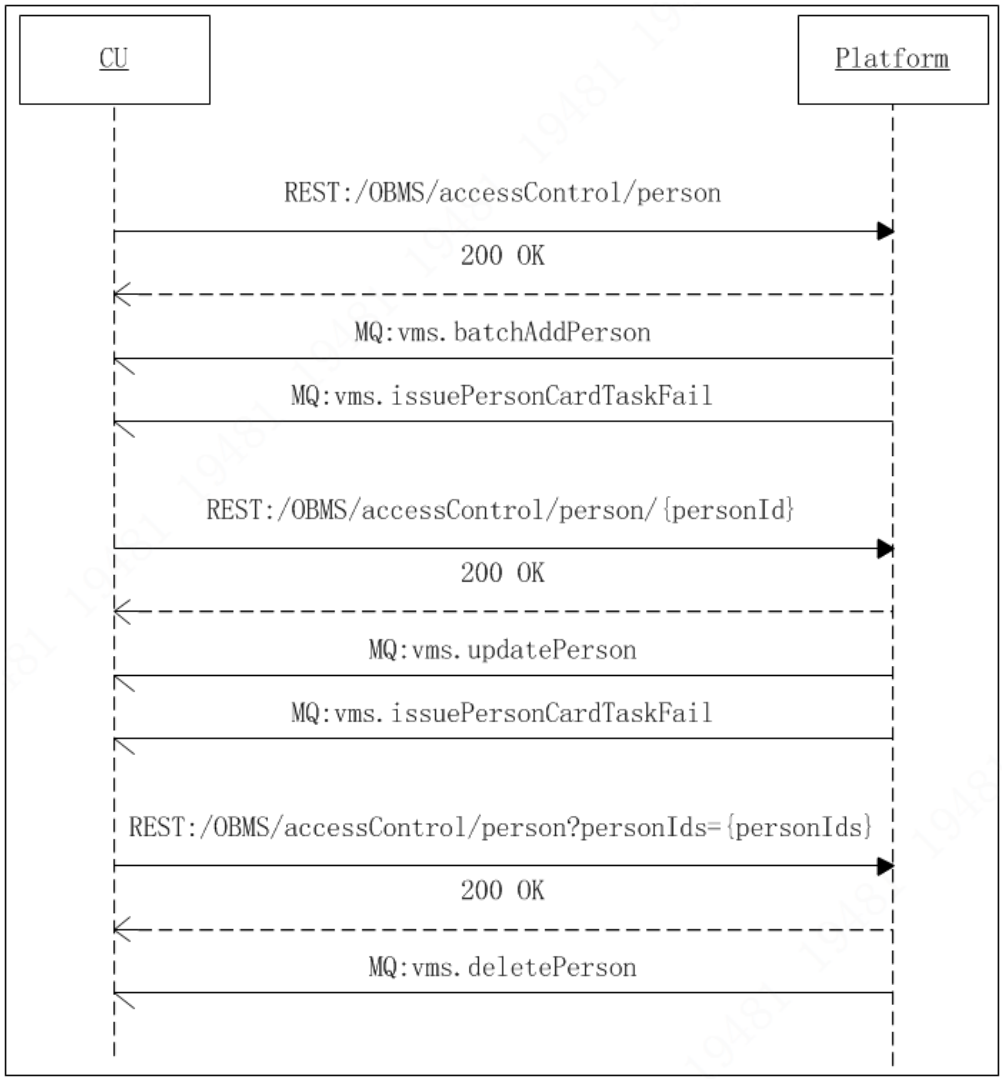
Get card reader data of access control channel Call (/OBMS/accessControl/door/reader?deviceCode={deviceCode}) interface to get the card reader data of access control channel. The channelId and readId are combined into a new readerId with underlined connection in the middle. The new readId is used to collect fingerprints.



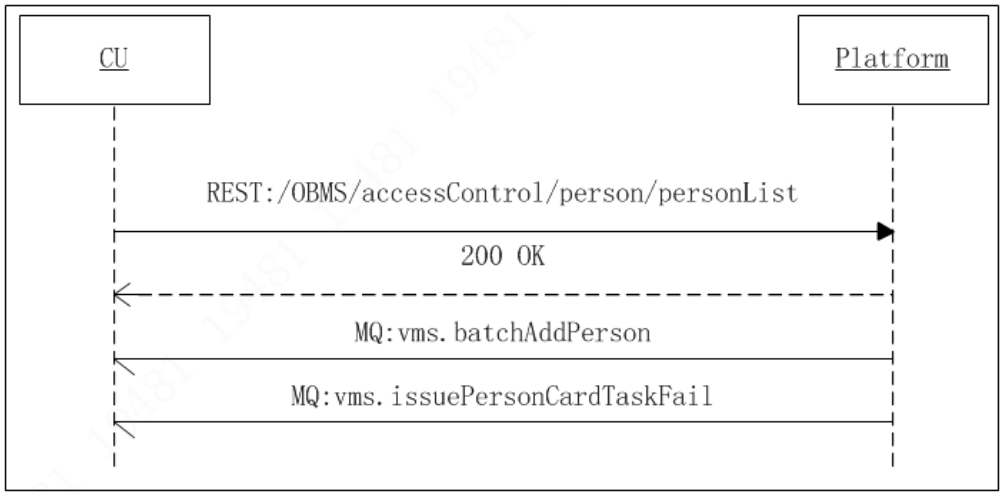
Collect fingerprints Call (/OBMS/ACCESS CONTROL/PERSON/CaptureFingerPrint) interface and issue the command to acquire fingerprint. After issuing the command to collect fingerprints successfully, the card reader enters the fingerprint collection mode, and at this time, the reader is pressed with fingers for fingerprint collection. After the fingerprint collection is successful, the platform reports the fingerprint collection notification (VMS. DOORACESSESCAPTURE FINGERPRINT) through MQ. The notification contains the fingerprint data used to configure person information.



Add, modify and delete person Call (/OBMS/accessControl/person) interface to add person information. After the successful addition of person information, the platform reports the notification on adding person successfully (vms.batchAddPerson) and on failure to issue the person (vms.issuePersonCardTaskFail) through MQ. Call (/OBMS/accessControl/person/{personId}) interface to modify person information. After the successful modification of person information, the platform reports the notification on successfully modifying person (vms.updatePerson) and on failure to issue the person (vms.issuePersonCardTaskFail) through MQ. Call (/OBMS/accessControl/person?personIds={personIds}) interface to delete person information. After successful deletion of person information, the platform reports the notification on successfully deleting the person (vms.deletePerson) through MQ.



Add person in batch Call (/OBMS/accessControl/person/personList) interface to add person information in batch. After the successful addition of person information, the platform reports the notification on adding person successfully (vms.batchAddPerson) and on failure to issue the person (vms.issuePersonCardTaskFail) through MQ.



Search person list Call (/OBMS/accessControl/personList) interface to get person list. Call (/OBMS/accessControl/personCount) interface to get the total number of persons.

