NABLA WORKS

Version: 2024/05/20

SenseNebula AIE
HTTP API Reference

Document Number: Nebula-PMO-IR001

Document Version No.: 20240520

Product Name: SenseNebula AIE LT-A

Product Models:
ST-SNMA-SX04MV
ST-SNMA-SX08MV

System Version: V2.4.0

Contents

About This Guide

Overview

This document is applicable to SenseNebula AIE LT-A (hereinafter referred to as SenseNebula AIE) and contains the following parts: Product Overview, Hardware Preparation, and Procedure.

Intended Audience

This document is intended for the following audience:

- Pre-sales engineers
- Technical support engineers

Copyright

Trademarks and Permissions

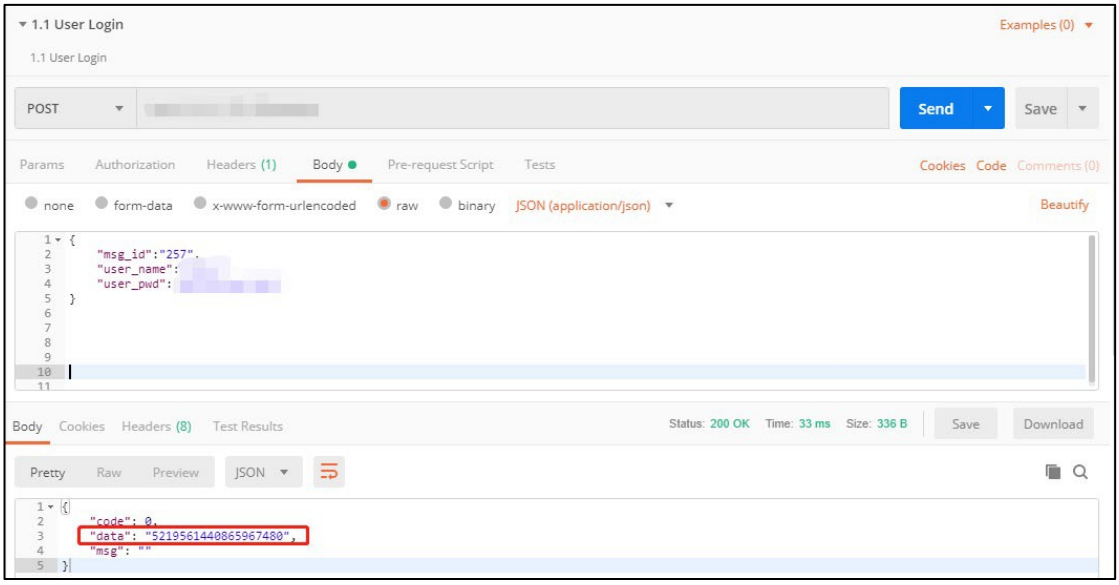
Disclaimer

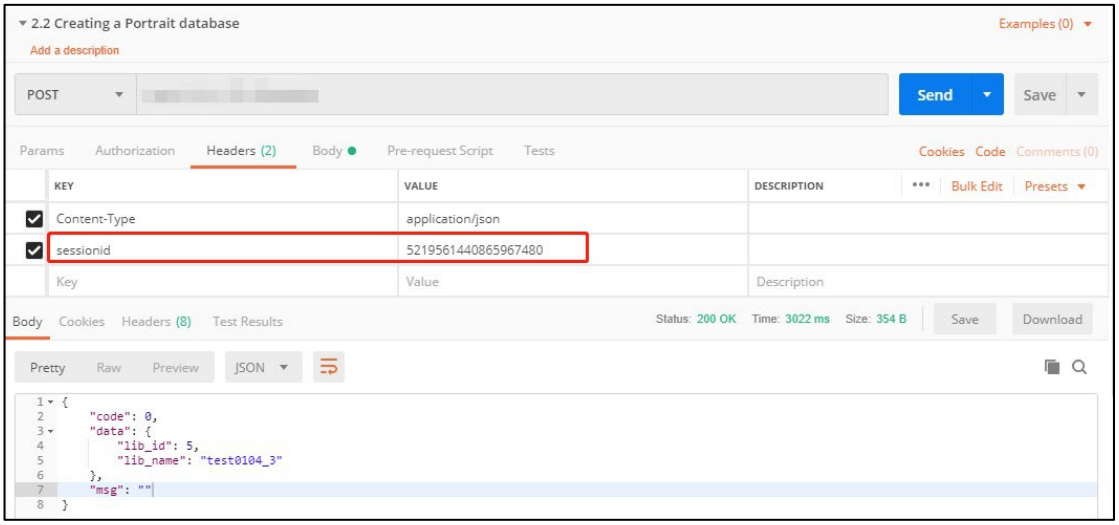
Revisions

Document Version	System Version	Release Date	Description
01	V2.0.0	2019-12-20	First official release
02	V2.0.1	2020-01-20	Modified based on software version V2.0.1.
03	V2.0.2	2020-04-30	Modified based on software version V2.0.2.
04	V2.1.0	2020-06-23	Modified based on software version V2.1.0.
05	V2.1.0	2020-07-31	Modified based on software version V2.1.0.
06	V2.1.1	2020-11-06	Modified based on software version V2.1.1.
07	V2.1.2	2021-03-05	Modified based on software version V2.1.2. Add stranger clustering, wandering alarm analysis, network relay and so on.
08	V2.1.3	2021-04-20	Modified based on software version V2.1.3. Add storage mode for portrait database and bugfix for V2.1.2.
09	V2.1.6	2021-08-21	Modified based on software version V2.1.6. Add non-motor vehicle recognition and comply with privacy and security regulation.
10	V2.2.0	2021-10-31	Modified based on software version V2.2.0. Add liveness recognition.
11	V2.2.2	2021-12-17	Modified based on software version V2.2.2. Bug fixed edition.
12	V2.2.5	2022-01-26	Modified based on software version V2.2.5. Bug fixed edition.
13	V2.3.0	2022-05-25	Modified based on software version V2.3.0. Add pedestrian intrusion, crowd density, fence armed, staff leaving their post.
14	V2.4.0	2022-06-17	Modified based on software version V2.4.0. Add access to SenseLink Standard.

1 Instruction

1. The general format of the request is https:/${ip}: ${port}/ api/json with port 443..
2. When switching to http, the interface URL is changed to http://${ip}:${port}/api/json with port 80.
3. Sessionid will be returned after user logining. All API requests need to include “sessionid” in the header of http request. If the user does not do anything 30 minutes after logining, the sessionid will expire automatically and the user needs to login again.
4. When 3 consecutive login failures occur, you need to enter the verification code, and the system will automatically prompt that the number of login errors exceeds the limit, and your account will be locked if there are 2 attempts left. When 5 consecutive login failures occur, the system will be locked for 5 minutes. Then it will be unlocked automatically and you can try to log in again.
5. Https mode is enabled by default, and you need to use the https to request API interface.
6. All sensitive information in query information is displayed as ******. To get the complete original information, you need to request a single query API for details.





2 User management interfaces

2.1 User login

Interface description

Interface url	http://${ip}:${port}/api/json
Request Method	POST
Request Parameter Format	JSON
Interface Description	Login the system

Request parameter

Parameter	Type	Description	Required	Example
msg_id	string	“257”	Yes	
user_name	string	User name, support capital letters, lowercase letters, numbers, special characters (@ and .-_), its length should be less than 50	Yes	
user_pwd	string	User password, must contain three types of following characters: capital letters, lowercase letters, special characters and numbers, and the password length should be greater than or equal to 8 and less than 32.	Yes	
verify_code	string	Verify Code. Using for 3 logining failed.	No	

Response parameter

Parameter	Type	Description
code	int	Result code (0:successful)

data	json/string	sucess:User sessionid(string), all API requests need to include “sessionid” in the header of http request. If the user does not do anything 30 minutes after logining, the sessionid will expire automatically and the user needs to login again.
msg	string	Result description

Field information(data)

Parameter	Type	Description
session_id	string	session identification
retry_times	int	user remain login times
mod_passwd_flag	int	action flag

Example

Request example

```
{
  "msg_id":"257",
  "user_name":"admin",
  "user_pwd":"newpassword"
}
```

Response example

```
{
  "code": 0,
  "data": "1287281518932891256",
  "msg": ""
}
```

2.2 User logout

Interface description

Interface url	http://${ip}:${port}/api/json
Request Method	POST
Request Parameter Format	JSON
Interface Description	Logout the system

Request parameter

Parameter	Type	Description	Required	Example
msg_id	string	“258”	Yes	

Response parameter

Parameter	Type	Description
code	int	Result code (0:successful)
data	string	NULL
msg	string	Result description

Example

Request example

```
{
   "msg_id":"258"
}
```

Response example

```
{
   "code": 0,
   "data": "",
   "msg": ""
}
```

2.3 Request verification code

Interface description

Interface url	http://${ip}:${port}/api/json
Request Method	POST
Request Parameter Format	JSON
Interface Description	Request verification code

Request parameter

Parameter	Type	Description	Required	Example
msg_id	string	“1322”	Yes	

Response parameter

Parameter	Type	Description
code	int	Result code (0:successful)
data	string	Verification code string
msg	string	Result description

Example

Request example

```
{
```

```
  "msg_id": "1322"
}
```

Response example

```
{
  "code": 0,
  "data": "9865",
  "msg": ""
}
```

2.4 Set user authorization information

Interface description

Interface url	http://${ip}:${port}/api/json
Request Method	POST
Request Parameter Format	JSON
Interface Description	Set user authorization information

Request parameter

Parameter	Type	Description	Required	Example
msg_id	string	“269”	Yes	
user_name	string	User name	Yes	
is_authed	int	Is authorized, 1:Yes, 0:No	Yes	

Response parameter

Parameter	Type	Description
code	int	Result code (0:successful)
data	string	NULL
msg	string	Result description

Example

Request example

```
{
  "msg_id":"269",
  "user_name": "userA",
  "is_authed":0
}
```

Response example

```
{
  "code": 0,
  "data": "",
```

```
  "msg": ""
}
```

2.5 Query user authorization information

Interface description

Interface url	http://${ip}:${port}/api/json
Request Method	POST
Request Parameter Format	JSON
Interface Description	Query user authorization information

Request parameter

Parameter	Type	Description	Required	Example
msg_id	string	“270”	Yes	
user_name	int	User name	Yes	

Response parameter

Parameter	Type	Description
code	int	Result code (0:successful)
data	string	NULL
msg	string	Result description

Field information(data)

Parameter	Type	Description
user_name	string	User name
is_authed	int	Is authorized
auth_items	int	Authorization items

Example

Request example

```
{
  "msg_id":"270"
}
```

Response example

```
{
  "code": 0,
  "data": {
    "user_name": "userA",
    "is_authed": 0,
    "auth_items": 0
  },
```

```
  "msg": ""
}
```

2.6 User password authentication

Interface description

Interface url	http://${ip}:${port}/api/json
Request Method	POST
Request Parameter Format	JSON
Interface Description	User password authentication

Request parameter

Parameter	Type	Description	Required	Example
msg_id	string	“271”	Yes	
username	string	User name	No	
userpasswd	string	User passwd	Yes	

Response parameter

Parameter	Type	Description
code	int	Result code (0:successful)
data	string	Results and user name
msg	string	Result description

Example

Request example

```
{
  "msg_id":"271",
        "userpasswd":"123456"
}
```

Response example

```
{
  "code": 0,
  "data": {
    "result": 0,
    "username": "userA",
  },
  "msg": ""
}
```

2.7 Set User Priveleges

Interface description

Interface url	http://${ip}:${port}/api/json
Request Method	POST
Request Parameter Format	JSON
Interface Description	Set User Priveleges

Request parameter

Parameter	Type	Description	Required	Example
msg_id	string	“272”	Yes	
user_name	string	User name	Yes	
right_item	int	The type of User Priveleges, 1: Image View Right	Yes	
right_value	int	The Right Value, 1:Yes, 0:No	Yes	

Response parameter

Parameter	Type	Description
code	int	Result code (0:successful)
data	string	Results and user right info
msg	string	Result description

Example

Request example

```
{
  "msg_id":"272", "user_name":
  "admin", "right_item":1,
          "right_value":1
}
```

Response example

```
{
  "code": 0,
  "data": "",
  "msg": ""
}
```

2.8 Query User Priveleges

Interface description

Interface url	http://${ip}:${port}/api/json
Request Method	POST
Request Parameter Format	JSON
Interface Description	Query User Priveleges

Request parameter

Parameter	Type	Description	Required	Example
msg_id	string	“273”	Yes	
user_name	int	User name	Yes	
right_item	int	The type of User Priveleges, 1: Image View Right	Yes	

Response parameter

Parameter	Type	Description
code	int	Result code (0:successful)
data	string	user right info
msg	string	Result description

Field information(data)

Parameter	Type	Description
user_name	string	User name
right_item	int	The type of User Priveleges, 1: Image View Right
right_value	int	The Right Value, 1:Yes, 0:No

Example

Request example

```
{
  "msg_id":"273",  "right_item":1
}
```

Response example

```
{
  "code": 0,
  "data": {
                    "user_name": "admin",
```

```
                    "right_item":1,
                    "right_value":1
  },
  "msg": ""
}
```

3 Portrait database-related interfaces

3.1 Querying all portrait databases

Interface description

Interface url	http://${ip}:${port}/api/json
Request Method	POST
Request Parameter Format	JSON
Interface Description	Queries all portrait databases

Request parameter

Parameter	Type	Description	Required	Example
msg_id	string	“1028”	Yes	

Response parameter

Parameter	Type	Description
code	int	Result code (0:successful)
data	json	Returned data
msg	string	Result description

Field information(data)

Parameter	Type	Description
create_time	string	Creation time
lib_id	int	Database ID
lib_name	string	Database name
lib_type	int	Database type (1:key personnel database, 2:whitelist database)
picture_no	int	Number of images in the database
update_time	string	Update time

Example

Request example

```
{
  "msg_id":"1028"
}
```

Response example


```
{
  "code": 0,
  "data": [
    {
      "create_time": "2019-01-23 10:54:13",
      "lib_id": 1,
      "lib_name": "test01",
      "lib_type": 1,
      "picture_no": 6,
      "update_time": "2019-01-23 10:54:13"
    }
  ],
  "msg": ""
}
```


3.2 Creating a portrait database

Interface description

Interface url	http://${ip}:${port}/api/json
Request Method	POST
Request Parameter Format	JSON
Interface Description	Creates a portrait database.

Request parameter

Parameter	Type	Description	Required	Example
msg_id	string	“1025”	Yes	
lib_name	string	Name of the new database	Yes	
lib_type	int	Database type (1:key personnel database, 2:whitelist database)	Yes	2

Response parameter

Parameter	Type	Description
code	int	Result code (0:successful)
data	json	Returned data

msg	string	Result description

Field information(data)

Parameter	Type	Description
lib_id	int	ID of the new database
lib_name	string	Name of the new database

Example

Request example

```
{
   "msg_id":"1025",
   "lib_name":"test",  "lib_type":1
}
```

Response example

```
{
   "code": 0,
   "data": {
      "lib_id": 2,
      "lib_name": "test"
   },
   "msg": ""
}
```

3.3 Deleting a portrait database

Interface description

Interface url	http://${ip}:${port}/api/json
Request Method	POST
Request Parameter Format	JSON
Interface Description	Deletes the specified portrait database.

Request parameter

Parameter	Type	Description	Required	Example
msg_id	string	“1026”	Yes	
lib_id	int	ID of the database to be deleted	Yes	

Response parameter

Parameter	Type	Description
code	int	Result code (0:successful)
data	json	NULL

msg	string	Result description

Example

Request example

```
{
   "msg_id":"1026",
   "lib_id":23
}
```

Response example

```
{
   "code":0,
   "data":"",
   "msg":""
}
```

3.4 Modifying a portrait database

Interface description

Interface url	http://${ip}:${port}/api/json
Request Method	POST
Request Parameter Format	JSON
Interface Description	Modifies the specified portrait database.

Request parameter

Parameter	Type	Description	Required	Example
msg_id	string	“1027”	Yes	
lib_id	int	ID of the database to be modified	Yes	
lib_name	string	Database name	Yes	
lib_type	int	Database type (1:key personnel database, 2:whitelist database)	Yes	1

Response parameter

Parameter	Type	Description
code	int	Result code (0:successful)
data	json	Returned data
msg	string	Result description

Field information(data)

Parameter	Type	Description
lib_id	int	Database ID
lib_name	string	Database name

Example

Request example

```
{
  "msg_id":"1027",
  "lib_id":1,
  "lib_name":"test",
  "lib_type":1
}
```

Response example

```
{
  "code": 0,
  "data": {
    "lib_id": 1,
    "lib_name": "test"
  },
  "msg": ""
}
```

3.5 Querying target databases based on criteria

Interface description

Interface url	http://${ip}:${port}/api/json
Request Method	POST
Request Parameter Format	JSON
Interface Description	Queries portrait databases based on criteria.