The third-party docking personnel management generally does not need to access MQ, but the platform client needs to. Person card password encrypted by AES, this is a java util:

```
package test.encrypt;

import javax.crypto.*;
import javax.crypto.spec.SecretKeySpec;

/**
 * AES encrypt
 */
public class EncryptionUtils {

    // Secret key
    public static final String AES_KEY = "4rfvgy7UJMko0pqa";

    // HEX value
     private static final String HEX_VALUE = "0123456789ABCDEF";

    /**
     * AES_128_ECB_PKCS5 encrypt
     * @param content
     * @param key
     * @return
     */
    public static String encryptWithAES(String content, String key) {
        try {
            SecretKey keySpec = new SecretKeySpec(key.getBytes("UTF-8"), "AES");
            Cipher cipher = Cipher.getInstance("AES/ECB/PKCS5Padding");
```

```
            cipher.init(Cipher.ENCRYPT_MODE, keySpec);
            byte[] bytes = cipher.doFinal(content.getBytes("UTF-8"));
            return bytesToHexString(bytes).toLowerCase();
        } catch (Exception e) {
            return content;
        }
    }

    /**
     * AES_128_ECB_PKCS5 decrypt
     * @param content
     * @param key
     * @return
     */
    public static String decryptWithAES(String content, String key) {
        try {
            SecretKey keySpec = new SecretKeySpec(key.getBytes("UTF-8"), "AES");
            Cipher cipher = Cipher.getInstance("AES/ECB/PKCS5Padding");
            cipher.init(Cipher.DECRYPT_MODE, keySpec);
            byte[] bytes = cipher.doFinal(hexStringToBytes(content.toUpperCase()));
            return new String(bytes, "UTF-8");
        } catch (Exception e) {
            return content;
        }
    }

    /**
     * Bytes to Hex String
```

```
 * @param bytes
 * @return
 */
public static String bytesToHexString(byte[] bytes) {
	StringBuilder sb = new StringBuilder();
	if (null == bytes || bytes.length <= 0) {
		return null;
	}
	for (int i = 0; i < bytes.length; i++) {
		int v = bytes[i] & 0xFF;
		String hv = Integer.toHexString(v);
		if (hv.length() < 2) {
			sb.append(0);
		}
		sb.append(hv);
	}
	return sb.toString().toUpperCase();
}

/**
 * Hex String to Bytes
 * @param str
 * @return
 */
private static byte[] hexStringToBytes(String str) {
	int len = str.length() / 2;
	byte[] result = new byte[len];
	char[] achar = str.toCharArray();
```

```
            for (int i = 0; i < len; i++) {
                int pos = i * 2;
                result[i] = (byte) ((byte) toByte(achar[pos]) << 4 | toByte(achar[pos + 1]));
            }
            return result;
        }

        /**
         * byte of hex value
         * @param c
         * @return
         */
        private static byte toByte(char c) {
            return (byte) HEX_VALUE.indexOf(c);
        }

        /**
         * test
         * @param args
         */
        public static void main(String[] args) {
            // print 447f26333607fa412dbd36f660899db9
            System.out.println(EncryptionUtils.encryptWithAES("123456", AES_KEY));
            // print 123456
            System.out.println(EncryptionUtils.decryptWithAES("447f26333607fa412dbd36f660899db9",
AES_KEY));
        }
}
```

3、API Interface

3.1、Login Authentication

3.1.1、First login

Brief description:

- First login when creating session

Request URL:

- /admin/API/accounts/authorize

Request example

```
POST /admin/API/accounts/authorize HTTP/1.1
Host: 10.35.93.66
Connection: close
Content-Type: application/json;charset=UTF-8
```

Request method:

- POST

Request example

```
{
    "userName": "system",
    "ipAddress": "",
    "clientType": "WINPC"
}
```

Parameters:

Parameter Name	Type	Description
userName	string	Username
ipAddress	string	IP address of client PC
clientType	string	Client type

Return example

```
{
    "realm": "DSS",
    "randomKey": "c078b9d42eb74ebb",
    "encryptType" : "MD5"
}
```

Description of return parameters

Parameter Name	Type	Description
realm	string	Domain information
randomKey	string	Random value
encryptType	string	Encryption type

Notes

– For more returned error codes, see the error code description on the home page

3.1.2、Second login

Brief description:

- Second login when creating session

Request URL:

- /admin/API/accounts/authorize

Request example

```
POST /admin/API/accounts/authorize HTTP/1.1
Host: 10.35.93.66
Connection: close
Content-Type: application/json;charset=UTF-8
```

Request method:

- POST

Request example

```
{
    "userName": "system",
    "randomKey": "c078b9d42eb74ebb",
    "mac": "30:9c:23:79:40:08",
    "ipAddress": "",
    "signature": "a83edfbd6a98f8cc3aedfea36d9b13bb",
    "clientType": "WINPC",
    "encryptType": "MD5",
    "userType": "0"
}
```

Parameters:

Parameter Name	Type	Description
userName	string	Username
randomKey	string	Random value
mac	string	Mac address
ipAddress	string	IP address
signature	string	Authentication data, authentication data=AES (PWD, randomKey) if the user type is domain user.
clientType	string	User Type
encryptTypee	string	Encryption type
userType	string	0=System user, 1=Domain user

Return example


```
{
    "duration": 30,
    "token": "5095d4a0c5cc4340b72b7e5c64fa3170",
    "userId": "1",
    "sipNum": "8888881000",
    "emapUrl": null,
    "serviceAbilty": null,
    "versionInfo": {
        "lastVersion": "1016871",
        "updateUrl": "/admin/x86/General_DSS-Express_Client-x86_V1.000.0000002.0.R.20181011.exe;/admin/x64/General_DSS-Express_Client-x64_V1.000.0000002.0.R.20181011.exe"
    }
}
```


Description of return parameters

Parameter Name	Type	Description
duration	int	Keep-alive interval
token	string	Token
userId	string	User ID
sipNum	string	Called number
emapUrl	string	Electronic map URL
serviceAbilty	object	None
versionInfo	object	Version information
- lastVersion	string	Latest client version
- updateUrl	string	Client update URL

Notes

– For more returned error codes, see the error code description on the home page

3.1.3、Update token

Brief description:

- Update token

Request URL:

- /admin/API/accounts/updateToken
- X-Subject-Token: {token}

Request example

```
POST /admin/API/accounts/updateToken HTTP/1.1
Host: 10.35.93.66
Connection: close
Content-Type: application/json;charset=UTF-8
X-Subject-Token: 841e473bacef4b31abc5f53272afeb84
```

Request method:

- POST

Request example

```
{
    "signature": "6684b3858a56d80c85bfa07f625baa10"
}
```

Parameters:

Parameter Name	Type	Description
signature	string	Signature

Return example

```
{
    "code": 1000,
    "desc": "Success",
    "data": {
        "token": "885c9c2afd784120922e9f5ef164b799",
        "duration": 30
    }
}
```

Description of return parameters

Parameter Name	Type	Description
code	int	Error Code
desc	string	Result description
data	object	None
- token	string	Token number
- duration	int	Next update interval

Notes

– For more returned error codes, see the error code description on the home page

3.1.4、Modify password

Brief description:

- Modify user password

Request URL:

- /admin/API/Rights/User/Password/{userId}
- X-Subject-Token: {token}

Request example

```
PUT /admin/API/Rights/User/Password/1 HTTP/1.1
Host: 10.35.93.66
Connection: close
Content-Type: application/json;charset=UTF-8
X-Subject-Token: 841e473bacef4b31abc5f53272afeb84
```

Parameter Name	Type	Description
userId	string	User ID
token	string	The token obtained when logging in

Request method:

- PUT

Request example

```
{
    "oldPassword": "1e27fadce9af09e120ab5142a83a679e",
    "newPassword": "3b314ab6808943c6732b52039a5290b2"
}
```

Parameters:

Parameter Name	Type	Description
oldPassword	string	Previous password, encrypted in an encryption type returned when creating session.
newPassword	string	New password, encrypted in an encryption type returned when creating session.

Note：

```
Encryption mode:
temp = md5(password)
temp = md5(userName + temp)
temp = md5(temp)
userName-user name, password-user password
```

Return example

```
{
    "code": 1000,
    "desc": "Success"
}
```

Description of return parameters

Parameter Name	Type	Description
code	int	Error Code
desc	string	Result description

Notes

– For more returned error codes, see the error code description on the home page

3.1.5、Logout

Brief description:

- Request to destroy a session authenticated by a user.

Request URL:

- /admin/API/accounts/unauthorize
- X-Subject-Token: {token}

请求示例

```
POST /admin/API/accounts/unauthorize HTTP/1.1
Host: 10.35.93.66
Connection: close
```

```
Content-Type: application/json;charset=UTF-8
X-Subject-Token: 841e473bacef4b31abc5f53272afeb84
```

Parameter Name	Type	Description
token	string	The token obtained when logging in

Request method:

- POST

Request example

```
{
    "userName": "system",
    "token": "bb15670e2ea24bac832dca223c7b52ac"
}
```

Parameters:

Parameter Name	Type	Description
userName	string	User Name
token	string	The token obtained when logging in

Return example

```
{
    "code": 1000,
    "desc": "Success"
}
```

Description of return parameters

Parameter Name	Type	Description
code	int	Error Code
desc	string	Result description

Notes

– For more returned error codes, see the error code description on the home page

3.2、Device Tree

3.2.1、Getting Device Organization Tree (all)

Brief description:

- Get device organization tree (all)

Request URL:

- /admin/API/tree/deviceOrg?channelTypes={channelTypes}&sort={sort}&orgCode={orgCode}
- X-Subject-Token: {token}

Request example

```
GET /admin/API/tree/deviceOrg?channelTypes=1,2,3,4,5,6,7,8,10,11,12,14,15,33&sort=&orgCode= HTTP/1.1
Host: 10.35.92.66
Connection: close
Content-Type: application/json;charset=UTF-8
X-Subject-Token: 433b9cac285d434683643f79fb86be6b
```

Parameter Name	Type	Description
channelTypes	string	Channel type
sort	string	Invalid parameter, null
orgCode	string	Organization code which means root node 001 if it is null
token	string	The token obtained when logging in

Notes:

```
channelTypes: 1 = Encoder channel, 2 = Decoder channel, 3 = Alarm input channel, 4 = Alarm output channel, 5 = Large screen input, 6 = Large screen output, 7 = Access control, 8 = Voice control, 9 = Transcoding, 10 = Power environment, 11 = POS; if the type is null, the channel is not displayed
```

Request method:

- GET

Request example

```
Empty
```

Parameters: None

Return example

```
{
    "code": 1000,
    "desc": "Success",
    "data": {
        "departments": [
            {
                "code": "001",
                "parentCode": "",
```

```
                "name": "root",
                "orgType": "1",
                "modifyTime": "1483528546",
                "deparmentsCount": "1",
                "domainId": "0",
                "device": [
                    {
                        "id": "1000004",
                        "sort": "0"
                    },
                    {
                        "id": "1000738",
                        "sort": "0"
                    },
                    {
                        "id": "1000747",
                        "sort": "0"
                    }
                ],
                "channel": [
                    {
                        "id": "1000004$7$0$0",
                        "sort": "0"
                    },
                    {
                        "id": "1000004$7$0$1",
                        "sort": "0"
                    },
                    ...
                ]
            },
            {
                "code": "001001",
                "parentCode": "001",
                "name": "test99",
                "orgType": "1",
                "modifyTime": "1544749099",
                "deparmentsCount": "0",
                "domainId": "",
                "device": [],
                "channel": []
            }
        ]
    }
```

```
}
```

Description of return parameters

Parameter Name	Type	Description
code	int	Error Code
desc	string	Result description
data	object	None
- departments	object	Organization information
- - code	string	Node code; 001 is the root node
- - parentCode	string	The parent code of the current node
- - name	string	Node name
- - orgType	string	Node Type
- - modifyTime	string	Change date
- - deparmentsCount	string	Number of subordinate organizations
- - domainId	string	Domain ID
- - device	object	Device node under the organization
- - - id	string	Device ID
- - - sort	string	Sort value
- - channel	object	Channel node under the organization
- - - id	string	Channel ID
- - - sort	string	Sort value

Notes

- For more returned error codes, see the error code description on the home page

3.2.2、Getting Device List (all)

Brief description:

- Get device list.

Request URL:

- /admin/API/tree/devices
- X-Subject-Token: {token}

Request example

```
POST /admin/API/tree/devices HTTP/1.1
Host: 10.35.92.66
Connection: close
Content-Type: application/json;charset=UTF-8
X-Subject-Token: 433b9cac285d434683643f79fb86be6b
```

Parameter Name	Type	Description
token	string	The token obtained when logging in

Request method：

- POST

Request example

```
{
    "orgCode": "",
    "deviceCodes": [],
    "categories": []
}
```

Parameters：

Parameter Name	Type	Description
orgCode	int	Organization code
deviceCodes	string	Device code, if it is null, all devices are obtained
categories	string	Device category, 1-Encoder,5-ANPR,8-Access control