Request parameter

Parameter	Type	Description	Required	Example
msg_id	string	“1034”	Yes	
start_no	int	Start position	No	Default value: 1
qry_len	int	Number of queried items; maximum value:50	No	Default value:10
lib_id	int	Database ID	No	
lib_name	string	Database name	No	

lib_type	int	Database type (1:key personnel database, 2:whitelist database)	No	1

Response parameter

Parameter	Type	Description
code	int	Result code (0:successful)
data	json	Returned data
msg	string	Result description

Field information(data)

Parameter	Type	Description
record	json list	Returned results
result_num	int	Number of query results
total_num	int	Total number of records

Field information(record)

Parameter	Type	Description
create_time	string	Creation time
lib_id	int	Database ID
lib_name	string	Database name
lib_type	int	Database type (1:key personnel database, 2:whitelist database)
picture_no	int	Number of images
update_time	string	Update time

Note: By default, portrait databases are sorted by creation time.

Example

Request example

```
{
  "msg_id":"1034",
  "start_no":1,
  "qry_len":1
}
```

Response example

```
{
  "code": 0,
```

```
  "data": {
    "record": [
      {
        "create_time": "2019-01-23 10:54:13",
        "lib_id": 1,
        "lib_name": "test",
        "lib_type": 1,
        "picture_no": 6,
        "update_time": "2019-01-23 10:54:13"
      }
    ],
    "result_num": 1,
    "total_num": 1
  },
  "msg": ""
}
```


3.6 Querying a single face image (Sensitive information is encrypted)

Interface description

Interface url	http://${ip}:${port}/api/json
Request Method	POST
Request Parameter Format	JSON
Interface Description	Queries a single face image

Request parameter

Parameter	Type	Description	Required	Example
msg_id	string	“1032”	Yes	
img_id	string	ID of the face image to be queried	Yes	
lib_id	int	ID of the database to be queried	Yes	

Response parameter

Parameter	Type	Description
code	int	Result code (0:successful)
data	json	Returned data
msg	string	Result description

Field information(data)

Parameter	Type	Description
create_time	string	Creation time
img	string	Image (base64-transcoded)
img_feat	string	Image features
img_id	string	Face image ID
img_path	string	Image path
lib_id	int	ID of the database that stores the face image
person_addr	string	Address
person_age	string	Age, between 0 and 999
person_gender	string	Gender (0:female, 1:male)
person_idcard	string	Identity code
person_name	string	Name

Example

Request example

```
{
  "msg_id":"1032",
  "lib_id":8,
  "img_id":"test"
}
```

Response example

```
{
  "code": 0,
  "data": {
    "create_time": "2019-01-23 11:34:52",
    "img":  "\ufffd\uffff\u001c\ufffd\u0014Q_-\ufffd\ufffdfdfd",
    "img_feat": "2442,501,111,179,482,239,414,486,49,111,143,169,192,247",
    "img_id": "test",
    "img_path": "img/1_a2d7ae51-f551-43c0-9bf5-af12ff32c630.png",  "lib_id":
    1,
    "person_addr": "*****",
    "person_age": "14",
    "person_gender": "1",
    "person_idcard": "*****",
    "person_name": "renlian1"
  },
  "msg": ""
}
```

3.7 Storing a single face image in a database

Interface description

Interface url	http://${ip}:${port}/api/form
Request Method	POST
Request Parameter Format	FORM
Interface Description	Adds a single face image to the specified face database.

Request parameter

Parameter	Type	Description	Required	Example
msg_id	string	"1029"	Yes	
lib_id	int	ID of the database in which the face image will be stored	Yes	
img	file	Face image to be stored	Yes	
img_id	string	ID of the face image to be stored, Identifier of the image. It can be customized and must be unique in the database. It may contain English letters, numbers, and hyphens (-) and is up to 48 characters in length.	No	
person_idcard	string	Identity code	No	
person_name	string	Name	No	
person_gender	string	Gender (0:female, 1:male)	No	
person_age	string	Age, between 0 and 999	No	
person_addr	string	Address	No	

Response parameter

Parameter	Type	Description
code	int	Result code (0:successful)
data	json	Returned data
msg	string	Result description

Field information(data)

Parameter	Type	Description
img_id	string	Image ID

Note 1: Image files only support the jpg / jpeg / png/bmp/tif formats.

Note 2: img_id uniquely identifies a face image and cannot be repeated. If it is not specified or is empty, an image ID will be automatically allocated.

Example

Request example

```
Input Files
{
  'img':  name='15427121547955542.jpg'
}
  "img_id":"test",
  "lib_id":11,
  "msg_id":"1029"
```

Response example

```
{
  "code": 0,
  "data": {
    "img_id": "test"
  },
  "msg": ""
}
```

3.8 Deleting a single face image

Interface description

Interface url	http://${ip}:${port}/api/json
Request Method	POST
Request Parameter Format	JSON
Interface Description	Deletes a single face image.

Request parameter

Parameter	Type	Description	Required	Example
msg_id	string	“1031”	Yes	

img_id	string	Image ID	Yes	
lib_id	int	ID of the face database where the image is stored	Yes	

Response parameter

Parameter	Type	Description
code	int	Result code (0:successful)
data	json	NULL
msg	string	Result description

Example

Request example

```
{
  "msg_id":"1031",
  "img_id":"test_3",
  "lib_id":8
}
```

Response example

```
{
  "code": 0,
  "data": "",
  "msg": ""
}
```

3.9 Modifying the attributes of a single face image

Interface description

Interface url	http://${ip}:${port}/api/json
Request Method	POST
Request Parameter Format	JSON
Interface Description	Modifies the attributes of a face image in the specified face database.

Request parameter

Parameter	Type	Description	Required	Example
msg_id	string	“1033”	Yes	
img_id	string	Image ID	Yes	
lib_id	int	ID of the face	Yes	

		database where the image is stored		
person_idcard	string	Identity code	No	
person_name	string	Name	No	
person_gender	string	Gender (0:female, 1:male)	No	
person_age	string	Age, between 0 and 999	No	
person_addr	string	Address	No	

Response parameter

Parameter	Type	Description
code	int	Result code (0:successful)
data	json	Returned data
msg	string	Result description

Field information(data)

Parameter	Type	Description
img_id	string	Image ID
lib_id	int	ID of the face database where the image is stored

Example

Request example

```
{
  "msg_id":"1033",
  "img_id":"test",
  "lib_id":8,
  "person_age":"90"
}
```

Response example

```
{
  "code": 0,
  "data": {  "img_id":
    "test",  "lib_id": 8
  },
  "msg": ""
}
```

3.10 Querying face images based on criteria

Interface description

Interface url	http://${ip}:${port}/api/json
Request Method	POST
Request Parameter Format	JSON
Interface Description	Queries images in the specified face database based on criteria.

Request parameter

Parameter	Type	Description	Required	Example
msg_id	string	“1035”	Yes	
lib_id	int	ID of the face database where the image is stored	Yes	
start_no	int	Start position	No	Default value: 1
qry_len	int	Number of queried items; maximum value: 50	No	Default value:10
create_time	string	Creation time	No	

Response parameter

Parameter	Type	Description
code	int	Result code (0:successful)
data	json	Returned data
msg	string	Result description

Field information(data)

Parameter	Type	Description
record	json list	Returned data
result_num	int	Number of query results
total_num	int	Total number of records

Field information(record)

Parameter	Type	Description
person_idcard	string	Identity code
person_addr	string	Address
person_age	string	Age, between 0 and 999
person_gender	string	Gender (0:female, 1:male)
img_id	string	Image ID

person_name	string	Name
img_path	string	Image path
lib_id	int	ID of the database that stores the face image
create_time	string	Creation time

Note: By default, face images are sorted by creation time.

Example

Request example

```
{
  "msg_id":"1035",
  "lib_id":1
}
```

Response example


```
{
  "code": 0,
  "data": {
    "record": [
      {
        "create_time": "2019-01-23 12:00:19",
        "img_id": "test_4",
        "img_path": "img/1_692198ba-e9ea-4ca3-bcf8-2b7a56fd734b.png",
        "lib_id": 1,
        "person_addr": "wuhan",
        "person_age": "14",
        "person_gender": "1",
        "person_idcard": "130182",
        "person_name": "wang"
      }
    ],
    "result_num": 1,
    "total_num": 1
  },
  "msg": ""
}
```


3.11 Decryption of sensitive information of a single face

Interface description

Interface url	http://${ip}:${port}/api/json
Request Method	POST
Request Parameter Format	JSON

Interface Description	Queries images in the specified face database based on criteria.

Request parameter

Parameter	Type	Description	Required	Example
msg_id	string	“1067”	Yes	
lib_id	int	ID of the face database where the image is stored	Yes	
lib_id	int	Image ID	Yes	

Response parameter

Parameter	Type	Description
code	int	Result code (0:successful)
data	json	Returned data
msg	string	Result description

Field information(data)

Parameter	Type	Description
record	json list	Returned data
result_num	int	Number of query results
total_num	int	Total number of records

Field information(record)

Parameter	Type	Description
person_idcard	string	Identity code
person_addr	string	Address
person_age	string	Age, between 0 and 999
person_gender	string	Gender (0:female, 1:male)
img_id	string	Image ID
person_name	string	Name
img_path	string	Image path
lib_id	int	ID of the database that stores the face image
create_time	string	Creation time

Note: By default, face images are sorted by creation time.

Example

Request example


```
{
  "msg_id":"1067",
  "lib_id":1,
  "uuid":"test_4"
}
```


Response example


```
{
  "code": 0,
  "data": {
    "record": [
      {
        "create_time": "2019-01-23 12:00:19",
        "img_id": "test_4",
        "img_path": "img/1_692198ba-e9ea-4ca3-bcf8-2b7a56fd734b.png",
        "lib_id": 1,
        "person_addr": "wuhan",
        "person_age": "14",
        "person_gender": "1",
        "person_idcard": "130182",
        "person_name": "wang"
      }
    ],
    "result_num": 1,
    "total_num": 1
  },
  "msg": ""
}
```


3.12 Exporting face images in pagination mode

Interface description

Interface url	http://${ip}:${port}/api/json
Request Method	POST
Request Parameter Format	JSON
Interface Description	Exports face images in pagination mode

Request parameter

Parameter	Type	Description	Required	Example
msg_id	string	“1038”	Yes	
lib_id	int	Database ID	Yes	

start_no	int	Start index	No	Default value: 1
qry_len	int	Number of exported items; maximum value:500	No	Default value:10

Response parameter

Parameter	Type	Description
code	int	Result code (0:successful)
data	json	Returned data
msg	string	Result description

Field information(data)

Parameter	Type	Description
export_url	string	Exported file path
result_num	int	Number of query results
total_num	int	Total number of records
zippasswd	string	Compressed password

Note: Do not include commas in the exported content, otherwise the export will fail.

Example

Request example

```
{
  "msg_id":"1038",
  "lib_id":1
}
```

Response example

```
{
  "code": 0,
  "data": {
    "export_url": "tmp/face_2019-01-23-12-04-27.zip",
    "result_num": 10,
    "total_num": 16,
                    "zippasswd":"xxxx"
  },
  "msg": ""
}
```

3.13 Exporting selected face images

Interface description

Interface url	http://${ip}:${port}/api/json
Request Method	POST
Request Parameter Format	JSON
Interface Description	Exports selected face images from the specified face database.

Request parameter

Parameter	Type	Description	Required	Example
msg_id	string	"1041"	Yes	
lib_id	int	Database ID	Yes	
img_ids	string	A set of image IDs, which can be obtained through query (Separate image IDs with commas [,]. The image quantity ranges from 1 to 50.)	Yes	

Response parameter

Parameter	Type	Description
code	int	Result code (0:successful)
data	json	Returned data
msg	string	Result description

Field information(data)

Parameter	Type	Description
export_url	string	Exported file path
failed_img_ids	string	ID of the image that cannot be exported
result_num	int	Number of query results
total_num	int	Total number of records
zippasswd	string	Compressed password

Note: Do not include commas in the exported content, otherwise the export will fail.

Example

Request example

```
{
   "msg_id":"1041",
   "lib_id":1,
   "img_ids":"test,test_4"
}
```

Response example

```
{
   "code": 0,
   "data": {
      "export_url": "tmp/face_2019-01-23-12-06-19.zip",
      "failed_img_ids": "",
      "result_num": 2,
      "total_num": 2,
                        "zippasswd":"xxxx"
   },
   "msg": ""
}
```

3.14 Querying capture records based on criteria

Interface description

Interface url	http://${ip}:${port}/api/json
Request Method	POST
Request Parameter Format	JSON
Interface Description	Queries captures based on criteria

Request parameter

Parameter	Type	Description	Required	Example
msg_id	string	“1037”	Yes	
channel	int	Channel; value range:[1,16] of SenseNebula-M4s, [1,32] of SenseNebula-M8s	No	
lib_name	string	Name of a face database	No	
lib_type	int	Database type (1:key personnel database, 2:whitelist database)	No	

start_time	string	Start time	No	yyyy-MM-dd HH:mm:ss
stop_time	string	End time	No, When start_time is filled in, stop_time must also be filled in	yyyy-MM-dd HH:mm:ss
start_no	int	Start position	No	Default value: 1
qry_len	int	Number of queried items; maximum value:50	No	Default value:10
alive_type	string	Alive type separate by comma, 0：Unknown; 1: Non alive; 2: Alive	No	Default value:0,1,2

Response parameter

Parameter	Type	Description
code	int	Result code (0:successful)
data	json	Returned data
msg	string	Result description

Field information(data)

Parameter	Type	Description
record	json	Returned data
result_num	int	Number of query results
total_num	int	Total number of records

Field information(record)

Parameter	Type	Description
camera_name	string	Camera name
channel	int	Channel; value range:[1,16] of SenseNebula-M4s, [1,32] of SenseNebula-M8s
lib_id	int	ID of the matched database
lib_name	string	Database name
lib_type	int	Database type (1:key personnel database, 2:whitelist database)
position	string	Camera position
ranking	int	1 indicates the highest ranking

		in the comparison results.
similarity	int	Similarity score; value range:[0,100]
snap_id	string	Capture ID
snap_path	string	Capture path
threshold	int	Threshold; value range:[0,100]
trigger	string	Capture time
alive_type	int	Alive type, 0： Unknown; 1: Non alive; 2: Alive
face_attr	json	Face capture attributes, for details, see the appendix B

Note: By default, captures are sorted by capture time.

Example

Request example

```
{
  "msg_id":"1037",
  "start_no":1,
  "qry_len":1
}
```

Response example

```
{
  "code": 0,
  "data": {
    "record": [
      {
        "camera_name": "no1",
        "channel": 3,
        "face_attr": {
          "cap_style": "hat_style_type_none",  "gender_code":
          "female",
          "glass_style": "glasses_style_type_none",
          "mustache_style": "mustache_style_type_none",
          "st_respirator": "st_respirator_mouth",
          "respirator_color": "color_type_none",   "st_age":
          "st_adult",
          "st_age_value": "26.000000",
          "st_expression": "st_angry"
        },
        "lib_id": 1,
        "lib_name": "test",
        "lib_type": 1,
```

```
            "position": "",
            "ranking": 1,
            "similarity": 32,
            "snap_id": "9448f6e5-0d8b-4fd0-a13f-98b1a05500ec",  "snap_path":
            "record/9448f6e5-0d8b-4fd0-a13f-98b1a05500ec.jpg",  "threshold":
            100,
            "alive_type": 2,
            "trigger": "2019-01-23 12:07:53"
        }
    ],
    "result_num": 1,
    "total_num": 105
  },
  "msg": ""
}
```


3.15 Exporting captures in pagination mode

Interface description

Interface url	http://${ip}:${port}/api/json
Request Method	POST
Request Parameter Format	JSON
Interface Description	Exports captures based on criteria.