Return example

```
{
    "code": 1000,
    "desc": "Success",
    "data": {
        "devices": [
            {
                "code": "1000004",
                "name": "10.35.93.67",
                "category": "8",
                "type": "2",
                "model": "4",
                "status": "1",
                "offlineReason": "0",
                "protocol": "1",
                "manufacturer": "1",
                "deviceIp": "10.35.93.67",
                "devicePort": "37777",
                "proxyIp": "10.35.92.66",
                "proxyPort": "",
                "userName": "admin",
                "password": "8c8944adede71eea53bec5d7624ea53a",
                "loginType": "0",
```

```
            "sn": "",
            "orgCode": "001",
            "units": [
                {
                    "unitType": "7",
                    "unitSeq": "0",
                    "fingerPrintAuth": "0",
                    "cardAuth": "0",
                    "faceAuth": "0",
                    "userIsolate": "0",
                    "unlockModes": "",
                    "channels": [
                        {
                            "channelCode": "1000004$7$0$0",
                            "channelName": "Door1",
                            "channelSeq": 0,
                            "status": "1",
                            "db33Code": "",
                            "intelliState": "",
                            "capability": "0",
                            "domainId": ""
                        },
                        {
                            "channelCode": "1000004$7$0$1",
                            "channelName": "Door2",
                            "channelSeq": 1,
                            "status": "1",
                            "db33Code": "",
                            "intelliState": "",
                            "capability": "0",
                            "domainId": ""
                        },
                        ...
                    ]
                }
            ],
            "sipId": "",
            "sipPwd": "",
            "unitEnable": "0",
            "buildingEnable": "0",
            "domainId": "",
            "modifyTime": ""
        },
        {
```

```
"code": "1000747",
"name": "NVR",
"category": "1",
"type": "6",
"model": "IPC_256",
"status": "1",
"offlineReason": "0",
"protocol": "1",
"manufacturer": "1",
"deviceIp": "10.35.106.16",
"devicePort": "37140",
"proxyIp": "",
"proxyPort": "",
"userName": "admin",
"password": "8c8944adede71eea53bec5d7624ea53a",
"loginType": "0",
"orgCode": "001",
"units": [
    {
        "unitType": "1",
        "unitSeq": "0",
        "assistStream": "3",
        "zeroChnEncode": "1",
        "streamType": "801",
        "channels": [
            {
                "channelCode": "1000747$1$0$0",
                "channelName": "IPC",
                "channelSeq": 0,
                "status": "1",
                "cameraType": "1",
                "channelType": "1",
                "cameraFunctions": "0",
                "forPeopleCount": "0",
                "faceFunctions": "0",
                "db33Code": "",
                "intelliState": "",
                "targetDetection": "0",
                "capability": "0",
                "domainId": ""
            },
            {
                "channelCode": "1000747$1$0$1",
                "channelName": "IPC",
```

```
                        "channelSeq": 1,
                        "status": "1",
                        "cameraType": "1",
                        "channelType": "1",
                        "cameraFunctions": "0",
                        "forPeopleCount": "0",
                        "faceFunctions": "0",
                        "db33Code": "",
                        "intelliState": "",
                        "targetDetection": "0",
                        "capability": "0",
                        "domainId": ""
                    }
                ]
            },
            {
                "unitType": "3",
                "unitSeq": "0",
                "channels": [
                    {
                        "channelCode": "1000747$3$0$0",
                        "channelName": "NVR_1",
                        "channelSeq": 0,
                        "status": "1",
                        "alarmType": "2",
                        "alarmLevel": "1",
                        "db33Code": "",
                        "intelliState": "",
                        "capability": "0",
                        "domainId": ""
                    },
                    {
                        "channelCode": "1000747$3$0$1",
                        "channelName": "NVR_2",
                        "channelSeq": 1,
                        "status": "1",
                        "alarmType": "2",
                        "alarmLevel": "1",
                        "db33Code": "",
                        "intelliState": "",
                        "capability": "0",
                        "domainId": ""
                    },
                    ...
```

```
                    ]
                },
                {
                    "unitType": "4",
                    "unitSeq": "0",
                    "channels": [
                        {
                            "channelCode": "1000747$4$0$0",
                            "channelName": "NVR_1",
                            "channelSeq": 0,
                            "status": "1",
                            "db33Code": "",
                            "intelliState": "",
                            "capability": "0",
                            "domainId": ""
                        },
                        {
                            "channelCode": "1000747$4$0$1",
                            "channelName": "NVR_2",
                            "channelSeq": 1,
                            "status": "1",
                            "db33Code": "",
                            "intelliState": "",
                            "capability": "0",
                            "domainId": ""
                        },
                        ...
                    ]
                }
            ],
            "unitEnable": "",
            "buildingEnable": "",
            "domainId": "",
            "modifyTime": ""
        },
        {
            "code": "1000738",
            "name": "IPC29",
            "category": "1",
            "type": "1",
            "model": "IPC_256",
            "status": "1",
            "offlineReason": "0",
            "protocol": "1",
```

```
                "manufacturer": "1",
                "deviceIp": "10.35.106.29",
                "devicePort": "31239",
                "proxyIp": "",
                "proxyPort": "",
                "userName": "admin",
                "password": "8c8944adede71eea53bec5d7624ea53a",
                "loginType": "0",
                "sn": "",
                "orgCode": "001",
                "units": [
                    {
                        "unitType": "1",
                        "unitSeq": "0",
                        "assistStream": "3",
                        "zeroChnEncode": "0",
                        "streamType": "801",
                        "channels": [
                            {
                                "channelCode": "1000738$1$0$0",
                                "channelName": "IPC29_1",
                                "channelSeq": 0,
                                "status": "1",
                                "cameraType": "2",
                                "channelType": "1",
                                "cameraFunctions": "0",
                                "forPeopleCount": "0",
                                "faceFunctions": "0",
                                "db33Code": "",
                                "intelliState": "",
                                "targetDetection": "0",
                                "capability": "0",
                                "domainId": ""
                            }
                        ]
                    }
                ],
                "unitEnable": "",
                "buildingEnable": "",
                "domainId": "",
                "modifyTime": ""
            }
        ]
    }
```

```
}
```

Description of return parameters

Parameter Name	Type	Description
code	int	Error Code
desc	string	Result description
data	object	None
- devices	object	Device Information
- - code	string	Device Code
- - name	string	Device name
- - category	string	Device categories
- - type	string	Device type
- - model	string	Model
- - status	string	Device status
- - offlineReason	string	Offline reason
- - protocol	string	Protocol
- - manufacturer	string	Manufacturer
- - deviceIp	string	Device IP
- - devicePort	string	Device ports
- - registId	string	Register ID
- - proxyIp	string	Proxy IP
- - proxyPort	string	Proxy port
- - userName	string	User name
- - password	string	Password
- - callNumber	string	SIP number
- - loginType	string	Login type
- - sn	string	SN number
- - orgCode	string	Department code
- - units	object	Unit information
- - - unitType	string	Unit type
- - - unitSeq	string	Unit sequence
- - - assistStream	string	Sub stream
- - - zeroChnEncode	string	Zero channel encode
- - - ssServiceId	string	Storage service ID
- - - ptsServiceId	string	PTS service ID
- - - streamType	string	Code-stream type
- - - decodeMode	string	Decoding mode
- - - streamMode	string	Stream processing mode
- - - conbineStatus	string	Video wall combining status, 0-combining is not supported,1-combining is supported
- - - thirdPartyControl	string	Third party control
- - - voiceServerIp	string	Voice service IP

Parameter Name	Type	Description
- - - voiceServerPort	string	Voice service port
- - - voiceStatusPort	string	Voice status port
- - - voiceClientIp	string	Voice client IP
- - - dynCode	string	Power environment resource code
- - - dynName	string	Power environment resource name
- - - dynType	string	Power environment resource type
- - - fingerPrintAuth	string	Whether fingerprint authentication is supported, 0-unknown, compatible with previous one (default),1-not supported,2-supported
- - - cardAuth	string	Whether card authentication is supported, 1-Yes,0-No
- - - faceAuth	string	Whether face recognition authentication is supported, 1-Yes,0-No
- - - userIsolate	string	Whether the user and card are separated,1 - Yes,0-No
- - - unlockModes	string	Unlock mode combination
- - - channels	object	Channel information
- - - - channelCode	string	Channel code
- - - - channelName	string	Channel name
- - - - channelSeq	string	Channel sequence
- - - - status	string	Enabling status
- - - - sn	string	SN number
- - - - gpsX	string	GPS(X)
- - - - gpsY	string	GPS(X)
- - - - mapId	string	Raster map ID
- - - - cameraType	string	Camera type
- - - - channelType	string	Channel type
- - - - remoteType	string	Remote type
- - - - cameraFunctions	string	Channel function set
- - - - multicastIp	string	Multicast IP
- - - - multicastPort	string	Multicast port
- - - - ipcIp	string	IPC(IP)
- - - - recordType	string	Record type
- - - - forPeopleCount	string	Whether it is used for people count
- - - - maxSplitNum	string	Maximum number of video wall split
- - - - interfaceType	string	Interface type
- - - - alarmType	string	Alarm type
- - - - alarmLevel	string	Alarm level
- - - - signalType	string	Signal type
- - - - accessType	string	Access control type
- - - - posType	string	Pos type
- - - - posModel	string	Access control type
- - - - dynCode	string	Power environment resource
- - - - faceFunctions	string	Face image function

Parameter Name	Type	Description
- - - - keyCode	string	Keyboard code
- - - - db33Code	string	Landmark code
- - - - intelliState	string	Intelligent status
- - - - faceAnalyseType	string	Face analysis type
- - - - videoSource	string	Video source for virtual channel
- - - - targetDetection	string	Target detection function
- - - - capability	string	Capability set
- - - - domainId	string	Domain ID
- - sipId	string	SIP ID
- - sipPwd	string	SIP password
- - unitEnable	string	Intercom unit enable
- - buildingEnable	string	Intercom building enable
- - softwareVersion	string	Software version
- - hardwareVersion	string	Hardware version
- - domainId	string	Domain ID
- - modifyTime	string	Change date

Notes

- For more returned error codes, see the error code description on the home page

3.2.3、Getting Device Organization Tree (hierarchical)

Brief description:

- Get device organization tree (hierarchical)

Request URL:

- /admin/API/tree/search
- X-Subject-Token: {token}

Request example

```
GET /admin/API/tree/search HTTP/1.1
Host: 10.35.92.55
Connection: close
Content-Type: application/json;charset=UTF-8
X-Subject-Token: 723ab08e7db844d0946b26d1cadf011f
```