Request parameter

Parameter	Type	Description	Required	Example
msg_id	string	“1040”	Yes	
start_no	int	Start index	No	Default value: 1
qry_len	int	Number of exported items; maximum value:500	No	Default value:10
channel	int	Channel; value range:[1,16] of SenseNebula-M4s, [1,32] of SenseNebula-M8s	No	
lib_name	string	Name of a face database	No	
lib_type	int	Database type (1:key personnel database,	No	1

		2:whitelist database)		
start_time	string	Start time	No	
stop_time	string	End time	No, When start_time is filled in, stop_time must also be filled in	
alive_type	string	Alive type separate by comma, 0：Unknown; 1: Non alive; 2: Alive	No	Default value:0,1,2

Response parameter

Parameter	Type	Description
code	int	Result code (0:successful)
data	json	Returned data
msg	string	Result description

Field information(data)

Parameter	Type	Description
export_url	string	Exported file path
result_num	int	Number of query results
total_num	int	Total number of records
zippasswd	string	Compressed password

Example

Request example

```
{
  "msg_id":"1040"
}
```

Response example

```
{
  "code": 0,
  "data": {
    "export_url": "tmp/catch_2019-01-23-12-09-42.zip",
    "result_num": 10,
    "total_num": 108,
                "zippasswd":"xxxx"
  },
  "msg": ""
```

```
}
```

3.16 Exporting selected face captures

Interface description

Interface url	http://${ip}:${port}/api/json
Request Method	POST
Request Parameter Format	JSON
Interface Description	Exports selected captures.

Request parameter

Parameter	Type	Description	Required	Example
msg_id	string	“1043”	Yes	
snap_ids	string	A set of image IDs, which can be obtained through query (Separate image IDs with commas [,]. The image quantity ranges from 1 to 50.)	Yes	

Response parameter

Parameter	Type	Description
code	int	Result code (0:successful)
data	json	Returned data
msg	string	Result description

Field information(data)

Parameter	Type	Description
export_url	string	Exported file path
failed_img_ids	string	ID of the image that cannot be exported
result_num	int	Number of query results
total_num	int	Total number of records
zippasswd	string	Compressed password

Example

Request example

```
{
```

```
  "msg_id":"1043",
  "snap_ids":"74a199aa-dad7-4d3a-95f0-f74dbb818974,f12ed623-87d6-49e0-939a-97262a0407e9"
}
```

Response example

```
{
  "code": 0,
  "data": {
    "export_url": "tmp/catch_2019-01-23-12-44-37.zip",
    "failed_img_ids": "",
    "result_num": 5,
    "total_num": 5,
                    "zippasswd":"xxxx"
  },
  "msg": ""
}
```

3.17 Querying body Alarm records based on criteria

Interface description

Interface url	http://${ip}:${port}/api/json
Request Method	POST
Request Parameter Format	JSON
Interface Description	Querying Body capture records based on criteria

Request parameter

Parameter	Type	Description	Required	Example
msg_id	string	“1068”	Yes	
channel	int	Channel; value range:[1,8] of SenseNebula-M4s, [1,16] of SenseNebula-M8s	No	
start_time	string	Start time	No	yyyy-MM-dd HH:mm:ss
stop_time	string	End time	No	
start_no	int	Start position	No	Default value:1
qry_len	int	Number of queried items; maximum value:50	No	Default value:10
alarm_type	int	Alarmtype	No	

Response parameter

Parameter	Type	Description
code	int	Result code (0:successful)
data	json	Returned data
msg	string	Result description

Field information(data)

Parameter	Type	Description
record	json list	Returned data
result_num	int	Number of query results
total_num	int	Total number of records

Field information(record)

Parameter	Type	Description
camera_name	string	Camera name
channel	int	Channel; value range:[1,8] of SenseNebula-M4s, [1,16] of SenseNebula-M8s
position	string	Camera position
quality	int	Face quality score； value range:[0,100]
snap_id	string	ID of the capture
snap_path	string	Path of the capture
trigger	string	Capture time
body_attr	json	Body capture attributes, for the details, see the appendix C

Note: By default, captures are sorted by capture time.

Example

Request example

```
{
  "msg_id":"1068",  "start_no":1,
  "qry_len":1,
  "channel":1,
  "start_time":"2019-10-16 13:20:00",
  "stop_time":"2019-10-16 18:00:00"
  "alarm_type":"371,372"
}
```

Response example

```
{
```

```
"code": 0,
"data": {
  "record": [
    {
      "body_attr": {
        "bag_style": "bag_style_type_none",
        "cap_style": "hat_style_type_none",
        "coat_color": "black",  "coat_length":
        "long_sleeve",  "coat_style": "t_shirt",
        "gender_code": "female",  "hair_color":
        "black",
        "hair_style": "long",
        "shoes_color": "black",
        "shoes_style": "sandal",
        "st_age": "st_adult",  "st_bag":
        "st_bag",  "st_coat_pattern":
        "st_pure",  "st_hat": "st_hat",
        "st_hold_object_in_front": "st_hold_object_in_front",
        "st_pedestrian_angle": "st_front",
        "st_respirator": "st_respirator",
        "st_trousers_pattern": "st_pure",
        "st_umbrella": "st_umbrella",
        "trousers_color": "black",
        "trousers_len": "st_skirt"
      },
      "camera_name": "c1",
      "channel": 1,
      "position": "",  "quality":
      67,
      "snap_id":  "cb19f0a6-0e55-4b86-8079-d22d530ad14e",
      "snap_path": "record/17/body/cb19f0a6-0e55-4b86-8079-d22d530ad14e.jpg",
      "trigger": "2019-10-16 14:01:41"
      "alarm_type": "371,372",
      "helmet_type":0,  "work_cloth_type":0,
      "work_cloth_color":0,
      "person_nums_in_area":0,
      "person_nums_crossline":0,
      "person_nums_cross_backline":0
    }
  ],
```

```
    "result_num": 1,
    "total_num": 371
  },
  "msg": ""
}
```


3.18 Exporting alarm body images in pagination mode

Interface description

Interface url	http://${ip}:${port}/api/json
Request Method	POST
Request Parameter Format	JSON
Interface Description	Exporting body images in pagination mode

Request parameter

Parameter	Type	Description	Required	Example
msg_id	string	“1070”	Yes	
start_no	int	Start index	No	Default value: 1
qry_len	int	Number of exported items; maximum value:500	No	Default value:10
channel	int	Channel; value range:[1,8] of SenseNebula-M4s, [1,16] of SenseNebula-M8s	No	
start_time	string	Start time	No	
stop_time	string	End time	No	
alarm_type	int	Alarmtype	No	

Response parameter

Parameter	Type	Description
code	int	Result code (0:successful)
data	json	Returned data
msg	string	Result description

Field information(data)

Parameter	Type	Description
export_url	string	Exported file path
result_num	int	Number of query results
total_num	int	Total number of records

Example

Request example

```
{
  "msg_id":"1070"
}
```

Response example

```
{
  "code": 0,
  "data": {
    "export_url": "tmp/body_2019-10-23-12-09-42.zip",
    "result_num": 10,
    "total_num": 108
  },
  "msg": ""
}
```

3.19 Exporting selected alarm body captures

Interface description

Interface url	http://${ip}:${port}/api/json
Request Method	POST
Request Parameter Format	JSON
Interface Description	Exporting Selected Body Captures

Request parameter

Parameter	Type	Description	Required	Example
msg_id	string	“1069”	Yes	
snap_ids	string	A set of image IDs, which can be obtained through query (Separate image IDs with commas [,]. The image quantity ranges from 1 to 50.)	Yes	

Response parameter

Parameter	Type	Description
code	int	Result code (0:successful)
data	json	Returned data

msg	string	Result description

Field information(data)

Parameter	Type	Description
export_url	string	Exported file path
failed_img_ids	string	ID of the image that cannot be exported
result_num	int	Number of query results
total_num	int	Total number of exported images

Example

Request example

```
{
  "msg_id":"1069",
  "snap_ids":"74a199aa-dad7-4d3a-95f0-f74dbb818974,f12ed623-87d6-49e0-939a-97262a0407e9"
}
```

Response example

```
{
  "code": 0,
  "data": {
    "export_url": "tmp/body_2019-01-23-12-44-37.zip",
    "failed_img_ids": "",
    "result_num": 2,
    "total_num": 2
  },
  "msg": ""
}
```

3.20 Querying alarm records based on criteria

Interface description

Interface url	http://${ip}:${port}/api/json
Request Method	POST
Request Parameter Format	JSON
Interface Description	Queries alarm images based on criteria.

Request parameter

Parameter	Type	Description	Required	Example
msg_id	string	“1036”	Yes	

channel	int	Channel; value range:[1,16] of SenseNebula-M4s, [1,32] of SenseNebula-M8s	No	
lib_name	string	Name of a face database	No	
lib_ids	string	A set of IDs	No	
lib_type	int	Database type (1:key personnel database, 2:whitelist database)	No	
start_time	string	Start time	No	
stop_time	string	End time	No, When start_time is filled in, stop_time must also be filled in	
start_no	int	Start position	No	Default value: 1
qry_len	int	Number of queried items; maximum value: 50	No	Default value:10
alive_type	string	Alive type separate by comma, 0：Unknown; 1: Non alive; 2: Alive, Non-alive images do not participate in the comparison	No	Default value:0,2

Response parameter

Parameter	Type	Description
code	int	Result code (0:successful)
data	json	Returned data
msg	string	Result description

Field information(data)

Parameter	Type	Description
record	json list	Returned data

result_num	int	Number of query results
total_num	int	Total number of records

Field information(record)

Parameter	Type	Description
camera_name	string	Camera name
channel	int	Channel; value range:[1,16] of SenseNebula-M4s, [1,32] of SenseNebula-M8s
create_time	string	Creation time
img_id	string	Image ID
img_path	string	Image path
lib_id	int	Database ID
lib_name	string	Database name
lib_type	int	Database type (1:key personnel database, 2:whitelist database)
person_addr	string	Address
person_age	string	Age, between 0 and 999
person_gender	string	Gender (0:female, 1:male)
person_idcard	string	Identity code
person_name	string	Name
position	string	Camera position
ranking	int	1 indicates the highest ranking in the comparison results.
similarity	int	Similarity score; value range:[0,100]
snap_id	string	Face capture id
snap_path	string	Face capture path Concatenate the URL
threshold	int	Comparison threshold (camera-bound); value range:[0,100]
trigger	string	Trigger time
alive_type	int	Alive type, 0： Unknown; 1: Non alive; 2: Alive
face_attr	json	Face capture attributes, for details, see the appendix B

Note: By default, captures are sorted by capture time.

Example

Request example

```
{
   "msg_id":"1036",
   "start_no":1,
   "qry_len":1
}
```

Response example


```
{
   "code": 0,
   "data": {
      "record": [
         {
            "camera_name": "no2",
            "channel": 2,
            "face_attr": {
               "cap_style": "hat_style_type_none",  "gender_code":
               "female",
               "glass_style": "glasses_style_type_none",
               "mustache_style": "mustache_style_type_none",
               "st_respirator": "st_respirator_mouth",
               "respirator_color": "color_type_none",   "st_age":
               "st_adult",
               "st_age_value": "26.000000",
               "st_expression": "st_angry"
            },
            "create_time": "2019-01-23 11:37:26",
            "img_id":  "71f11a77-94c0-43c9-ab23-6e9076f6127a",   "img_path":
            "img/1_71f11a77-94c0-43c9-ab23-6e9076f6127a.jpg",  "lib_id": 1,
            "lib_name": "test",
            "lib_type": 1,
            "person_addr": "wang",
            "person_age": "14",
            "person_gender": "",
            "person_idcard": "wang",
            "person_name": "1101060000000061481",
            "position": "aaa",
            "ranking": 1,
            "similarity": 35,
            "snap_id": "1d9abf3d-09e1-4ef2-8832-f3d025ae0f91",   "snap_path":
            "record/1d9abf3d-09e1-4ef2-8832-f3d025ae0f91.jpg",  "threshold":
            10,
            "trigger": "2019-01-23 14:29:22",
```

```
            "alive_type": 2
        }
    ],
    "result_num": 1,
    "total_num": 297
  },
  "msg": ""
}
```


3.21 Exporting alarm images in pagination mode

Interface description

Interface url	http://${ip}:${port}/api/json
Request Method	POST
Request Parameter Format	JSON
Interface Description	Exports alarm images in pagination mode.

Request parameter

Parameter	Type	Description	Required	Example
msg_id	string	“1039”	Yes	
start_no	int	Start index	No	Default value: 1
qry_len	int	Number of exported items; maximum value:500	No	Default value:10
channel	int	Channel; value range:[1,16] of SenseNebula-M4s, [1,32] of SenseNebula-M8s	No	
lib_name	string	Name of a face database	No	
lib_type	int	Database type (1:key personnel database, 2:whitelist database)	No	
start_time	string	Start time	No	
stop_time	string	End time	No, When start_time is filled in, stop_time	

			must also be filled in	
lib_ids	string	A set of IDs	Yes	
alive_type	string	Alive type separate by comma, 0：Unknown; 1: Non alive; 2: Alive, Non-alive images do not participate in the comparison	No	Default value:0,2

Response parameter

Parameter	Type	Description
code	int	Result code (0:successful)
data	json	Returned data
msg	string	Result description

Field information(data)

Parameter	Type	Description
export_url	string	Exported file path
result_num	int	Number of query results
total_num	int	Total number of records
zippasswd	string	Compressed password

Example

Request example

```
{
  "msg_id": "1039",
  "lib_ids": "1"
}
```

Response example

```
{
  "code": 0,
  "data": {
    "export_url": "tmp/alarm_2019-01-23-14-31-51.zip",
    "result_num": 10,
    "total_num": 298,
                    "zippasswd":"xxxx"
  },
  "msg": ""
}
```

3.22 Exporting selected alarm images

Interface description

Interface url	http://${ip}:${port}/api/json
Request Method	POST
Request Parameter Format	JSON
Interface Description	Exports selected alarm images.

Request parameter

Parameter	Type	Description	Required	Example
msg_id	string	“1042”	Yes	
snap_ids	string	A set of image IDs, which can be obtained through query (Separate image IDs with commas [,]. The image quantity ranges from 1 to 50.)	Yes	

Response parameter

Parameter	Type	Description
code	int	Result code (0:successful)
data	json	Returned data
msg	string	Result description

Field information(data)

Parameter	Type	Description
export_url	string	Exported file path
failed_img_ids	string	ID of the image that cannot be exported
result_num	int	Number of query results
total_num	int	Total number of records
zippasswd	string	Compressed password

Example

Request example

```
{
  "msg_id":"1042",
  "snap_ids":"91f3705f-539b-4474-8f8b-fffb86d8ffa7"
```

```
}
```

Response example

```
{
   "code": 0,
   "data": {
      "export_url": "tmp/alarm_2019-01-23-14-35-29.zip",
      "failed_img_ids": "",
      "result_num": 1,
      "total_num": 1,
                        "zippasswd":"xxxx"
   },
   "msg": ""
}
```

3.23 snap or alarm image option delete

Interface description

Interface url	http://${ip}:${port}/api/json
Request Method	POST
Request Parameter Format	JSON
Interface Description	snap or alarm image option delete

Request parameter

Parameter	Type	Description	Required	Example
msg_id	string	“2129”	Yes	
catch_ids	string	A set of image IDs, which can be obtained through query (Separate image IDs with commas [,]. The image quantity ranges from 1 to 50.)	Yes	
type	int	delete type 0:face image 1:body image 2:alarm image 3:object image 4:violate operate image	Yes	

Response parameter

Parameter	Type	Description
code	int	Result code (0:successful)
data	json	NULL
msg	string	Result description

Example

Request example

```
{
  "msg_id": "2129", "catch_ids":
  "11,22,33,44", "type": 0
}
```

Response example

```
{
  "code": 0,
  "data": "",
  "msg": ""
}
```

3.24 Storing face images in a database in batches

Interface description

Interface url	http://${ip}:${port}/api/form
Request Method	POST
Request Parameter Format	FORM
Interface Description	Stores face images in a database in batches

Request parameter

Parameter	Type	Description	Required	Example
msg_id	string	“1030”	Yes	
lib_id	int	Database ID	Yes	
img_*	file list	Face images	Yes	
uuid	string	Batch identifier	Yes	

Response parameter

Parameter	Type	Description
code	int	Result code (0:successful)
data	json	Failure response
msg	string	Result description

Field information(data)

Parameter	Type	Description
url	string	CSV file that contains information about database storage
err_count	int	Number of images that cannot be stored in the database

Note 1: The image files to be imported in a batch must be named in the format of Name_img_id_Identity_Gender_Age_Address.
Note 2: img_id is an attribute that identifies an image. It can be customized and must be unique in the database. It may contain English letters, numbers, and hyphens (-) and is up to 48 characters in length. Note 3: Gender can be set to 0 or 1. 0 indicates female and 1 indicates male.
Note 4: Give the attributes in each image filename in the specified order and separate attributes with underscores (_). The system automatically retrieves the image attributes. If an attribute is empty, fill it with a pound sign (#).

Example

Request example

```
Input Files
{
  'img_1': name='zhangsan_asdqwe12345_32011111_0_23_beijing.jpg',
  'img_2': name='lisi_abcdse0987_32011111_1_41_shanghai.jpg',
   ...
}
  "msg_id":"1030",
  "lib_id":11,
  "uuid":"test"
```

Response example

```
{
  "code": 0,
  "data":
    { "err_count":
    0,
    "url": "tmp/image_.csv"
  },
  "msg": ""
}
```

3.25 Deleting face images in batches

Interface description

Interface url	http://${ip}:${port}/api/json
Request Method	POST

Request Parameter Format	JSON
Interface Description	Deletes face images in batches

Request parameter

Parameter	Type	Description	Required	Example
msg_id	string	“1044”	Yes	
img_ids	string	Image ID, separate multiple image IDs with commas (,) and up to 50 IDs	Yes	
lib_id	int	ID of the face database where the image is stored	Yes	

Response parameter

Parameter	Type	Description
code	int	Result code (0:successful)
data	json	NULL
msg	string	Result description

Example

Request example

```
{
  "msg_id":"1044",
  "img_ids":"f30c0f4b-d783-44a5-b118-b4db9ea2b8c4,85b728f8-ed96-42c3-a7e2-f811cf7a1481",
  "lib_id":2
}
```

Response example

```
{
  "code": 0,
  "data": "",
  "msg": ""
}
```

3.26 Deleting capture records and alarm records

Interface description

Interface url	http://${ip}:${port}/api/json
Request Method	POST
Request Parameter Format	JSON

Interface Description	Deleting all capture records and alarm records

Request parameter

Parameter	Type	Description	Required	Example
msg_id	string	“1045”	Yes	

Response parameter

Parameter	Type	Description
code	int	Result code (0:successful)
data	json	NULL
msg	string	Result description

Example

Request example

```
{
  "msg_id":"1045"
}
```

Response example

```
{
  "code": 0,
  "data": "",
  "msg": ""
}
```

3.27 Querying body capture records based on criteria

Interface description

Interface url	http://${ip}:${port}/api/json
Request Method	POST
Request Parameter Format	JSON
Interface Description	Querying Body capture records based on criteria

Request parameter

Parameter	Type	Description	Required	Example
msg_id	string	“1046”	Yes	
channel	int	Channel; value range:[1,16] of SenseNebula-M4s, [1,32] of	No	
		SenseNebula-M8s		
start_time	string	Start time	No	yyyy-MM-dd HH:mm:ss

stop_time	string	End time	No, When start_time is filled in, stop_time must also be filled in	
start_no	int	Start position	No	Default value:1
qry_len	int	Number of queried items; maximum value:50	No	Default value:10

Response parameter

Parameter	Type	Description
code	int	Result code (0:successful)
data	json	Returned data
msg	string	Result description

Field information(data)

Parameter	Type	Description
record	json list	Returned data
result_num	int	Number of query results
total_num	int	Total number of records

Field information(record)

Parameter	Type	Description
camera_name	string	Camera name
channel	int	Channel; value range:[1,16] of SenseNebula-M4s, [1,32] of SenseNebula-M8s
position	string	Camera position
quality	int	Face quality score； value range:[0,100]
snap_id	string	ID of the capture
snap_path	string	Path of the capture
trigger	string	Capture time
body_attr	json	Body capture attributes, for the details, see the appendix C

Note: By default, captures are sorted by capture time.

Example

Request example


```
{
  "msg_id":"1046",  "start_no":1,
  "qry_len":1,
  "channel":1,
  "start_time":"2019-10-16 13:20:00",
  "stop_time":"2019-10-16 18:00:00"
}
```


Response example


```
{
  "code": 0,
  "data": {
    "record": [
      {
        "body_attr": {
          "bag_style": "bag_style_type_none",
          "cap_style": "hat_style_type_none",
          "coat_color": "black",   "coat_length":
          "long_sleeve",  "coat_style": "t_shirt",
          "gender_code": "female",  "hair_color":
          "black",
          "hair_style": "long",  "shoes_color":
          "black",  "shoes_style": "sandal",
          "st_age":      "st_adult",   "st_bag":
          "st_bag",  "st_coat_pattern":
          "st_pure",  "st_hat": "st_hat",
          "st_pedestrian_angle": "st_front",
          "st_respirator": "st_respirator",
          "st_trousers_pattern": "st_pure",
          "st_umbrella": "st_umbrella",
          "trousers_color": "black",
          "pants_length_type": "st_skirt",
          "glasses_style_type": "glasses_style_type_none",
          "st_glove": "st_glove",
          "st_status_stype": "st_normal",
          "st_uniform": "st_common_clothing",
          "trousers_len": "st_skirt"
```

```
            },
            "camera_name": "c1",
            "channel": 1,
            "position": "",
            "quality": 67,
            "snap_id":  "cb19f0a6-0e55-4b86-8079-d22d530ad14e",
            "snap_path": "record/17/body/cb19f0a6-0e55-4b86-8079-d22d530ad14e.jpg",
            "trigger": "2019-10-16 14:01:41"
        }
    ],
    "result_num": 1,
    "total_num": 371
  },
  "msg": ""
}
```


3.28 Exporting body images in pagination mode

Interface description

Interface url	http://${ip}:${port}/api/json
Request Method	POST
Request Parameter Format	JSON
Interface Description	Exporting body images in pagination mode

Request parameter

Parameter	Type	Description	Required	Example
msg_id	string	“1048”	Yes	
start_no	int	Start index	No	Default value: 1
qry_len	int	Number of exported items; maximum value:500	No	Default value:10
channel	int	Channel; value range:[1,16] of SenseNebula-M4s, [1,32] of SenseNebula-M8s	No	
start_time	string	Start time	No	
stop_time	string	End time, When start_time is filled in, stop_time must also be filled	No	

		in		

Response parameter

Parameter	Type	Description
code	int	Result code (0:successful)
data	json	Returned data
msg	string	Result description

Field information(data)

Parameter	Type	Description
export_url	string	Exported file path
result_num	int	Number of query results
total_num	int	Total number of records
zippasswd	string	Compressed password

Example

Request example

```
{
  "msg_id":"1048"
}
```

Response example

```
{
  "code": 0,
  "data": {
    "export_url": "tmp/body_2019-10-23-12-09-42.zip",
    "result_num": 10,
    "total_num": 108,
                    "zippasswd":"xxxx"
  },
  "msg": ""
}
```

3.29 Exporting selected body captures

Interface description

Interface url	http://${ip}:${port}/api/json
Request Method	POST
Request Parameter Format	JSON
Interface Description	Exporting Selected Body Captures

Request parameter

Parameter	Type	Description	Required	Example
msg_id	string	“1047”	Yes	
snap_ids	string	A set of image IDs, which can be obtained through query (Separate image IDs with commas [,]. The image quantity ranges from 1 to 50.)	Yes	

Response parameter

Parameter	Type	Description
code	int	Result code (0:successful)
data	json	Returned data
msg	string	Result description

Field information(data)

Parameter	Type	Description
export_url	string	Exported file path
failed_img_ids	string	ID of the image that cannot be exported
result_num	int	Number of query results
total_num	int	Total number of exported images
zippasswd	string	Compressed password

Example

Request example

```
{
  "msg_id":"1047",
  "snap_ids":"74a199aa-dad7-4d3a-95f0-f74dbb818974,f12ed623-87d6-49e0-939a-97262a0407e9"
}
```

Response example

```
{
  "code": 0,
  "data": {
    "export_url": "tmp/body_2019-01-23-12-44-37.zip",
    "failed_img_ids": "",
```

```
    "result_num": 2,
    "total_num": 2,
                    "zippasswd":"xxxx"
  },
  "msg": ""
}
```


3.30 Querying a single stranger database image

Interface description

Interface url	http://${ip}:${port}/api/json
Request Method	POST
Request Parameter Format	JSON
Interface Description	Queries a single stranger database image

Request parameter

Parameter	Type	Description	Required	Example
msg_id	string	“2057”	Yes	
person_id	string	Stranger id in the stranger database	Yes	

Response parameter

Parameter	Type	Description
code	int	Result code (0:successful)
data	json	Returned data
msg	string	Result description

Field information(data)

Parameter	Type	Description
record	json list	Returned data
result_num	int	Number of query results
total_num	int	Total number of records

Field information(record)

Parameter	Type	Description
person_id	string	Stranger id in the stranger database
camera_name	string	Camera name
channel	int	Channel; value range:[1,16] of SenseNebula-M4s, [1,32] of SenseNebula-M8s
position	string	Camera position

trigger	string	Capture time
count	int	Reserved field
similarity	int	Similarity score; value range:[0,100]
threshold	int	Threshold; value range:[0,100]
snap_path	string	Storage path of the capture image
quality	double	Face quality score； value range:[0,100]
feature	string	Features of the image
lib_id	int	Stranger database id
img_id	string	Stranger id in the stranger database
img_path	string	Stranger database image path
face_attr	json	Face attributes
wander_thresHold	int	The threshold of occurrences number
appear_count	int	Current occurrences
wander_channels	string	The set wandering places for the wandering alarm

Example

Request example

```
{
  "msg_id":"2057",
  "person_id":"74a199aa-dad7-4d3a-95f0-f74dbb818974"
}
```

Response example

```
{
  "code": 0,
  "data": {
    "record": [
      {
        "appear_count": 617,
        "camera_name": "2222",
        "channel": 1,
        "count": 0,
        "face_attr": {
          "cap_style": "hat_style_type_none",
          "gender_code": "male",  "glass_style":
          "transparent_glasses",
```

```
            "mustache_style": "mustache_style_type_none",
            "respirator_color": "color_type_none",   "st_age":
            "st_adult",
            "st_age_value": "34.000000",
            "st_expression": "st_calm",
            "st_respirator": "st_respirator_mouth"
        },
        "feature": "-0.104127,0.002276,0.024537,-0.033187,0.030285,-0.072717,0.019915,0.024345,-
0.044331,0.049733,-0.035291,0.059003,-0.037726,0.007244,-0.072045,0.037151,0.039421,-0.070434,-
0.061541,-0.097024,-0.126349,0.075204,0.006515,-0.009571,-0.031487,0.038014,-
0.065441,0.119259,0.076393,0.026436,-0.031877,0.016795,-0.009321,0.026091,-0.046108,0.011354,-
0.010204,0.040885,0.062935,0.029434,0.078745,0.017767,-0.015069,0.022536,-0.017032,0.086200,-
0.026295,-0.028654,0.037675,-0.044829,-0.018687,-0.078848,-0.027031,-0.019416,0.004731,0.000032,-
0.036339,-0.083483,0.017351,0.065678,0.000914,-0.097164,-0.008561,0.016648,0.034728,-0.068145,-
0.021820,0.061823,-0.087133,0.013586,-0.055781,0.038379,0.131119,-0.005952,0.006476,-
0.076553,0.096455,0.017044,0.081987,0.060761,-0.031640,-0.062999,-
0.043806,0.084327,0.050123,0.015222,-0.018393,0.027325,0.011661,-0.032823,0.092382,-
0.052642,0.085183,-0.071866,0.043570,-0.046830,0.075638,0.058057,-0.051325,-0.001055,-0.043557,-
0.047252,0.079871,0.032689,0.032209,-0.043519,0.029665,0.081827,-0.022376,-0.018227,-
0.093412,0.044292,-0.008605,-0.025573,-0.012838,0.005268,-0.033475,-0.015778,-
0.001937,0.047540,0.144391,0.046389,0.047483,0.068101,0.115001,0.029729,-0.022760,-
0.079314,0.008100,0.020663,-0.003158,0.051261,-0.046869,0.051095,0.088540,-0.130428,-0.043711,-
0.007544,-0.137595,0.060365,0.100227,-0.009078,-0.076617,-
0.049388,0.166211,0.087785,0.033366,0.044970,-0.080261,0.059016,-0.095253,-0.009456,-0.033149,-
0.045405,0.002692,0.070632,0.007301,0.015203,-0.136112,-0.014628,-0.035668,-0.041288,-0.032030,-
0.005895,-0.001279,0.015612,-0.030636,0.065767,0.027069,0.076898,-0.126541,0.056970,0.024870,-
0.071534,-0.048723,-0.048205,-0.043423,0.069501,0.031384,-0.005242,0.046313,0.063133,-
0.019480,0.018758,0.051721,-0.022255,0.069942,-0.002007,0.077761,-0.055775,-
0.127181,0.048390,0.099645,0.014321,0.031103,-0.046453,0.085196,-0.032407,0.070198,0.016354,-
0.077038,0.052725,0.000173,0.034798,0.067052,-0.008190,0.104932,0.081501,-0.105322,-
0.082914,0.023751,0.046402,-0.128050,-0.036224,0.020746,0.032580,0.028603,-0.147907,-
0.033891,0.107099,0.108218,0.030483,0.010408,-0.170111,-0.069021,0.088303,-0.080043,-0.068459,-
0.079340,0.094466,-0.113902,-
0.049835,0.041236,0.062973,0.013637,0.062961,0.024588,0.159089,0.108877,0.126343,-
0.057526,0.041761,0.106236,-0.038487,0.015644,-0.005179,-0.044414,0.000652,0.004399,-0.009162,-
0.042643,0.203688,-0.051139,0.042477,0.016853,0.033053,",
        "img_id": "9297df3e-7fbd-4c9f-a856-1c394b2fad94",
        "img_path": "",
        "lib_id": 2147483647,
        "position": "",
        "quality": 0.9495999813079834,
        "similarity": 0,
        "person_id":  "9297df3e-7fbd-4c9f-a856-1c394b2fad94",
```

```
            "snap_path": "record/channel1/face/9297df3e-7fbd-4c9f-a856-1c394b2fad94.jpg", "threshold":
            85,
            "trigger": "2020-12-22 14:40:02",
            "wander_channels": "",
            "wander_threshold": 0
        }
    ],
    "result_num": 1,
    "total_num": 1
  },
  "msg": ""
}
```


3.31 Querying the stranger database image based on criteria

Interface description

Interface url	http://${ip}:${port}/api/json
Request Method	POST
Request Parameter Format	JSON
Interface Description	陌生人记录条件查询

Request parameter

Parameter	Type	Description	Required	Example
msg_id	string	“2053”	Yes	
appear_time	int	Current occurrences(Reserved,not used currently)	No	6
channels	string	A comma-separated list of channel numbers, empty means to query all channels	No	1,2,3
start_no	int	Start position (Default value:1)	No	1
qry_len	int	Number of query items [1,50] (Default value:10)	No	10
start_time	string	Start time	No	2020-12-09 00:00:00
stop_time	string	End time	No,When start_time is	2020-12-09 00:00:00

			filled in, stop_time must also be filled in	

Response parameter

Parameter	Type	Description
code	int	Result code (0:successful)
data	json	Returned data
msg	string	Result description

Field information(data)

Parameter	Type	Description
record	json list	Returned data
result_num	int	Number of query results
total_num	int	Total number of records

Field information(record)

Parameter	Type	Description
img_id	string	Stranger id in the stranger database
person_id	string	Stranger id in the stranger database
camera_name	string	Camera name
channel	int	Channel; value range:[1,16] of SenseNebula-M4s, [1,32] of SenseNebula-M8s
position	string	Camera position
trigger	string	Capture time
count	int	Reserved field
similarity	int	Similarity score; value range:[0,100]
threshold	int	Threshold; value range:[0,100]
snap_path	string	Storage path of the capture image
lib_id	int	Stranger database id
quality	int	Quality score, value range:[0,100]
feature	string	Features of the image
img_path	string	Stranger database image path
face_attr	json	Face attributes
wander_thresHold	int	The threshold of occurrences number

appear_count	int	Current occurrences
wander_channels	string	The set wandering places for the wandering alarm

Example

Request example


```
{
  "appear_time": 170,
  "channels": "1,2,3",
  "start_no": 1,
  "qry_len": 5,
  "start_time": "2020-12-09 00:00:00",
  "stop_time": "2020-12-25 23:59:59",
  "msg_id": "2053"
}
```


Response example


```
{
  "code": 0,
  "data": {
    "record": [
      {
        "appear_count": 4,
        "camera_name": "2222",
        "channel": 1,
        "count": 0,
        "face_attr": {
          "cap_style": "hat_style_type_none",
          "gender_code": "female",  "glass_style":
          "transparent_glasses",
          "mustache_style": "mustache_style_type_none",
          "respirator_color": "color_type_none",   "st_age":
          "st_adult",
          "st_age_value": "25.000000",
          "st_expression": "st_calm",
          "st_respirator": "st_respirator_mouth"
        },
        "feature": "-0.026478,0.065735,-0.006256,-0.086443,-0.050911,-
0.048465,0.010536,0.000436,0.096051,0.099756,-0.080074,-
0.030513,0.029515,0.039651,0.026401,0.132975,0.010522,0.116246,0.060077,-
0.007500,0.042533,0.056197,-0.002587,-0.027385,0.006628,-0.250936,0.012139,-
0.050103,0.036270,0.017565,-0.117694,0.014557,0.056977,-0.009651,0.002137,0.068392,-0.065546,-
0.073523,-0.049962,-0.017537,-0.092938,-0.073812,0.031167,0.025311,-0.036445,0.021284,-0.087750,-
```