Parameter Name	Type	Description
token	string	The token obtained when logging in

Request method:

- GET

Request example

```
{
    "id": "001",
    "nodeType": "1,2",
    "typeCode": "01;01;01,02,03,04,05,06,07,08,21,40,43;01,02,03,04,05,06,07,08,09,10,11,12",
    "keyType": "name",
    "keyWord": "123",
    "cascading": "0",
    "page": "1",
    "pageSize": "10",
    "act": ""
}
```

Parameters:

Parameter Name	Type	Description
id	string	Node ID
nodeType	string	Node type, separated by commas if there are multiple node types; 1-organization, 2-device, 3-channel
typeCode	string	Node filtering criteria, {org};{device};{unit&channel};{camera}
keyType	string	Keyword type; null, name-node name, ip-device IP, id-node ID
keyWord	string	Keywords
cascading	string	Whether cascading data is displayed; 0-not displayed, 1-displayed
page	string	Page
pageSize	string	Record number per page
act	string	Action; search-search; null-hierarchically obtained

Return example

```
{
    "code": 1000,
    "desc": "Success",
    "data": {
        "nextPage": "-1",
        "totalCount": "1",
        "results": [
            {
                "id": "001",
                "name": "root",
                "isParent": "true",
                "nodeType": "1",
```

```
                "domainId": "",
                "sort": "0"
            }
        ]
    }
}
```

Description of return parameters

Parameter Name	Type	Description
code	int	Error Code
desc	string	Result description
data	object	None
- nextPage	string	Next page, -1 means no next page
- totalCount	string	Total number of queries
- results	object	None
- - id	string	Node ID
- - name	string	Node name
- - isParent	string	Whether the node is a parent node
- - nodeType	string	Node type, 1-device organization,2-device,3-channel
- - domainId	string	Domain ID
- - sort	string	Sort

Notes

- For more returned error codes, see the error code description on the home page

3.2.4、Obtain subordinate information about a specified node based on the search data type

Brief description:

- Obtain subordinate information about a specified node based on the search data type

Request URL:

- /admin/API/tree/child-node
- X-Subject-Token: {token}

Request example

```
POST /admin/API/tree/child-node HTTP/1.1
Host: 10.35.92.55
Connection: close
```

```
Content-Type: application/json;charset=UTF-8
X-Subject-Token: 723ab08e7db844d0946b26d1cadf011f
```

Parameter Name	Type	Description
token	string	The token obtained when logging in

Request method:

- POST

Request example

```
{
    "dataType": "01,03;01,06,38,45",
    "isDomain": 1,
    "nodes": [
        {
            "nodeId": "001",
            "checked": true
        }
    ],
    "size": 100
}
```

Parameters:

Parameter Name	Type	Description
dataType	string	Queried data type
isDomain	number	Whether to obtain cascaded subordinate data: 0—no; 1—yes
nodes	object	Node information
- nodeId	string	Node number
- checked	boolean	Whether the node needs to be queried: true—no; false—yes
size	number	Length of the returned data. All data is returned by default if the parameter is left blank.

Return example

```
{
    "code": 1000,
    "desc": "Success",
    "data": {
        "results": [
            {
                "nodeId": "1000000",
                "nodeName": "37790",
                "pId": "001001",
```

```
                "gId": "001",
                "nodeType": "2"
            }
        ],
        "hasMoreNodes": false
    }
}
```

Description of return parameters

Parameter Name	Type	Description
code	number	Error code
desc	string	Result description
data	object	None
- results	object	None
- - nodeId	string	Node ID
- - nodeName	string	Node name
- - pId	string	Parent node ID
- - gId	string	Grandparent node ID
- - nodeType	string	Node type: 1—organization; 2—device; 3—channel
- hasMoreNodes	boolean	Whether there are more nodes: true—yes; false—no

Notes

- For more returned error codes, see the error code description on the home page

3.3、Personnel Management

3.3.1、Add one-card user

Brief description:

- Add one-card user.

Request URL:

- /OBMS/accessControl/person
- X-Subject-Token: {Token}

Request example

```
POST /OBMS/accessControl/person HTTP/1.1
Host: 10.33.68.138
Connection: close
Content-Type: application/json;charset=UTF-8
```

X-Subject-Token: d725f34e4f034038bf1771a8ea43c931		
Parameter Name	**Type**	**Description**
Token	string	The token obtained when logging in

Request method:

- POST

Request example

```
{
    "details": {
        "companyName": "test",
        "expireTime": "1877615999",
        "position": "test",
        "nickName": "test",
        "useTimes": "200",
        "remark": "test",
        "tel": "13611111111",
        "email": "test@test.com",
        "idType": "0",
        "maritalStatus": "2",
        "birthday": "2019-06-17",
        "idNum": "362532195805165563",
        "address": "city",
        "degree": "4",
        "nationalityId": "brazil",
        "initialTime": "1560700800"
    },
    "baseInfo": {
        "authority": "0",
        "householder": "0",
        "unitId": "1",
        "id": "",
        "pictureData": "",
        "lastName": " ",
        "personId": "2047",
        "firstName": "test",
        "departmentId": "001001",
        "cardType": "0",
        "gender": "1",
        "status": "0",
        "stageId": "",
        "roomId": "36",
        "buildingId": "12"
```

```
    },
    "accessRight": {
        "entranceDeviceCode": [],
        "channelId": [
            "1000006$7$0$0"
        ],
        "doorGroupId": [
            "1"
        ]
    },
    "authentication": {
        "password": "9bb8291f7f9426004b3a1586c78244e4",
        "cars": [
            {
                "carNo": "A2563D8",
                "carRight": "2",
                "carGroup": "1"
            }
        ],
        "infraredFaceCodes": [],
        "carportAmount": "1",
        "fingerPrints": [],
        "cards": [
            {
                "cardState": "0",
                "masterFlag": "1",
                "duressFlag": "0",
                "cardNo": "00666998",
                "changeDate": "1560700800",
                "issueDate": "1560700800"
            }
        ],
        "facePictures": [
            "/9j/4AAQS......FAH/9k="
        ]
    }
}
```


Parameters:

Parameter Name	Type	Description
details	object	User details
- companyName	string	Company name
- expireTime	string	End time of time stamp

Parameter Name	Type	Description
- position	string	Position
- nickName	string	Nick name (used for address book distribution)
- useTimes	string	200 by default
- remark	string	Notes
- tel	string	Phone number
- email	string	Email
- idType	string	ID type, 0—ID card, 1—military officer ID card, 2—student ID card, 3—driving license, 4—passport, 5—tax identification number
- maritalStatus	string	Marital status, 0—confidential, 1—married, 2—single
- birthday	string	Birthday
- idNum	string	ID number
- address	string	Address
- degree	string	Degree, 0—no, 1—primary school, 2—junior middle school, 3—senior middle school, 4—Bachelor, 5—Master, 6—Doctor, 7—Professor
- nationalityId	string	Reference nationality ID
- initialTime	string	Start time of time stamp
baseInfo	object	Basic information of user
- authority	string	Administrator, 0—no, 1—yes
- householder	string	Householder, 0—no, 1—yes
- unitId	string	Unit
- id	string	Unique constraint on id recorded in database
- pictureData	string	Picture data, face picture
- lastName	string	Last name
- personId	string	Person ID
- firstName	string	First name
- departmentId	string	Department ID
- cardType	string	Card type, 0—common, 1—blacklist, 2—visitor, 3—patrol, 4—VIP
- gender	string	Gender, 0—female, 1—male
- status	string	Status, 0—on the job, 1—quit
- stageId	string	Stage, only domestic version of express is valid.
- roomId	string	Room ID
- buildingId	string	Building ID
accessRight	object	User-associated permission information
- entranceDeviceCode	object	Entrance/exit device code
- channelId	object	Device channel id under user control
- doorGroupId	object	Door group id under user control