0.020813,0.073959,0.003894,0.196398,0.046645,0.040543,0.020757,0.018493,-0.065046,0.011753,-0.064238,0.017235,0.102595,-0.022872,-0.050686,0.062516,-0.012462,-0.121988,-0.076012,-0.024257,-0.105815,0.054974,0.130866,-0.029993,0.003866,-0.116464,0.009327,0.058256,-0.016975,-0.026823,-0.069650,0.023554,-0.080053,-0.046082,-0.058355,-0.116140,-0.120337,0.004977,0.049625,0.047122,0.056541,0.009602,0.075393,-0.028699,0.032959,-0.002495,-0.051157,0.033599,-0.029655,0.037444,0.025649,0.049027,-0.005764,-0.035771,0.069594,-0.006207,0.021684,-0.019491,-0.018128,0.014100,-0.014944,-0.012961,0.014761,0.044409,-0.059613,-0.049006,0.027990,0.003058,0.027736,0.043798,0.120337,-0.039777,-0.020574,-0.088249,-0.020377,0.057750,0.050834,0.048142,0.063423,0.026190,-0.084770,-0.013995,0.021762,-0.000415,-0.024960,0.054004,0.101843,-0.056893,-0.078662,0.038097,0.074655,-0.004787,0.123436,0.086155,-0.014199,-0.101499,-0.004702,0.011176,-0.067872,-0.017284,-0.031385,0.038083,0.054580,0.059023,0.008625,-0.012469,0.054369,-0.060246,-0.138971,0.100487,-0.004681,-0.037415,-0.117174,0.020574,-0.075646,-0.027582,-0.045028,0.106419,0.012491,-0.017038,-0.067450,0.081621,0.030815,-0.024580,-0.008491,-0.081361,0.024763,0.039306,-0.037008,-0.039552,-0.099390,-0.198879,0.010382,0.096494,-0.059972,0.001153,0.027343,-0.017741,-0.007395,0.076904,-0.005349,-0.001799,-0.031216,0.037507,-0.084510,0.052760,0.027104,0.075787,0.078352,-0.030710,0.127323,-0.024489,-0.020890,0.013454,0.041499,-0.028770,-0.197051,0.006902,0.063458,0.024932,0.046223,-0.032818,-0.103094,-0.007486,0.047073,-0.052704,-0.019491,0.006214,-0.011654,-0.009131,-0.056232,-0.050384,0.051621,0.032629,-0.016054,0.016525,0.042455,-0.065046,-0.054046,0.062333,0.022901,0.012779,0.074564,-0.093387,0.152656,-0.049189,-0.079871,0.023414,-0.105829,-0.007416,-0.021263,-0.004344,0.028538,-0.149809,-0.001638,-0.039967,0.055909,0.015787,0.048226,0.089395,0.025895,-0.109350,0.037458,0.002074,0.017355,-0.020476,-0.024602,-0.119022,0.006087,",
            "img_id": "687fc7d7-eea7-4ba0-b4f8-d153002f7720",
            "img_path": "",
            "lib_id": 2147483647,
            "position": "",
            "quality": 0.91949999332427979,
            "similarity": 0,
            "person_id": "687fc7d7-eea7-4ba0-b4f8-d153002f7720",
            "snap_path": "record/channel1/face/687fc7d7-eea7-4ba0-b4f8-d153002f7720.jpg",
            "threshold": 85,
            "trigger": "2020-12-24 15:44:10",
            "wander_channels": "1",
            "wander_threshold": 5
        }
    ],
    "result_num": 1,
    "total_num": 24
  },
  "msg": ""
}

3.32 Deleting greylist images in batches

Interface description

Interface url	http://${ip}:${port}/api/json
Request Method	POST
Request Parameter Format	JSON
Interface Description	Deletes greylist images in batches

Request parameter

Parameter	Type	Description	Required	Example
msg_id	string	“1072”	Yes	
img_ids	string	Image ID, separate multiple image IDs with commas (,) and up to 50 IDs	Yes	
lib_id	int	ID of the face database where the image is stored,greylist is 2147483647	Yes	

Response parameter

Parameter	Type	Description
code	int	Result code (0:successful)
data	json	NULL
msg	string	Result description

Example

Request example

```
{
  "msg_id":"1072",
  "img_ids":"v214_1654673649_1389_0",
  "lib_id":2147483647
}
```

Response example

```
{
  "code": 0,
  "data": "",
  "msg": ""
}
```

3.33 Exporting the stranger database images in pagination mode

Interface description

Interface url	http://${ip}:${port}/api/json
Request Method	POST
Request Parameter Format	JSON
Interface Description	Exports the stranger database images in pagination mode

Request parameter

Parameter	Type	Description	Required	Example
msg_id	string	“2049”	Yes	
appear_time	int	Current occurrences(Reserved,not used currently)	No	6
channels	string	A comma-separated list of channel numbers, empty means to query all channels	No	1,2,3
start_no	int	Start position	Yes	Default value: 1
qry_len	int	Number of exported items; maximum value:500	Yes	10
start_time	string	Start time	No	2020-12-09 00:00:00
stop_time	string	End time	No,When start_time is filled in, stop_time must also be filled in	2020-12-09 00:00:00

Response parameter

Parameter	Type	Description
code	int	Result code (0:successful)
data	json	Returned data
msg	string	Result description

Field information(data)

Parameter	Type	Description
export_url	string	Exported file path

result_num	int	Number of query results
total_num	int	Total number of records
zippasswd	string	Compressed password

Example

Request example

```
{
  "appear_time": 6,  "channels":
  "1,2,3",  "start_no": 1,
  "qry_len": 10,
  "start_time": "2020-12-09 00:00:00",
  "stop_time": "2020-12-25 23:59:59",
  "msg_id": "2049"
}
```

Response example

```
{
  "code": 0,
  "data": {
    "export_url": "tmp/grey_2020-12-24-17-21-52.zip",
    "result_num": 10,
    "total_num": 22,
                    "zippasswd":"xxxx"
  },
  "msg": ""
}
```

3.34 Exporting the selected stranger database images

Interface description

Interface url	http://${ip}:${port}/api/json
Request Method	POST
Request Parameter Format	JSON
Interface Description	Exports the selected stranger database images

Request parameter

Parameter	Type	Description	Required	Example
msg_id	string	2050	Yes	
person_ids	string	A set of stranger image IDs in the	Yes	

		stranger database, which can be obtained through query (Separate vehicle IDs with commas [,]. The quantity ranges from 1 to 50.)		

Response parameter

Parameter	Type	Description
code	int	Result code (0:successful)
data	json	Returned data
msg	string	Result description

Field information(data)

Parameter	Type	Description
export_url	string	Exported file path
failed_img_ids	string	ID of the image that cannot be exported
result_num	int	Number of query results
total_num	int	Total number of exported images
zippasswd	string	Compressed password

Example

Request example

```
{
  "msg_id":"2050",
  "person_ids":"74a199aa-dad7-4d3a-95f0-f74dbb818974,f12ed623-87d6-49e0-939a-97262a0407e9"
}
```

Response example

```
{
  "code": 0,
  "data": {
    "export_url": "tmp/catch_2019-01-23-12-44-37.zip",
    "failed_img_ids": "",
    "result_num": 5,
    "total_num": 5,
                    "zippasswd":"xxxx"
  },
```

```
  "msg": ""
}
```

3.35 Querying the wandering alarm records based on creteria

Interface description

Interface url	http://${ip}:${port}/api/json
Request Method	POST
Request Parameter Format	JSON
Interface Description	Queries the wandering alarm records based on creteria

Request parameter

Parameter	Type	Description	Required	Example
msg_id	string	2054	Yes	
channel	int	Channel; value range:[1,16] of SenseNebula-M4s, [1,32] of SenseNebula-M8s	No	
start_time	string	Start time	No	
stop_time	string	End time	No,When start_time is filled in, stop_time must also be filled in	
start_no	int	Start position	No	Default value: 1
qry_len	int	Number of queried items; maximum value: 50	No	Default value:10

Response parameter

Parameter	Type	Description
code	int	Result code (0:successful)
data	json	Returned data
msg	string	Result description

Field information(data)

Parameter	Type	Description
record	json list	Returned data
result_num	int	Number of query results
total_num	int	Total number of records

Field information(record)

Parameter	Type	Description
camera_name	string	Camera name
channel	int	Channel; value range:[1,16] of SenseNebula-M4s, [1,32] of SenseNebula-M8s
create_time	string	Creation time
img_id	string	Stranger id in the stranger database
img_path	string	Stranger database image path
lib_id	int	Stranger database id
lib_name	string	Database name
lib_type	int	Database type (1:key personnel database, 2:whitelist database)
position	string	Camera position
ranking	int	1 indicates the highest ranking in the comparison results
similarity	int	Similarity score; value range:[0,100]
snap_id	string	Capture ID
snap_path	string	Capture path
threshold	int	Threshold; value range:[0,100]
trigger	string	Capture time
alive_type	int	Alive type，0：Unknown；1：Non alive；2：Alive
face_attr	json	Face capture attributes, for details, see the appendix B
alarm_type	int	Face alarm type (0: no alarm, 1: key personnel/allowlist alarm, 2: stranger alarm, 3: hit stranger db alarm)
appear_count	int	Current occurrences
event_type	int	Event alarm type (0: no alarm, 1: wandering alarm, 2: staying alarm)

wander_channels	string	The set wandering places for the wandering alarm
wander_deviceID	string	The trigger place for wandering alarm
wander_trigger	string	The trigger time for wandering alarm
stranger_appear_channel	int	The capture channel of the stranger database image

Note: By default, portrait databases are sorted by creation time.

Example

Request example


```
{
  "msg_id":"2054",
  "start_no":1,
  "qry_len":1
}
```


Response example


```
{
  "code": 0,
  "data": {
    "record": [
      {
        "camera_name": "no2",
        "channel": 2,
        "face_attr": {
          "cap_style": "hat_style_type_none",  "gender_code":
          "female",
          "glass_style": "glasses_style_type_none",
          "mustache_style": "mustache_style_type_none",
          "st_respirator": "st_respirator_mouth",
          "respirator_color": "color_type_none",   "st_age":
          "st_adult",
          "st_age_value": "26.000000",
          "st_helmet_style": "st_helmet_style_type_none",
          "st_expression": "st_angry"
        },
        "create_time": "2019-01-23 11:37:26",
        "img_id":  "71f11a77-94c0-43c9-ab23-6e9076f6127a",   "img_path":
        "img/1_71f11a77-94c0-43c9-ab23-6e9076f6127a.jpg",  "lib_id": 1,
        "lib_name": "test",
```

```
            "lib_type": 1,
            "position": "aaa",
            "ranking": 1,
            "similarity": 35,
            "snap_id": "1d9abf3d-09e1-4ef2-8832-f3d025ae0f91",   "snap_path":
            "record/1d9abf3d-09e1-4ef2-8832-f3d025ae0f91.jpg",  "threshold":
            10,
            "trigger": "2019-01-23 14:29:22",
            "alive_type": 2
        }
    ],
    "result_num": 1,
    "total_num": 297
  },
  "msg": ""
}
```


3.36 Exporting the wandering alarms in pagination mode

Interface description

Interface url	http://${ip}:${port}/api/json
Request Method	POST
Request Parameter Format	JSON
Interface Description	Exports the wandering alarms in pagination mode

Request parameter

Parameter	Type	Description	Required	Example
msg_id	string	2055	Yes	
start_no	int	Start position	No	Default value: 1
qry_len	int	Number of exported items; maximum value:500	No	Default value:10
channel	int	Channel; value range:[1,16] of SenseNebula-M4s, [1,32] of SenseNebula-M8s	No	
start_time	string	Start time	No	
stop_time	string	End time	No,When start_time is filled in, stop_time	

			must also be filled in	

Response parameter

Parameter	Type	Description
code	int	Result code (0:successful)
data	json	Returned data
msg	string	Result description

Field information(data)

Parameter	Type	Description
export_url	string	Exported file path
result_num	int	Number of query results
total_num	int	Total number of exported images
zippasswd	string	Compressed password

Example

Request example

```
{
  "msg_id": "2055"
}
```

Response example

```
{
  "code": 0,
  "data": {
    "export_url": "tmp/alarm_2019-01-23-14-31-51.zip",
    "result_num": 10,
    "total_num": 298,
                    "zippasswd":"xxxx"
  },
  "msg": ""
}
```

3.37 Exporting the selected wandering alarms

Interface description

Interface url	http://${ip}:${port}/api/json
Request Method	POST
Request Parameter Format	JSON
Interface Description	Exports the selected wandering alarms

Request parameter

Parameter	Type	Description	Required	Example
msg_id	string	2056	Yes	
snap_ids	string	A set of capture images IDs, which can be obtained through query (Separate image IDs with commas [,]. The image quantity ranges from 1 to 50.)	Yes	

Response parameter

Parameter	Type	Description
code	int	Result code (0:successful)
data	json	Returned data
msg	string	Result description

Field information(data)

Parameter	Type	Description
export_url	string	Exported file path
failed_img_ids	string	ID of the image that cannot be exported
result_num	int	Number of query results
total_num	int	Total number of exported images
zippasswd	string	Compressed password

Example

Request example

```
{
  "msg_id":"2056",
  "snap_ids":"91f3705f-539b-4474-8f8b-fffb86d8ffa7"
}
```

Response example

```
{
  "code": 0,
  "data": {
    "export_url": "tmp/alarm_2019-01-23-14-35-29.zip",
    "failed_img_ids": "",
```

```
    "result_num": 1,
    "total_num": 1,
                    "zippasswd":"xxxx"
  },
  "msg": ""
}
```


3.38 Storing a single face image(base64Encoding) in a database

Interface description

Interface url	http://${ip}:${port}/api/json
Request Method	POST
Request Parameter Format	JSON
Interface Description	Adds a single face image to the specified face database.

Request parameter

Parameter	Type	Description	Required	Example
msg_id	string	“1051”	Yes	
lib_id	int	ID of the database in which the face image will be stored	Yes	
img	json	Face image to be stored	Yes	
img_id	string	ID of the face image to be stored, Identifier of the image. It can be customized and must be unique in the database. It may contain English letters, numbers, and hyphens (-) and is up to 48 characters in length.	No	
person_idcard	string	Identity code	No	

person_name	string	Name	No	
person_gender	string	Gender (0:female, 1:male)	No	
person_age	string	Age, between 0 and 999	No	
person_addr	string	Address	No	

Field information(img)

Parameter	Type	Description
filename	string	filename
data	string	base64 encoding data

Response parameter

Parameter	Type	Description
code	int	Result code (0:successful)
data	json	Returned data
msg	string	Result description

Field information(data)

Parameter	Type	Description
img_id	string	Image ID

Note 1: Image files only support the jpg / jpeg / png/bmp/tif formats.

Note 2: img_id uniquely identifies a face image and cannot be repeated. If it is not specified or is empty, an image ID will be automatically allocated.

Example

Request example

```
Input Files
{
   "msg_id":"1051",
   "lib_id":1,
   "img": {
            "filename":"haha.jpg",
            "data":""
   }
}
```

Response example

```
{
   "code": 0,
   "data": {
```

```
      "img_id": "test"
   },
   "msg": ""
}
```


3.39 Storing face images(base64encoding)in a database in batches

Interface description

Interface url	http://${ip}:${port}/api/json
Request Method	POST
Request Parameter Format	JSON
Interface Description	Stores face images in a database in batches

Request parameter

Parameter	Type	Description	Required	Example
msg_id	string	“1052”	Yes	
lib_id	int	Database ID	Yes	
imgs	json	Face images, img list	Yes	
uuid	string	Batch identifier	Yes	

Field information(img)

Parameter	Type	Description
filename	string	file name
data	string	base64 encoding data

Response parameter

Parameter	Type	Description
code	int	Result code (0:successful)
data	json	Failure response
msg	string	Result description

Field information(data)

Parameter	Type	Description
url	string	CSV file that contains information about database storage
err_count	int	Number of images that cannot be stored in the database

Note 1: The image files to be imported in a batch must be named in the format of Name_img_id_Identity_Gender_Age_Address.
Note 2: img_id is an attribute that identifies an image. It can be customized and must be unique in the database. It may contain English letters, numbers, and hyphens (-) and is up to 48 characters in length. Note 3: Gender can be set to 0 or 1. 0 indicates female and 1 indicates male.
Note 4: Give the attributes in each image filename in the specified order and separate attributes with underscores (_). The system automatically retrieves the image attributes. If an attribute is empty, fill it with a pound sign (#).