Parameter Name	Type	Description
authentication	object	User-related authentication, access card and fingerprint information. One card and multiple fingerprints can be associated for a user.
- password	string	Password, AES encryption
- cars	object	Cars information
- - carNo	string	License plate number
- - carRight	string	Car permission, 1—blacklist, 2—whitelist; 2 by default
- - carGroup	string	Car group, 1—no, 3—common, 4—VIP; 1 by default
- infraredFaceCodes	object	Face feature
- carportAmount	string	Carport amount for user
- fingerPrints	object	Fingerprint information
- - name	string	Fingerprint name
- - fingerPrint	string	Fingerprint code/600 bits
- - threadFlag	string	Duress or not, 0—no, 1—yes
- - masterFlag	string	Master fingerprint or not, 0—no, 1—yes
- cards	object	Cards information
- - cardState	string	Card state, -1—unknown, 0—normal, 1—reported for loss, 2—canceled, 4—frozen, 8—arrearage, 16—overdue, 32—arrearage in advance
- - masterFlag	string	Master card or not (Generation 1 card), 0—no, 1—yes
- - duressFlag	string	Duress or not, 0—no, 1—yes
- - cardNo	string	Card No., length 8 or 16, range 0-9,A-F
- - changeDate	string	Change date
- - issueDate	string	Card issue date
- facePictures	object	Face picture _ base64

Return example


```
{
    "code": 1000,
    "desc": "Success",
    "data": {
        "personId": "2047"
    }
}
OR Repeat car num
{
    "code": 10004,
    "desc": "The carNo. is alresdy exists or repeat!",
    "data": {
        "repeatCarNos": [],
        "existingCarNos": ["A2563D8"]
```

```
    }
}
```

Description of return parameters

Parameter Name	Type	Description
code	int	Error Code
desc	string	Result description
data	object	None
- personId	string	Person ID
- repeatCarNos	object	Repeat car num
- existingCarNos	object	Exist car num

Notes

– For more returned error codes, see the error code description on the home page

3.3.2、Update one-card user

Brief description:

- Update one-card user.

Request URL:

- /OBMS/accessControl/person/{personId}
- X-Subject-Token: {Token}

Request example

```
PUT /OBMS/accessControl/person/2047 HTTP/1.1
Host: 10.33.68.138
Connection: close
Content-Type: application/json;charset=UTF-8
X-Subject-Token: d725f34e4f034038bf1771a8ea43c931
```

Parameter Name	Type	Description
Token	string	The token obtained when logging in
personId	string	Person ID

Request method:

- PUT

Request example

```
{
    "details": {
        "companyName": "test",
        "expireTime": "1877615999",
        "position": "test",
        "nickName": "test",
        "useTimes": "200",
        "remark": "test",
        "tel": "13611111111",
        "email": " test@test.com ",
        "idType": "5",
        "maritalStatus": "1",
        "birthday": "2019-06-17",
        "idNum": "",
        "address": "city",
        "degree": "4",
        "nationalityId": "brazil",
        "initialTime": "1560700800"
    },
    "baseInfo": {
        "authority": "0",
        "householder": "0",
        "unitId": "1",
        "id": "3",
        "pictureData": "",
        "lastName": " ",
        "personId": "2047",
        "firstName": "test",
        "departmentId": "001001",
        "cardType": "3",
        "gender": "1",
        "status": "0",
        "stageId": "",
        "roomId": "36",
        "buildingId": "12"
    },
    "accessRight": {
        "entranceDeviceCode": [],
        "channelId": [
            "1000006$7$0$0"
        ],
        "doorGroupId": [
            "1"
        ]
```

```
    },
    "authentication": {
        "password": "9bb8291f7f9426004b3a1586c78244e4",
        "cars": [
            {
                "carNo": "...A2563D8",
                "carRight": "2",
                "carGroup": "1"
            }
        ],
        "infraredFaceCodes": [],
        "carportAmount": "1",
        "fingerPrints": [],
        "cards": [
            {
                "cardState": "0",
                "masterFlag": "1",
                "duressFlag": "0",
                "cardNo": "00666998",
                "changeDate": "1560700800",
                "issueDate": "1560700800"
            }
        ],
        "facePictures": [
            "/9j/4AAQSk......UAf/9k="
        ]
    }
}
```


Parameters:

Parameter Name	Type	Description
details	object	User details
- companyName	string	Company name
- expireTime	string	End time of time stamp
- position	string	Position
- nickName	string	Nick name (used for address book distribution)
- useTimes	string	200 by default
- remark	string	Notes
- tel	string	Phone number
- email	string	Email
- idType	string	ID type, 0—ID card, 1—military officer ID card, 2—student ID card, 3—driving license, 4—passport, 5—tax identification number

Parameter Name	Type	Description
- maritalStatus	string	Marital status, 0—confidential, 1—married, 2—single
- birthday	string	Birthday
- idNum	string	ID number
- address	string	Address
- degree	string	Degree, 0—no, 1—primary school, 2—junior middle school, 3—senior middle school, 4—Bachelor, 5—Master, 6—Doctor, 7—Professor
- nationalityId	string	Reference nationality ID
- initialTime	string	Start time of time stamp
baseInfo	object	Basic information of user
- authority	string	Administrator, 0—no, 1—yes
- householder	string	Householder, 0—no, 1—yes
- unitId	string	Unit
- id	string	Unique constraint on id recorded in database
- pictureData	string	Picture data, face picture
- lastName	string	Last name
- personId	string	Person ID
- firstName	string	First name
- departmentId	string	Department ID
- cardType	string	Card type, 0—common, 1—blacklist, 2—visitor, 3—patrol, 4—VIP
- gender	string	Gender, 0—female, 1—male
- status	string	Status, 0—on the job, 1—quit
- stageId	string	Stage, only domestic version of express is valid.
- roomId	string	Room ID
- buildingId	string	Building ID
accessRight	object	User-associated permission information
- entranceDeviceCode	object	Entrance/exit device code
- channelId	object	Device channel id under user control
- doorGroupId	object	Door group id under user control
authentication	object	User-related authentication, access card and fingerprint information. One card and multiple fingerprints can be associated for a user.
- password	string	Password, AES encryption
- cars	object	Cars information
- - carNo	string	License plate number
- - carRight	string	Car permission, 1—blacklist, 2—whitelist; 2 by default
- - carGroup	string	Car group, 1—no, 3—common, 4—VIP; 1 by default
- infraredFaceCodes	object	Face feature
- carportAmount	string	Carport amount for user

Parameter Name	Type	Description
- fingerPrints	object	Fingerprint information
- - name	string	Fingerprint name
- - fingerPrint	string	Fingerprint code/600 bits
- - threadFlag	string	Duress or not, 0—no, 1—yes
- - masterFlag	string	Master fingerprint or not, 0—no, 1—yes
- cards	object	Cards information
- - cardState	string	Card state, -1—unknown, 0—normal, 1—reported for loss, 2—canceled, 4—frozen, 8—arrearage, 16—overdue, 32—arrearage in advance
- - masterFlag	string	Master card or not (Generation 1 card), 0—no, 1—yes
- - duressFlag	string	Duress or not, 0—no, 1—yes
- - cardNo	string	Card No., length 8 or 16, range 0-9,A-F
- - changeDate	string	Change date
- - issueDate	string	Card issue date
- facePictures	object	Face picture _ base64

Return example

```
{
    "code": 1000,
    "desc": "Success"
}
OR Repeat car num
{
    "code": 10004,
    "desc": "The carNo. is alresdy exists or repeat!",
    "data": {
        "repeatCarNos": [],
        "existingCarNos": ["A2563D8"]
    }
}
```

Description of return parameters

Parameter Name	Type	Description
code	int	Error Code
desc	string	Result description
data	object	None
- repeatCarNos	object	Repeat car num
- existingCarNos	object	Exist car num

Notes

– For more returned error codes, see the error code description on the home page

3.3.3、Delete one-card user in batches

Brief description:

- Delete one-card user in batches

Request URL:

- /OBMS/accessControl/person?personIds={personIds}
- X-Subject-Token: {Token}

Request example

```
DELETE /OBMS/accessControl/person?personIds=1123,2077 HTTP/1.1
Host: 10.33.68.138
Connection: close
Content-Type: application/json;charset=UTF-8
X-Subject-Token: 1b90ee2b0a5f4f7cb44149bbbb5f973f
```

Parameter Name	Type	Description
personIds	string	Person ID, separated by comma for multiple IDs, such as: 1123,2077
Token	string	The token obtained when logging in

Request method:

- DELETE

Request example

```
Empty
```

Parameters:

None

Return example

```
{
    "code": 1000,
    "desc": "Success"
}
```

Description of return parameters

Parameter Name	Type	Description
code	int	Error Code
desc	string	Result description

Notes

– For more returned error codes, see the error code description on the home page

3.3.4、Search one-card user list

Brief description:

- Search one-card user list.

Request URL:

- /OBMS/accessControl/personList?key={key}&deparmentId={deparmentId}&channelId={channelId}&personType={personType}&orderType={orderType}&direction={direction}&page={page}&pagesize={pagesize}
- X-Subject-Token: {token}

Request example

```
GET /OBMS/accessControl/personList?key=liu&deparmentId=001&channelId=1000006$7$0$0&personType=0&orderType=0&direction=0&page=1&pagesize=20 HTTP/1.1
Host: 10.33.68.138
Connection: close
Content-Type: application/json;charset=UTF-8
X-Subject-Token: d725f34e4f034038bf1771a8ea43c931
```

Parameter Name	Type	Description
key	string	Search keyword
deparmentId	int	Department ID, such as 001
channelId	string	Channel ID
personType	int	Person type, 0—common, 1—blacklist, 2—visitor, 3—patrol, 4—VIP
orderType	int	Sort field, 0—person ID (PERSON_ID), 1—first name (FIRST_NAME), 2—last name (LAST_NAME), 3—gender (GENDER), 4—person type (PERSON_TYPE), 5—stage ID (STAGE_ID), 6—building ID (BUILDING_ID), 7—unit ID (UNIT_ID), 8—room ID(ROOM_ID)
direction	int	Sort order, 0—ascending order, 1—descending order
page	int	First page of search, min.1
pagesize	int	Record count per page, min.1 and max.256
token	string	The token obtained when logging in

Request method:

- GET

Request example

```
Empty
```

Parameters:

None

Return example


```
{
    "code": 1000,
    "desc": "Success",
    "data": {
        "pageData": [
            {
                "id": "3",
                "personId": "2047",
                "stageId": "",
                "buildingId": "12",
                "unitId": "1",
                "roomId": "36",
                "firstName": "test",
                "lastName": " ",
                "departmentId": "001001",
                "gender": "1",
                "cardType": "0",
                "status": "0",
                "hasFinger": "0",
                "authority": "0",
                "cardNo": [
                    "00666998"
                ],
                "cars": [
                    "zA2563D8"
                ],
                "infraredFaceCodes": ["1"],
                "headPicUrl": ""
            }
        ],
        "nextPage": "-1",
        "totalCount": "1"
    }
}
```


Description of return parameters

Parameter Name	Type	Description
code	int	Error Code
desc	string	Result description
data	object	None

Parameter Name	Type	Description
- pageData	object	None
- - id	string	ID
- - personId	string	Person ID
- - stageId	string	Stage ID
- - buildingId	string	Building ID
- - unitId	string	Unit
- - roomId	string	Room ID
- - firstName	string	First name
- - lastName	string	Last name
- - departmentId	string	Department code
- - gender	string	Gender, 0—female, 1—male
- - cardType	string	Person type, 0—common, 1—blacklist, 2—visitor, 3—patrol, 4—VIP
- - status	string	Status, 0—on the job, 1—quit
- - hasFinger	string	Fingerprint or not, 0—no, 1—yes
- - authority	string	Administrator or not, 0—no, 1—yes
- - cardNo	object	Card No.
- - cars	object	License plate number
- - infraredFaceCodes	object	Has face feature code or not
- - headPicUrl	string	Head picture URL
- nextPage	string	Next page, -1 means no next page
- totalCount	string	Total count

Notes

– For more returned error codes, see the error code description on the home page

3.3.5、Search one-card user

Brief description:

- Search one-card user.

Request URL:

- /OBMS/accessControl/person/{personId}
- X-Subject-Token: {token}

Request example

```
GET /OBMS/accessControl/person/2047 HTTP/1.1
Host: 10.33.68.138
Connection: close
Content-Type: application/json;charset=UTF-8
X-Subject-Token: d725f34e4f034038bf1771a8ea43c931
```

Parameter Name	Type	Description
personId	int	Person ID
token	string	The token obtained when logging in

Request method:

- GET

Request example

```
Empty
```

Parameters:

None

Return example


```
{
    "code": 1000,
    "desc": "Success",
    "data": {
        "baseInfo": {
            "id": 32,
            "personId": "2047",
            "stageId": "",
            "buildingId": "12",
            "unitId": "1",
            "roomId": "36",
            "firstName": "test",
            "lastName": " ",
            "departmentId": "001001",
            "pictureData": "",
            "gender": "1",
            "cardType": "2",
            "status": "0",
            "householder": "0",
            "authority": "0"
        },
        "details": {
            "tel": "13611111111",
            "email": "joao@joao.com",
            "idType": "5",
            "idNum": "",
            "maritalStatus": "1",
            "nationalityId": "brazil",
```

```
            "birthday": "2019-06-17",
            "degree": "4",
            "initialTime": "1560700800",
            "expireTime": "1877615999",
            "address": "city",
            "remark": "test",
            "useTimes": "200",
            "nickName": "test",
            "position": "test",
            "companyName": "test"
        },
        "authentication": {
            "password": "9bb8291f7f9426004b3a1586c78244e4",
            "carportAmount": "1",
            "cards": [
                {
                    "cardNo": "00666998",
                    "issueDate": "1560753819",
                    "cardState": "0",
                    "changeDate": "1560759729",
                    "duressFlag": "0",
                    "masterFlag": "1"
                }
            ],
            "cars": [
                {
                    "carNo": "zA2563D8",
                    "carRight": "2",
                    "carGroup": "1"
                }
            ],
            "fingerPrints": [
                {
                    "name": "f009",
                    "masterFlag": "1",
                    "threadFlag": "0",
                    "fingerPrint": "..."
                }
            ],
            "facePictures": [
                "upload/obms/facePic/2047@15607597291720.jpg"
            ],
            "infraredFaceCodes": [
                "..."
```

```
            ]
        },
        "accessRight": {
            "channelIds": [
                "1000006$7$0$0"
            ],
            "doorGroupIds": [
                "1"
            ],
            "entranceDeviceCodes": [
                "1000001"
            ],
            "departmentDoorGroupIds": [
                "3"
            ]
        }
    }
}
```