Example

Request example

```
Input Files
{
  "msg_id":"1052",
  "lib_id":1,
  "uuid":"xxyyzz",
  "imgs": [
    {
          "filename":"haha.jpg",
          "data":""
    }
  ]
}
```

Response example

```
{
  "code": 0,
  "data":
    { "err_count":
    0,
    "url": "tmp/image_.csv"
  },
  "msg": ""
}
```

3.40 Querying today's results

Interface description

Interface url	http://${ip}:${port}/api/json
Request Method	POST
Request Parameter Format	JSON
Interface Description	Querying today's results

Request parameter

Parameter	Type	Description	Required	Example
msg_id	string	“1050”	Yes	

Response parameter

Parameter	Type	Description
code	int	Result code (0:successful)
data	json	Returned data
msg	string	Result description

Field information(data)

Parameter	Type	Description
key_person_cnt	int	key person count
white_list_cnt	int	white list count
stranger_cnt	int	stranger count

Example

Request example

```
{
  "msg_id":"1050"
}
```

Response example

```
{
  "code": 0,
  "data": {  "key_person_cnt":
    1145,  "white_list_cnt":
    88,  "stranger_cnt": 44
  },
  "msg": ""
}
```

4 Camera-related interfaces

4.1 Querying a camera

Interface description

Interface url	http://${ip}:${port}/api/json
Request Method	POST
Request Parameter Format	JSON
Interface Description	Queries a camera

Request parameter

Parameter	Type	Description	Required	Example
msg_id	string	“516”	Yes	

Response parameter

Parameter	Type	Description
code	int	Result code (0:successful)
data	json	Returned data
msg	string	Result description

Field information(data)

Parameter	Type	Description
camera	json list	Camera
computing_power_capacity	int	The computing power capacity of the system; The computing power of the 1 channel capture camera is 1, and the computing power of the 1-channel IP camera is 2. The maximum computing power of the M4s is 8, and the maximum computing power of the M8s is 16.
used_computing_power	int	The used computing power of the systme currently

Field information(camera[RTSP])

Parameter	Type	Description
camera_name	string	Camera name
channel	int	Channel; value range:[1,16] of SenseNebula-M4s, [1,32] of SenseNebula-M8s
decoder	string	Protocol decoding method

camera_mode	string	Camera mode (VIDEO: applicable to IP cameras, IMAGE: applicable to capture cameras)
description	string	Camera description
lib_ids	string	Returned IDs of camera alert deployment databases in video mode (support multiple IDs returned)
vehicle_lib_ids	string	Returned IDs of camera alert deployment databases in video mode for vehicle detection (support multiple IDs returned)
position	string	Camera position
protocol	string	Camera protocol
status	int	Real-time camera status (1:online, 0:offline)
threshold	int	Threshold; value range:[0,100]
url	string	Camera URL
camera_roi	string	Region of interest of the camera (x1,y1,x2,y2 indicate the coordinates of the left, top, right, and bottom boundaries of the Region of interest, respectively.)
roi_max	int	Maximum area allowed in the camera Region of interest setting, which is calculated by using the following formula: (x2 – x1) x (y2 – y1)
snap_mode	int	Capture strategy (1:Accurate mode, 2:Timing mode, 3:Realtime mode)
facesize_min	int	Minimum face size that SenseNebula-M can capture and recognize. The range is [30,facesize_max], Default to 60 when the value is invalid.
facesize_max	int	Maximum face size that SenseNebula-M can capture and recognize. The range is [facesize_min, 1000], Default to 1000 when the value is invalid.
task_type	string	task type, eg: face,body,vehicle,abnormal,violate operate means simultaneously verifying face, body and vehicle
frame_interval	int	frame interval; value range:[3,25]
liveness_switch	int	liveness switch (0:disable, 1:enable)
liveness_threshold	int	liveness threshold; value range:[0,100]
body_sub_task	string	body task, (3:smoke, 4:doze, 5:call, 6:no helmet 7: no work cloth, 10:invade, 11:leave, 12: over boundary, 14:crowd), eg:3,5
abnormal_sub_task	string	abnormal task, (1:firesmoke), eg:1
violate_operate_sub_task	string	violate operate task, (1:fire extinguisher), eg:1
dup_alarm_gap	int	alarm gap, default:5 seconds
invade_area	string	body invade area: x1*y1,x2*y2,x3*y3
leave_area	string	body leave area: x1*y1,x2*y2,x3*y3
over_boundary_line	string	over boundary line: x1*y1,x2*y2

over_boundary_direction	int	over boundary direction: 1:left to right 2：right to left 3:double-sided
crowd_area	string	crowd area: x1*y1,x2*y2,x3*y3
crowd_threshold	int	crowd threshold, default:5
invade_time_threshold	int	invade time threshold
leave_time_threshold	int	leave time threshold
leave_num_threshold	int	leave num threshold
doze_time_threshold	int	doze time threshold
helmet_type	string	helmet type: bonnet,cap,bucket_hat,st_hard_hat
work_cloth_color	string	work cloth color:black,white,gray,red,yellow,blue,green,purple
work_cloth_type	string	work cloth type:st_worker_uniform,st_common_clothing,st_office_uniform,st_chef_uniform,st_medical_uniform,st_police_uniform,st_firefighter_uniform,st_reflective_clothing
smoke_alarm_gap	int	smoke alarm gap
call_alarm_gap	int	call alarm gap
helmet_alarm_gap	int	helmet alarm gap
work_cloth_alarm_gap	int	work cloth alarm gap
invade_alarm_gap	int	invade alarm gap
leave_alarm_gap	int	leave alarm gap
doze_alarm_gap	int	doze alarm gap
crowd_alarm_gap	int	crowd alarm gap
firesmoke_alarm_gap	int	firesmoke alarm gap
fireExtinguisher_alarm_gap	int	fireExtinguisher alarm gap
fireExtinguisher_area	string	fireExtinguisher roi area: x1*y1,x2*y2,x3*y3

fireExtinguisher_time_threshold	int	fireExtinguisher time threshold

Field information(camera[onvif])

Parameter	Type	Description
camera_name	string	Camera name
camera_pwd	int	Camera login password
camera_user	string	Camera login username
channel	int	Channel; value range:[1,16] of SenseNebula-M4s, [1,32] of SenseNebula-M8s
decoder	string	Decoding method
camera_mode	string	Camera mode
description	string	Camera description
ip	string	Camera IP address
lib_ids	string	Returned IDs of camera alert deployment databases in video mode.
vehicle_lib_ids	string	Returned IDs of camera alert deployment databases in video mode for vehicle detection (support multiple IDs returned)
port	int	Camera port number
position	string	Camera position
protocol	string	Camera protocol
status	int	Real-time camera status (1:online, 0:offline)
threshold	int	Threshold; value range:[0,100]
camera_roi	string	Region of interest of the camera (x1,y1,x2,y2 indicate the coordinates of the left, top, right, and bottom boundaries of the Region of interest, respectively.)
roi_max	int	Maximum area allowed in the camera Region of interest setting, which is calculated by using the following formula: (x2 – x1) x (y2 – y1)
snap_mode	int	Capture strategy (1:Accurate mode, 2:Timing mode, 3:Realtime mode)
facesize_min	int	Minimum face size that SenseNebula-M can capture and recognize. The range is [30,facesize_max], Default to 60 when the value is invalid.
facesize_max	int	Maximum face size that SenseNebula-M can capture and recognize. The range is [facesize_min, 1000], Default to 1000 when the value is invalid.
task_type	string	task type, eg: face,body,vehicle,abnormal,violate operate means simultaneously verifying face, body and vehicle
frame_interval	int	frame interval; value range:[3,25]
liveness_switch	int	liveness switch (0:disable, 1:enable)
liveness_threshold	int	liveness threshold; value range:[0,100]
body_sub_task	string	body task, (3:smoke, 4:doze, 5:call, 6:no helmet 7: no work cloth, 10:invade, 11:leave, 12: over boundary, 14:crowd), eg:3,5

abnormal_sub_task	string	abnormal task, (1:firesmoke), eg:1
violate_operate_sub_task	string	violate operate task, (1:fire extinguisher), eg:1
dup_alarm_gap	int	alarm gap, default:5 seconds
invade_area	string	body invade area: x1*y1,x2*y2,x3*y3
leave_area	string	body leave area: x1*y1,x2*y2,x3*y3
over_boundary_line	string	over boundary line: x1*y1,x2*y2
over_boundary_direction	int	over boundary direction: 1:left to right 2：right to left 3:double-sided
crowd_area	string	crowd area: x1*y1,x2*y2,x3*y3
crowd_threshold	int	crowd threshold, default:5
invade_time_threshold	int	invade time threshold
leave_time_threshold	int	leave time threshold
leave_num_threshold	int	leave num threshold
doze_time_threshold	int	doze time threshold
helmet_type	string	helmet type: bonnet,cap,bucket_hat,st_hard_hat
work_cloth_color	string	work cloth color:black,white,gray,red,yellow,blue,green,purple
work_cloth_ty pe	string	work cloth type:st_worker_uniform,st_common_clothing,st_office_uniform,st_chef_uniform,st_medical_uniform,st_police_uniform,st_firefighter_uniform,st_reflective_clothing
smoke_alarm_gap	int	smoke alarm gap
call_alarm_gap	int	call alarm gap
helmet_alarm_gap	int	helmet alarm gap
work_cloth_alarm_gap	int	work cloth alarm gap
invade_alarm_gap	int	invade alarm gap
leave_alarm_gap	int	leave alarm gap

doze_alarm_gap	int	doze alarm gap
crowd_alarm_gap	int	crowd alarm gap
firesmoke_alarm_gap	int	firesmoke alarm gap
fireExtinguisher_alarm_gap	int	fireExtinguisher alarm gap
fireExtinguisher_area	string	fireExtinguisher roi area: x1*y1,x2*y2,x3*y3
fireExtinguisher_time_threshold	int	fireExtinguisher time threshold

Field information(camera[GB28181])

Parameter	Type	Description
channel	int	Channel; value range:[1,16] of SenseNebula-M4s, [1,32] of SenseNebula-M8s
camera_mode	string	Camera mode
ip	string	SIP server IP
port	int	SIP server port, usually is 5060
server_sip_id	string	SIP server SIP ID, the length should be less than 21
camera_sip_id	string	Camera SIP ID, the length should be less than 21
camera_name	string	Camera name
camera_pwd	string	GB28181 authentication password for SIP server
description	string	Camera description
lib_ids	string	Returned IDs of camera alert deployment databases in video mode.
vehicle_lib_ids	string	Returned IDs of camera alert deployment databases in video mode for vehicle detection (support multiple IDs returned)
position	string	Camera position
protocol	string	Camera protocol , should be “GB28181”
status	int	Real-time camera status (1:online, 0:offline)
threshold	int	Threshold; value range:[0,100]
snap_mode	int	Capture strategy (1:Accurate mode, 2:Timing mode, 3:Realtime mode)
facesize_min	int	Minimum face size that SenseNebula-M can capture and recognize. The range is [30,facesize_max], Default to 60 when the value is invalid.
facesize_max	int	Maximum face size that SenseNebula-M can capture and recognize. The range is [facesize_min, 1000], Default to 1000 when the value is invalid.
task_type	string	task type, eg: face,body,vehicle,abnormal,violate operate means simultaneously verifying face, body and vehicle
frame_interval	int	frame interval; value range:[3,25]
liveness_switc	int	liveness switch (0:disable, 1:enable)

h		
liveness_threshold	int	liveness threshold; value range:[0,100]
body_sub_task	string	body task, (3:smoke, 4:doze, 5:call, 6:no helmet 7: no work cloth, 10:invade, 11:leave, 12: over boundary, 14:crowd), eg:3,5
abnormal_sub_task	string	abnormal task, (1:firesmoke), eg:1
violate_operate_sub_task	string	violate operate task, (1:fire extinguisher), eg:1
dup_alarm_gap	int	alarm gap, default:5 seconds
invade_area	string	body invade area: x1*y1,x2*y2,x3*y3
leave_area	string	body leave area: x1*y1,x2*y2,x3*y3
over_boundary_line	string	over boundary line: x1*y1,x2*y2
over_boundary_direction	int	over boundary direction: 1:left to right 2：right to left 3:double-sided
crowd_area	string	crowd area: x1*y1,x2*y2,x3*y3
crowd_threshold	int	crowd threshold, default:5
invade_time_threshold	int	invade time threshold
leave_time_threshold	int	leave time threshold
leave_num_threshold	int	leave num threshold
doze_time_threshold	int	doze time threshold
helmet_type	string	helmet type: bonnet,cap,bucket_hat,st_hard_hat
work_cloth_color	string	work cloth color:black,white,gray,red,yellow,blue,green,purple
work_cloth_ty pe	string	work cloth type:st_worker_uniform,st_common_clothing,st_office_uniform,st_chef_uniform,st_medical_uniform,st_police_uniform,st_firefighter_uniform,st_reflective_clothing
smoke_alarm_gap	int	smoke alarm gap
call_alarm_gap	int	call alarm gap
helmet_alarm_gap	int	helmet alarm gap
work_cloth_al	int	work cloth alarm gap

arm_gap		
invade_alarm_gap	int	invade alarm gap
leave_alarm_gap	int	leave alarm gap
doze_alarm_gap	int	doze alarm gap
crowd_alarm_gap	int	crowd alarm gap
firesmoke_alarm_gap	int	firesmoke alarm gap
fireExtinguisher_alarm_gap	int	fireExtinguisher alarm gap
fireExtinguisher_area	string	fireExtinguisher roi area: x1*y1,x2*y2,x3*y3
fireExtinguisher_time_threshold	int	fireExtinguisher time threshold

Field information(camera[vcn])

Parameter	Type	Description
channel	int	Channel; value range:[1,16] of SenseNebula-M4s, [1,32] of SenseNebula-M8s
camera_mode	string	Camera mode
camera_sip_id	string	Camera SIP ID, the length should be less than 21
camera_name	string	Camera name
description	string	Camera description
lib_ids	string	Returned IDs of camera alert deployment databases in video mode.
vehicle_lib_ids	string	Returned IDs of camera alert deployment databases in video mode for vehicle detection (support multiple IDs returned)
position	string	Camera position
protocol	string	Camera protocol , should be vcn
trd_login_user	string	vcn platform UserID
trd_login_pwd	string	vcn platform Password
trd_login_url	string	vcn platform login url
trd_fetch_stream_url	string	vcn platform fetch rtsp stream url
status	int	Real-time camera status (1:online, 0:offline)
threshold	int	Threshold; value range:[0,100]
snap_mode	int	Capture strategy (1:Accurate mode, 2:Timing mode, 3:Realtime mode)
facesize_min	int	Minimum face size that SenseNebula-M can capture and recognize. The range is [30,facesize_max], Default to 60 when the value is invalid.

facesize_max	int	Maximum face size that SenseNebula-M can capture and recognize. The range is [facesize_min, 1000], Default to 1000 when the value is invalid.
task_type	string	task type, eg: face,body,vehicle,abnormal,violate operate means simultaneously verifying face, body and vehicle
frame_interval	int	frame interval; value range:[3,25]
liveness_switch	int	liveness switch (0:disable, 1:enable)
liveness_threshold	int	liveness threshold; value range:[0,100]
body_sub_task	string	body task, (3:smoke, 4:doze, 5:call, 6:no helmet 7: no work cloth, 10:invade, 11:leave, 12: over boundary, 14:crowd), eg:3,5
abnormal_sub_task	string	abnormal task, (1:firesmoke), eg:1
violate_operate_sub_task	string	violate operate task, (1:fire extinguisher), eg:1
dup_alarm_gap	int	alarm gap, default:5 seconds
invade_area	string	body invade area: x1*y1,x2*y2,x3*y3
leave_area	string	body leave area: x1*y1,x2*y2,x3*y3
over_boundary_line	string	over boundary line: x1*y1,x2*y2
over_boundary_direction	int	over boundary direction: 1:left to right 2：right to left 3:double-sided
crowd_area	string	crowd area: x1*y1,x2*y2,x3*y3
crowd_threshold	int	crowd threshold, default:5
invade_time_threshold	int	invade time threshold
leave_time_threshold	int	leave time threshold
leave_num_threshold	int	leave num threshold
doze_time_threshold	int	doze time threshold
helmet_type	string	helmet type: bonnet,cap,bucket_hat,st_hard_hat
work_cloth_color	string	work cloth color:black,white,gray,red,yellow,blue,green,purple
work_cloth_ty pe	string	work cloth type:st_worker_uniform,st_common_clothing,st_office_uniform,st_chef_uniform,st_medical_uniform,st_police_uniform,st_firefighter_uniform,st_reflective_clothing
smoke_alarm_	int	smoke alarm gap

gap		
call_alarm_gap	int	call alarm gap
helmet_alarm_gap	int	helmet alarm gap
work_cloth_alarm_gap	int	work cloth alarm gap
invade_alarm_gap	int	invade alarm gap
leave_alarm_gap	int	leave alarm gap
doze_alarm_gap	int	doze alarm gap
crowd_alarm_gap	int	crowd alarm gap
firesmoke_alarm_gap	int	firesmoke alarm gap
fireExtinguisher_alarm_gap	int	fireExtinguisher alarm gap
fireExtinguisher_area	string	fireExtinguisher roi area: x1*y1,x2*y2,x3*y3
fireExtinguisher_time_threshold	int	fireExtinguisher time threshold