Description of return parameters

Parameter Name	Type	Description
code	int	Error Code
desc	string	Result description
data	object	None
- baseInfo	object	Basic information of user
- - id	string	ID
- - personId	string	Person ID
- - stageId	string	Stage, only domestic version of express is valid.
- - buildingId	string	Building ID
- - unitId	string	Unit
- - roomId	string	Room ID
- - firstName	string	First name
- - lastName	string	Last name
- - departmentId	string	Department ID
- - pictureData	string	Picture data, face picture
- - gender	string	Gender, 0—female, 1—male
- - cardType	string	Card type, 0—common, 1—blacklist, 2—visitor, 3—patrol, 4—VIP
- - status	string	Status, 0—on the job, 1—quit
- - householder	string	Householder, 0—no, 1—yes
- - authority	string	Administrator, 0—no, 1—yes
- details	object	User details
- - tel	string	Phone number
- - email	string	Email

Parameter Name	Type	Description
- - idType	string	ID type, 0—ID card, 1—military officer ID card, 2—student ID card, 3—driving license, 4—passport, 5—tax identification number
- - idNum	string	ID number
- - maritalStatus	string	Marital status, 0—confidential, 1—married, 2—single
- - nationalityId	string	Reference nationality ID
- - birthday	string	Birthday
- - degree	string	Degree, 0—no, 1—primary school, 2—junior middle school, 3—senior middle school, 4—Bachelor, 5—Master, 6—Doctor, 7—Professor
- - initialTime	string	Start time(second)
- - expireTime	string	Expiration time(second)
- - address	string	Address
- - remark	string	Notes
- - useTimes	string	200 by default
- - nickName	string	Nick name (used for address book distribution)
- - position	string	Position
- - companyName	string	Company name
- authentication	object	User-related authentication, access card and fingerprint information. One card and multiple fingerprints can be associated for a user.
- - password	string	Password, AES encryption
- - carportAmount	string	Carport amount for user
- - cards	object	Cards information
- - - cardNo	string	Card No.
- - - issueDate	string	Card issue date
- - - cardState	string	Card state, –1—unknown, 0—normal, 1—reported for loss, 2—canceled, 4—frozen, 8—arrearage, 16—overdue, 32—arrearage in advance
- - - changeDate	string	Change date
- - - duressFlag	string	Duress or not, 0—no, 1—yes
- - - masterFlag	string	Master card or not (Generation 1 card), 0—no, 1—yes
- - cars	object	Cars information
- - - carNo	string	License plate number
- - - carRight	string	Car permission, 1—blacklist, 2—whitelist; 2 by default
- - - carGroup	string	Car group, 1—no, 3—common, 4—VIP; 1 by default
- - fingerPrints	object	Fingerprint information
- - - name	string	Fingerprint name
- - - masterFlag	string	Master fingerprint or not, 0—no, 1—yes
- - - threadFlag	string	Duress or not, 0—no, 1—yes
- - - fingerPrint	string	Fingerprint code/600 bits
- - facePictures	object	Face picture
- - infraredFaceCodes	object	Face feature code
- - accessRight	object	User-associated permission information
- - - channelIds	object	Device channel id under user control

Parameter Name	Type	Description
- - - doorGroupIds	object	Door group id under user control
- - - entranceDeviceCodes	object	Entrance/exit device code
- - - departmentDoorGroupIds	object	Department associate with door group id

Notes

– For more returned error codes, see the error code description on the home page

4、Dictionary Definition

4.1、Manufacture type

Value	Type
1	Intelbras
2	Hikvision
3	Others

4.2、Device type

Device category		Device subcategory	
Value	Type	Value	Type
1	Coding device	1	DVR
		2	IPC
		3	NVS
		6	NVR
		10	EVS
		21	VTT
		26	Thermal imaging
		34	Full view PTZ
		35	MCS
		43	IVSS
5	ANPR	1	ANPR device
7	Intelligent device	4	IVS-7500
8	A&C	1	A&C controller
		2	A&C centralized controller
		3	Face gate
		5	Second-generation A&C
		6	Infrared face clocking terminal
21	Video intercom	1	VTNC
		2	Unit VTO (unit gate)
		3	VTH (indoor)
		4	MVTH

		5	Fence VTO (wall)
		6	LOCKVTH
		7	2nd VTO (secondary confirmation)
		8	VTS (management)
		10	SIP PHONE (SIP phone)
34	Emergency device	1	Alarm pillar
		2	Alarm box
45	Monitor control device	2	NVD (decoder)
		4	Large screen device
		6	Matrix switcher

4.3、Unit type

Value	Type
1	Encoding unit
2	Decoding output
3	Alarm input
4	Alarm output
7	A&C unit
8	Voice unit
11	POS unit
12	Virtual unit, used for a single radar
15	LED unit
16	CVI alarm unit
17	Alarm box subunit

4.4、Camera Type

Value	Type
1	Bullet camera
2	Speed dome camera
3	Dome camera

4.5、Capability set for coding channel

Value	Type
1	Intelligent alarm
2	Fisheye dewarp
3	Electric focus
4	Infrared temperature measurement
5	Heat map analysis
6	Crossing line people count
7	Regional people count
8	Face snapshot

Value	Type
9	Face recognition
10	Object snapshot
11	Master-slave tracking
14	Multi-region people count - used for AI fisheye multi-region count
15	Smart track

4.6、Access Control Device Type

Value	Type
1	First generation of general access control device type
2	First generation of centralized controller type
3	First generation of face recognition terminals type
5	Second generation of access control device type

4.7、Access Control Event Type

Event Type		Event subtype	
Value	Type	Value	Type
5	Normal access control	42	Legal password unlocking
		45	Legal fingerprint unlocking
		48	Remote unlocking (indoor monitor/platform)
		49	Normal button unlocking
		51	Legal card swiping
		56	Normal locking
		57	Normal unlocking
		13109	Patrol user
		13123	Combination unlocking by card and password
		13124	Combination unlocking by card and fingerprint
		13125	Remote verification
		600005	Legal face unlocking
		600013	Combination unlocking by card and face
		700000	Combination unlocking by fingerprint and password
		700001	Combination unlocking by fingerprint and face
		700002	Combination unlocking by face and password
		700003	Combination unlocking by card, fingerprint, and password
		700004	Combination unlocking by card, fingerprint, and face
		700005	Combination unlocking by fingerprint, face, and password
		700006	Combination unlocking by card, face, and password
		700007	Combination unlocking by card, fingerprint, face, and password
15	Abnormal access control	43	Illegal password unlocking
		46	Illegal fingerprint unlocking
		52	Illegal card swiping

		54	Abnormal unlocking
		55	Abnormal locking
		13100	No permission (card/password/fingerprint/face)
		13101	Card missing or frozen
		13103	Unlocking mode error (the user is authorized but the locking mode does not match)
		13104	Validity period error
		13106	Duress alarm not opened
		13107	Door always closed unlocking
		13108	Multi-door interlock (can open up to one door)
		13111	Time period error
		13112	Open time error during holidays
		13113	Not the first card (the first card everyday)
		13114	Correct card plus wrong password
		13115	Correct card plus password timeout
		13116	Correct card plus wrong fingerprint
		13117	Correct card plus fingerprint timeout
		13118	Correct fingerprint plus wrong password
		13119	Correct fingerprint plus password timeout
		13120	Wrong sequence of the combination for gate opening
		13121	Combination unlocking with further verification
		13122	Verification passed, console unauthorized
		600011	Correct card plus face timeout
		600012	Correct card plus face error
16	Access control alarms	41	Duress alarm
		72	Timeout alarm
		1433	Restricted list alarm
		1446	Alarm of excessive use of illegal card
		13105	Anti-passback alarm
		13110	Intrusion alarm
		600003	Card reader tamper alarm
25	Access control device alarm	600004	Device tamper alarm