Field information(camera[image])

Parameter	Type	Description
camera_name	string	Camera name
camera_pwd	string	Camera login password
camera_user	string	Camera login username
channel	int	Channel; value range:[1,16] of SenseNebula-M4s, [1,32] of SenseNebula-M8s
description	string	Camera description
ip	string	Camera IP address
lib_ids	string	IDs of portrait databases (support multiple IDs returned)
port	int	Camera port number
position	string	Camera position
product	string	Camera version
status	int	Real-time camera status (1:online, 0:offline)

threshold	int	Threshold; value range:[0,100]
vendor	string	Camera manufacturers
camera_mode	string	Camera mode
task_type	string	task type, eg: face,body,vehicle,cyclist means simultaneously verifying face, body, vehicle and cyclist

Example

Request example

```
{
   "msg_id":"516"
}
```

Response example

```
{

   "code": 0,
   "data": {
      "camera": [
          {
            "camera_mode": "IMAGE",
            "camera_name": "123",
            "camera_pwd": "1",
            "camera_user": "1",
            "channel": 4,
            "description": "",
            "ip": "1.1.1.1",
            "lib_ids": "22",
            "port": 1,
            "position": "",
            "product": "",
            "status": 0,
            "threshold": 85,
            "vendor": "dahua",
            "task_type": "face,cyclist"
         },
         {
            "camera_mode": "VIDEO",
            "camera_name": "rtsp12",
            "camera_roi": "",
            "channel": 7,
            "decoder": "",
```

```
            "description": "",
            "facesize_max": 1000,
            "facesize_min": 60,
            "lib_ids": "221",
            "vehicle_lib_ids": "21",
            "position": "",
            "protocol": "rtsp",
            "roi_max": 2073600,
            "snap_mode": 3,
            "status": 1,
            "threshold": 75,
            "url": "rtsp://admin:admin123@10.151.116.112:554",
            "task_type": "face,body,vehicle",
            "frame_interval": 3,
            "body_sub_task":"3,5",
            "abnormal_sub_task":"1",
            "violate_operate_sub_task":"1",
            "dup_alarm_gap": 5,  "invade_area":
            "",  "fireExtinguisher_area": "",
            "leave_area": "",
            "over_boundary_direction": 0,
            "over_boundary_line": "",
            "crowd_area": "",
            "crowd_threshold": 10,
            "liveness_threshold": 0,
            "liveness_switch": 0
        }
    ],
     "computing_power_capacity": 16,
     "used_computing_power": 3
  },
  "msg": ""
}
```


Response packet that is returned when no camera is available:

Response example


```
{
  "code": 0,
  "data":
    {  "camera":
    [],
    "computing_power_capacity":  16,
    "used_computing_power": 0
```

```
  },
  "msg": ""
}
```

4.2 Adding a camera

Interface description

Interface url	http://${ip}:${port}/api/json
Request Method	POST
Request Parameter Format	JSON
Interface Description	Adds a Camera

a. Request parameter list for adding a camera in rtsp video mode:

Request parameter

Parameter	Type	Description	Required	Example
msg_id	string	“513”	Yes	
protocol	string	Camera-supported protocol (RTSP)	Yes	
decoder	string	Camera-supported decoding method	No	
camera_mode	string	Camera mode	Yes	
url	string	URL of the camera to be added	Yes	
position	string	Position of the camera to be added	No	
lib_ids	string	IDs of camera alert deployment databases to be added in video mode.	No	
vehicle_lib_ids	string	IDs of camera alert deployment databases to be added in video mode for vehicle detection.	No	
description	string	Description of the camera to be added	No	
threshold	int	Threshold; value range:[0,100]	Yes	
camera_name	string	Camera name	Yes	
snap_mode	int	Capture strategy (1:Accurate mode, 2:Timing mode, 3:Realtime mode)	Yes	
facesize_min	int	Minimum face size that SenseNebula-M can capture and recognize. The range is [30,facesize_max], Default to 60 when the value is invalid.	No	
facesize_max	int	Maximum face size that SenseNebula-M can capture and recognize. The range is [facesize_min, 1000], Default to 1000 when the value is invalid.	No	
task_type	string	Task type. Default to	No	face,body

		face,body,vehicle,cyclist,abnormal,violate operate when the value is invalid		,vehicle means simultaneously verifying face, body and vehicle
frame_interval	int	frame interval; value range:[3,25]	No	
liveness_switch	int	liveness switch (0:disable, 1:enable)	No	
liveness_threshold	int	liveness threshold; value range:[0,100]	No	
body_sub_task	string	body task, (3:smoke, 4:doze, 5:call, 6:no helmet 7: no work cloth, 10:invade, 11:leave, 12: over boundary, 14:crowd)	No	3,5
abnormal_sub_task	string	abnormal task, (1:firesmoke)	No	1
violate_operate_sub_task	string	violate operate task, (1:fire extinguisher)	No	1
dup_alarm_gap	int	alarm gap, default:5 seconds	No	5
invade_area	string	body invade area: x1*y1,x2*y2,x3*y3	No	0*0,1000*0,1000*500,0*500
leave_area	string	body leave area: x1*y1,x2*y2,x3*y3	No	0*0,1000*0,1000*500,0*500
over_boundary_line	string	over boundary line: x1*y1,x2*y2	No	0*500,1000*500
over_boundary_direction	int	over boundary direction: 1:left to right 2：right to left 3:double-sided	No	1
crowd_area	string	crowd area: x1*y1,x2*y2,x3*y3	No	0*0,1000*0,1000*500,0*500
crowd_threshold	int	crowd threshold, default:5	No	10
invade_time_threshold	int	invade time threshold	No	10
leave_time_	int	leave time threshold	No	10

threshold				
leave_num_threshold	int	leave num threshold	No	10
doze_time_threshold	int	doze time threshold	No	10
helmet_type	string	helmet type: bonnet,cap,bucket_hat,st_hard_hat	No	bonnet
work_cloth_color	string	work cloth color:black,white,gray,red,yellow,blue,green,purple	No	purple,green
work_cloth_type	string	work cloth type:st_worker_uniform,st_common_clothing,st_office_uniform,st_chef_uniform,st_medical_uniform,st_police_uniform,st_firefighter_uniform,st_reflective_clothing	No	st_reflective_clothing
smoke_alarm_gap	int	smoke alarm gap	No	
call_alarm_gap	int	call alarm gap	No	
helmet_alarm_gap	int	helmet alarm gap	No	
work_cloth_alarm_gap	int	work cloth alarm gap	No	
invade_alarm_gap	int	invade alarm gap	No	
leave_alarm_gap	int	leave alarm gap	No	
doze_alarm_gap	int	doze alarm gap	No	
crowd_alarm_gap	int	crowd alarm gap	No	
firesmoke_alarm_gap	int	firesmoke alarm gap	No	
fireExtinguisher_alarm_gap	int	fireExtinguisher alarm gap	No	
fireExtinguisher_area	string	fireExtinguisher roi area: x1*y1,x2*y2,x3*y3	No	0*0,1000*0,1000*500,0*500
fireExtinguisher_time_threshold	int	fireExtinguisher time threshold	No	10

Response parameter

Parameter	Type	Description
code	int	Result code (0:successful)
data	json	Returned data
msg	string	Result description

Field information(data)

Parameter	Type	Description
channel	int	Channel; value range:[1,16] of SenseNebula-M4s, [1,32] of SenseNebula-M8s

Example

Request example

```
{
  "camera_mode":"VIDEO",
  "msg_id": "513",
  "protocol":"rtsp",
  "url": "rtsp://admin:admin123@10.5.4.178:554",
  "position": "test",
  "lib_ids": "10",
  "vehicle_lib_ids":"2,10",
  "threshold":1,
  "camera_name":"test",
  "snap_mode":1,  "facesize_min":60,
  "facesize_max":1000,  "task_type":
  "face,body,vehicle",
  "frame_interval": 3,
  "body_sub_task":"3,5",
  "abnormal_sub_task":"1",
  "violate_operate_sub_task":"1",
  "dup_alarm_gap": 5,
  "invade_area": "",
  "fireExtinguisher_area": "",
  "leave_area": "",
  "over_boundary_direction": 0,
  "over_boundary_line": "",
  "crowd_area": "",
  "crowd_threshold": 10,
  "liveness_threshold": 0,
  "liveness_switch": 0
```

```
}
```

Response example

```
{
   "code": 0,
   "data":
      {  "channel": 1
   },
   "msg": ""
}
```

b. Request parameter list for adding a camera in onvif video mode:

Request parameter

Parameter	Type	Description	Required	Example
msg_id	string	“513”	Yes	
ip	string	IP address of the camera to be added	Yes	
port	int	Port number of the camera to be added	Yes	
camera_mode	string	Camera mode	Yes	
position	string	Position of the camera to be added	No	
camera_user	string	Login username of the camera to be added	Yes	
camera_pwd	string	Login password of the camera to be added	Yes	
protocol	string	Video stream protocol (ONVIF) of the camera	Yes	
decoder	string	Camera-supported decoding method	No	
lib_ids	string	IDs of camera alert deployment databases to be added in video mode.	No	
vehicle_lib_ids	string	IDs of camera alert deployment databases to be added in video mode for vehicle detection.	No	
description	string	Camera description	No	
threshold	int	Threshold; value range:[0,100]	Yes	
camera_name	string	Camera name	Yes	
snap_mode	int	Capture strategy (1:Accurate mode, 2:Timing mode, 3:Realtime mode)	Yes	
facesize_min	int	Minimum face size that SenseNebula-M can capture and recognize. The range is [30,facesize_max], Default to 60 when the value is invalid.	No	
facesize_max	int	Maximum face size that SenseNebula-M can capture and recognize. The range is [facesize_min, 1000], Default to	No	

		1000 when the value is invalid.		
task_type	string	Task type. Default to face,body,vehicle,cyclist,abnormal,violate operate when the value is invalid	No	face,body ,vehicle means simultane ously verifying face, body and vehicle
frame_inter val	int	frame interval; value range:[3,25]	No	
liveness_sw itch	int	liveness switch (0:disable, 1:enable)	No	
liveness_thr eshold	int	liveness threshold; value range:[0,100]	No	
body_sub_t ask	string	body task, (3:smoke, 4:doze, 5:call, 6:no helmet 7: no work cloth, 10:invade, 11:leave, 12: over boundary, 14:crowd)	No	3,5
abnormal_s ub_task	string	abnormal task, (1:firesmoke)	No	1
violate_ope rate_sub_ta sk	string	violate operate task, (1:fire extinguisher)	No	1
dup_alarm_ gap	int	alarm gap, default:5 seconds	No	5
invade_are a	string	body invade area: x1*y1,x2*y2,x3*y3	No	0*0,1000* 0,1000*50 0,0*500
leave_area	string	body leave area: x1*y1,x2*y2,x3*y3	No	0*0,1000* 0,1000*50 0,0*500
over_bound ary_line	string	over boundary line: x1*y1,x2*y2	No	0*500,100 0*500
over_bound ary_directio n	int	over boundary direction: 1:left to right 2： right to left 3:double-sided	No	1
crowd_area	string	crowd area: x1*y1,x2*y2,x3*y3	No	0*0,1000* 0,1000*50 0,0*500
crowd_thre shold	int	crowd threshold, default:5	No	10
invade_tim	int	invade time threshold	No	10

e_threshold				
leave_time_threshold	int	leave time threshold	No	10
leave_num_threshold	int	leave num threshold	No	10
doze_time_threshold	int	doze time threshold	No	10
helmet_type	string	helmet type: bonnet,cap,bucket_hat,st_hard_hat	No	bonnet
work_cloth_color	string	work cloth color:black,white,gray,red,yellow,blue,green,purple	No	purple,green
work_cloth_type	string	work cloth type:st_worker_uniform,st_common_clothing,st_office_uniform,st_chef_uniform,st_medical_uniform,st_police_uniform,st_firefighter_uniform,st_reflective_clothing	No	st_reflective_clothing
smoke_alarm_gap	int	smoke alarm gap	No	
call_alarm_gap	int	call alarm gap	No	
helmet_alarm_gap	int	helmet alarm gap	No	
work_cloth_alarm_gap	int	work cloth alarm gap	No	
invade_alarm_gap	int	invade alarm gap	No	
leave_alarm_gap	int	leave alarm gap	No	
doze_alarm_gap	int	doze alarm gap	No	
crowd_alarm_gap	int	crowd alarm gap	No	
firesmoke_alarm_gap	int	firesmoke alarm gap	No	
fireExtinguisher_alarm_gap	int	fireExtinguisher alarm gap	No	
fireExtinguisher_area	string	fireExtinguisher roi area: x1*y1,x2*y2,x3*y3	No	0*0,1000*0,1000*500,0*500
fireExtinguisher_time_threshold	int	fireExtinguisher time threshold	No	10

Response parameter

Parameter	Type	Description
code	int	Result code (0:successful)
data	json	Returned data
msg	string	Result description

Field information(data)

Parameter	Type	Description
channel	int	Channel; value range:[1,16] of SenseNebula-M4s, [1,32] of SenseNebula-M8s

Example

Request example

```
{
  "camera_mode":"VIDEO",
  "description":"test",
  "ip":"10.5.4.178",
  "lib_ids":"18",
  "port":80,
  "position":"test",
  "camera_pwd":"admin123",
  "camera_user":"admin",
  "decoder": "onvif",
  "protocol": "onvif",
  "threshold":1,  "msg_id":"513",
  "camera_name":"test",
  "snap_mode":1,
  "facesize_min":60,
  "facesize_max":1000,
  "task_type": "face,body,vehicle",
  "frame_interval": 3,
  "liveness_threshold": 0,
  "liveness_switch": 0
}
```

Response example

```
{
  "code": 0,
  "data":
    {  "channel": 2
  },
```

```
  "msg": ""
}
```

c. Request parameter list for adding a camera in GB28181 video mode:

When SenseNebula-M acts as a SIP server, the authentication password is sensetime.

Request parameter

Parameter	Type	Description	Required	Example
msg_id	string	“513”	Yes	
ip	string	SIP server IP	Yes	
port	int	SIP server port, usually is 5060	Yes	
camera_mode	string	Camera mode	Yes	
server_sip_id	string	SIP server SIP ID, the length should be less than 21	Yes	
camera_sip_id	string	Camera SIP ID, the length should be less than 21	Yes	
position	string	Camera position	No	
camera_pwd	string	GB28181 authentication password for SIP server	No	
protocol	string	Camera protocol (GB28181 Result code (0 indicates that the operation is successful.)	Yes	
lib_ids	string	IDs of camera alert deployment databases to be added in video mode.	No	
vehicle_lib_ids	string	IDs of camera alert deployment databases to be added in video mode for vehicle detection.	No	
description	string	Camera description	No	
threshold	int	Threshold; value range:[0,100]	Yes	
camera_name	string	Camera name	Yes	
snap_mode	int	Capture strategy (1:Accurate mode, 2:Timing mode, 3:Realtime mode)	Yes	
facesize_min	int	Minimum face size that SenseNebula-M can capture and recognize. The range is [30,facesize_max], Default to 60 when the value is invalid.	No	
facesize_max	int	Maximum face size that SenseNebula-M can capture and recognize. The range is [facesize_min, 1000], Default to 1000 when the value is invalid.	No	
task_type	string	Task type. Default to face,body,vehicle,cyclist,abnormal,violate operate when the value is invalid	No	face,body,vehicle means

				simultaneously verifying face, body and vehicle
frame_interval	int	frame interval; value range:[3,25]	No	
liveness_switch	int	liveness switch (0:disable, 1:enable)	No	
liveness_threshold	int	liveness threshold; value range:[0,100]	No	
body_sub_task	string	body task, (3:smoke, 4:doze, 5:call, 6:no helmet 7: no work cloth, 10:invade, 11:leave, 12: over boundary, 14:crowd)	No	3,5
abnormal_sub_task	string	abnormal task, (1:firesmoke)	No	1
violate_operate_sub_task	string	violate operate task, (1:fire extinguisher)	No	1
dup_alarm_gap	int	alarm gap, default:5 seconds	No	5
invade_area	string	body invade area: x1*y1,x2*y2,x3*y3	No	0*0,1000*0,1000*500,0*500
leave_area	string	body leave area: x1*y1,x2*y2,x3*y3	No	0*0,1000*0,1000*500,0*500
over_boundary_line	string	over boundary line: x1*y1,x2*y2	No	0*500,1000*500
over_boundary_direction	int	over boundary direction: 1:left to right 2：right to left 3:double-sided	No	1
crowd_area	string	crowd area: x1*y1,x2*y2,x3*y3	No	0*0,1000*0,1000*500,0*500
crowd_threshold	int	crowd threshold, default:5	No	10
invade_time_threshold	int	invade time threshold	No	10
leave_time_threshold	int	leave time threshold	No	10
leave_num	int	leave num threshold	No	10

_threshold				
doze_time_threshold	int	doze time threshold	No	10
helmet_type	string	helmet type: bonnet,cap,bucket_hat,st_hard_hat	No	bonnet
work_cloth_color	string	work cloth color:black,white,gray,red,yellow,blue,green,purple	No	purple,green
work_cloth_type	string	work cloth type:st_worker_uniform,st_common_clothing,st_office_uniform,st_chef_uniform,st_medical_uniform,st_police_uniform,st_firefighter_uniform,st_reflective_clothing	No	st_reflective_clothing
smoke_alarm_gap	int	smoke alarm gap	No	
call_alarm_gap	int	call alarm gap	No	
helmet_alarm_gap	int	helmet alarm gap	No	
work_cloth_alarm_gap	int	work cloth alarm gap	No	
invade_alarm_gap	int	invade alarm gap	No	
leave_alarm_gap	int	leave alarm gap	No	
doze_alarm_gap	int	doze alarm gap	No	
crowd_alarm_gap	int	crowd alarm gap	No	
firesmoke_alarm_gap	int	firesmoke alarm gap	No	
fireExtinguisher_alarm_gap	int	fireExtinguisher alarm gap	No	
fireExtinguisher_area	string	fireExtinguisher roi area: x1*y1,x2*y2,x3*y3	No	0*0,1000*0,1000*500,0*500
fireExtinguisher_time_threshold	int	fireExtinguisher time threshold	No	10

Response parameter

Parameter	Type	Description
code	int	Result code (0:successful)

data	json	Returned data
msg	string	Result description

Field information(data)

Parameter	Type	Description
channel	int	Channel; value range:[1,16] of SenseNebula-M4s, [1,32] of SenseNebula-M8s

Example

Request example

```
{
   "camera_mode":"VIDEO",
   "description":"test",
   "ip":"10.5.4.178",
   "lib_ids":"18",
   "port":80,
   "position":"test",
   "camera_pwd":"admin123",
   "server_sip_id":"34020000002000000001",
   "camera_sip_id":"34020000001320000001",
   "protocol": "gb28181",
   "threshold":1,
   "msg_id":"513",
   "camera_name":"test",
   "snap_mode":"1",
   "facesize_min":60,
   "facesize_max":1000,
   "task_type": "face",
   "frame_interval": 3,
   "liveness_threshold": 0,
   "liveness_switch": 0
}
```

Response example

```
{
   "code": 0,
   "data":
      {  "channel": 2
   },
   "msg": ""
}
```

d. Request parameter list for adding a camera in vcn video mode:

Request parameter

Parameter	Type	Description	Required	Example
msg_id	string	“513”	Yes	
camera_mode	string	Camera mode	Yes	
camera_sip_id	string	Camera SIP ID, the length should be less than 21	Yes	
position	string	Camera position	No	
protocol	string	Camera protocol (GB28181 Result code (0 indicates that the operation is successful.)	Yes	
trd_login_user	string	vcn platform UserId	Yes	nablaworks
trd_login_pwd	string	vcn platform Password	Yes	nablaworks
trd_login_url	string	vcn platform login url	Yes	http://ip:port/rtsp/login
trd_fetch_stream_url	string	vcn platform rtsp stream fetch url	Yes	http://ip:port/rtsp/url
lib_ids	string	IDs of camera alert deployment databases to be added in video mode.	No	
vehicle_lib_ids	string	IDs of camera alert deployment databases to be added in video mode for vehicle detection.	No	
description	string	Camera description	No	
threshold	int	Threshold; value range:[0,100]	Yes	
camera_name	string	Camera name	Yes	
snap_mode	int	Capture strategy (1:Accurate mode, 2:Timing mode, 3:Realtime mode)	Yes	
facesize_min	int	Minimum face size that SenseNebula-M can capture and recognize. The range is [30,facesize_max], Default to 60 when the value is invalid.	No	
facesize_max	int	Maximum face size that SenseNebula-M can capture and recognize. The range is [facesize_min, 1000], Default to 1000 when the value is invalid.	No	
task_type	string	Task type. Default to face,body,vehicle,cyclist,abnormal,violate operate when the value is invalid	No	face,body,vehicle means simultane

				ously verifying face, body and vehicle
frame_interval	int	frame interval; value range:[3,25]	No	
liveness_switch	int	liveness switch (0:disable, 1:enable)	No	
liveness_threshold	int	liveness threshold; value range:[0,100]	No	
body_sub_task	string	body task, (3:smoke, 4:doze, 5:call, 6:no helmet 7: no work cloth, 10:invade, 11:leave, 12: over boundary, 14:crowd)	No	3,5
abnormal_sub_task	string	abnormal task, (1:firesmoke)	No	1
violate_operate_sub_task	string	violate operate task, (1:fire extinguisher)	No	1
dup_alarm_gap	int	alarm gap, default:5 seconds	No	5
invade_area	string	body invade area: x1*y1,x2*y2,x3*y3	No	0*0,1000*0,1000*500,0*500
leave_area	string	body leave area: x1*y1,x2*y2,x3*y3	No	0*0,1000*0,1000*500,0*500
over_boundary_line	string	over boundary line: x1*y1,x2*y2	No	0*500,1000*500
over_boundary_direction	int	over boundary direction: 1:left to right 2： right to left 3:double-sided	No	1
crowd_area	string	crowd area: x1*y1,x2*y2,x3*y3	No	0*0,1000*0,1000*500,0*500
crowd_threshold	int	crowd threshold, default:5	No	10
invade_time_threshold	int	invade time threshold	No	10
leave_time_threshold	int	leave time threshold	No	10
leave_num_threshold	int	leave num threshold	No	10

doze_time_threshold	int	doze time threshold	No	10
helmet_type	string	helmet type: bonnet,cap,bucket_hat,st_hard_hat	No	bonnet
work_cloth_color	string	work cloth color:black,white,gray,red,yellow,blue,green,purple	No	purple,green
work_cloth_type	string	work cloth type:st_worker_uniform,st_common_clothing,st_office_uniform,st_chef_uniform,st_medical_uniform,st_police_uniform,st_firefighter_uniform,st_reflective_clothing	No	st_reflective_clothing
smoke_alarm_gap	int	smoke alarm gap	No	
call_alarm_gap	int	call alarm gap	No	
helmet_alarm_gap	int	helmet alarm gap	No	
work_cloth_alarm_gap	int	work cloth alarm gap	No	
invade_alarm_gap	int	invade alarm gap	No	
leave_alarm_gap	int	leave alarm gap	No	
doze_alarm_gap	int	doze alarm gap	No	
crowd_alarm_gap	int	crowd alarm gap	No	
firesmoke_alarm_gap	int	firesmoke alarm gap	No	
fireExtinguisher_alarm_gap	int	fireExtinguisher alarm gap	No	
fireExtinguisher_area	string	fireExtinguisher roi area: x1*y1,x2*y2,x3*y3	No	0*0,1000*0,1000*500,0*500
fireExtinguisher_time_threshold	int	fireExtinguisher time threshold	No	10

Response parameter

Parameter	Type	Description
code	int	Result code (0:successful)
data	json	Returned data

msg	string	Result description

Field information(data)

Parameter	Type	Description
channel	int	Channel; value range:[1,16] of SenseNebula-M4s, [1,32] of SenseNebula-M8s

Example

Request example

```
{
   "camera_mode":"VIDEO",
   "protocol": "vcn",
   "camera_name": "vcn1",
   "position": "",
   "snap_mode": 2,
   "camera_sip_id": "34000001001320000005",
   "lib_ids": "",
   "vehicle_lib_ids": "",
   "threshold": 85,
   "task_type": "face",
   "frame_interval": 5,
   "trd_login_user": "nablaworks",
   "trd_login_pwd":  "nablaworks",
   "trd_login_url": "http://10.151.144.84:8888/rtsp/login",
   "trd_fetch_stream_url": "http://10.151.144.84:8888/rtsp/url",
   "liveness_threshold": 0,
   "liveness_switch": 0,
   "msg_id": "513"
}
```

Response example

```
{
   "code": 0,
   "data":
      {  "channel": 2
   },
   "msg": ""
}
```

e. Request parameter list for adding a camera in image mode:

Request parameter

Parameter	Type	Description	Requir	Example
			ed	

msg_id	string	“513”	Yes	
ip	string	IP address of the camera to be added	Yes	
port	int	Port number of the camera to be added	Yes	
camera_mode	string	Camera mode	Yes	
position	string	Position of the camera to be added	No	
camera_user	string	Login username of the camera to be added	Yes	
camera_pwd	string	Login password of the camera to be added	Yes	
vendor	string	Camera manufacturers	No	Support hikvision,sensedlc11,sensedlct,sensedlcd,dahuas,dahua; No or wrong filling is default to dahua.
product	string	Camera version	No	
lib_ids	string	Returned IDs of camera alert deployment databases	No	
description	string	Camera description	No	
threshold	int	Threshold; value range:[0,100]	Yes	
camera_name	string	Camera name	Yes	
task_type	string	Task type. Default to face,body,vehicle, cyclist when the value is invalid	No	face,body,vehicle means simultaneously verifying face, body and vehicle

Response parameter

Parameter	Type	Description
code	int	Result code (0:successful)

data	json	Returned data
msg	string	Result description

Field information(data)

Parameter	Type	Description
channel	int	Channel; value range:[1,16] of SenseNebula-M4s, [1,32] of SenseNebula-M8s

Example

Request example

```
{
   "camera_mode":"IMAGE",
   "description":"test",
   "ip":"10.5.4.178",
   "lib_ids":"18",
   "port":37777,
   "position":"test",
   "camera_pwd":"admin123",
   "camera_user":"admin",
   "vendor": "dahua",
   "product": "01",
   "threshold":1,  "msg_id":"513",
   "camera_name":"test",
   "task_type": "face"
}
```

Response example

```
{
   "code": 0,
   "data":
      {  "channel": 2
   },
   "msg": ""
}
```

Note 1: Currently, cameras of video streams only support RTSP, ONVIF, GB28181 and vcn. The decoder and protocol parameters are temporarily reserved.

Note 2: The lib_ids parameter is in the format "lib1,lib2,lib3,...". Separate multiple IDs with commas (,).

Note 3: The channel parameter value ranges from 1 to 8 for SenseNebula-M4s and from 1 to 16 for SenseNebula-M8s.

Note 4: When the entered camera IP address is incorrect or invalid, wait until the server responds.

Note 5: The vendor field can be filled with hikvision, sensedlc11, sensedlct, sensedlcd, dahuas, dahua, which respectively represent the brand of capture cameras of Hikvision, SenseDLC 11 , SenseDLC T series, SenseDLC D series , and Dahua/SenseDLC AA series.

4.3 Modifying a camera

Interface description

Interface url	http://${ip}:${port}/api/json
Request Method	POST
Request Parameter Format	JSON
Interface Description	Modifies a camera.

a. Request parameter list for modifying a camera in rtsp video mode:

Request parameter

Parameter	Type	Description	Required	Example
msg_id	string	“515”	Yes	
url	string	Camera URL after modification	No	
channel	int	Channel; value range:[1,16] of SenseNebula-M4s, [1,32] of SenseNebula-M8s	Yes	
position	string	Camera position after modification	No	
lib_ids	string	IDs of camera alert deployment databases after modification in video mode.	No	
vehicle_lib_ids	string	IDs of camera alert deployment databases after modification in video mode for vehicle detection.	No	
camera_roi	string	Camera Region of interest after modification	No	
description	string	Camera description after modification	No	
threshold	int	Similarity threshold after modification; value range:[0,100]	No	
camera_name	string	Camera name after modification	No	
snap_mode	int	Capture strategy (1:Accurate mode, 2:Timing mode, 3:Realtime mode)	No	
facesize_min	int	Minimum face size that SenseNebula-M can capture and recognize. The range is [30,facesize_max], Default to 60 when the value is invalid.	No	
facesize_max	int	Maximum face size that SenseNebula-M can capture and recognize. The range is [facesize_min, 1000], Default to 1000 when the value is invalid.	No	
task_type	string	Task type. Default to face,body,vehicle,cyclist,abnormal,violate operate when	No	face,body,vehicle

		the value is invalid		means simultaneously verifying face, body and vehicle
frame_interval	int	frame interval; value range:[3,25]	No	
liveness_switch	int	liveness switch (0:disable, 1:enable)	No	
liveness_threshold	int	liveness threshold; value range:[0,100]	No	
body_sub_task	string	body task, (3:smoke, 4:doze, 5:call, 6:no helmet 7: no work cloth, 10:invade, 11:leave, 12: over boundary, 14:crowd)	No	3,5
abnormal_sub_task	string	abnormal task, (1:firesmoke)	No	1
violate_operate_sub_task	string	violate operate task, (1:fire extinguisher)	No	1
dup_alarm_gap	int	alarm gap, default:5 seconds	No	5
invade_area	string	body invade area: x1*y1,x2*y2,x3*y3	No	0*0,1000*0,1000*500,0*500
leave_area	string	body leave area: x1*y1,x2*y2,x3*y3	No	0*0,1000*0,1000*500,0*500
over_boundary_line	string	over boundary line: x1*y1,x2*y2	No	0*500,1000*500
over_boundary_direction	int	over boundary direction: 1:left to right 2：right to left 3:double-sided	No	1
crowd_area	string	crowd area: x1*y1,x2*y2,x3*y3	No	0*0,1000*0,1000*500,0*500
crowd_threshold	int	crowd threshold, default:5	No	10
invade_time_threshold	int	invade time threshold	No	10
leave_time_threshold	int	leave time threshold	No	10

leave_num_threshold	int	leave num threshold	No	10
doze_time_threshold	int	doze time threshold	No	10
helmet_type	string	helmet type: bonnet,cap,bucket_hat,st_hard_hat	No	bonnet
work_cloth_color	string	work cloth color:black,white,gray,red,yellow,blue,green,purple	No	purple,green
work_cloth_type	string	work cloth type:st_worker_uniform,st_common_clothing,st_office_uniform,st_chef_uniform,st_medical_uniform,st_police_uniform,st_firefighter_uniform,st_reflective_clothing	No	st_reflective_clothing
smoke_alarm_gap	int	smoke alarm gap	No	
call_alarm_gap	int	call alarm gap	No	
helmet_alarm_gap	int	helmet alarm gap	No	
work_cloth_alarm_gap	int	work cloth alarm gap	No	
invade_alarm_gap	int	invade alarm gap	No	
leave_alarm_gap	int	leave alarm gap	No	
doze_alarm_gap	int	doze alarm gap	No	
crowd_alarm_gap	int	crowd alarm gap	No	
firesmoke_alarm_gap	int	firesmoke alarm gap	No	
fireExtinguisher_alarm_gap	int	fireExtinguisher alarm gap	No	
fireExtinguisher_area	string	fireExtinguisher roi area: x1*y1,x2*y2,x3*y3	No	0*0,1000*0,1000*500,0*500
fireExtinguisher_time_threshold	int	fireExtinguisher time threshold	No	10

Response parameter

Parameter	Type	Description

code	int	Result code (0:successful)
data	json	Returned data
msg	string	Result description

Field information(data)

Parameter	Type	Description
channel	int	Channel; value range:[1,16] of SenseNebula-M4s, [1,32] of SenseNebula-M8s

Example

Request example

```
{
  "msg_id":                                  "515",   "url":
  "rtsp://admin:admin123@10.5.4.178:554",
  "channel": 1,
  "position": "test",
  "lib_ids": "1",
  "threshold":1,
  "camera_name":"test",
  "description":"home",
  "camera_roi":"0,180,1920,900",
  "snap_mode":1,  "facesize_min":60,
  "facesize_max":1000,  "task_type":
  "face,body,vehicle",
  "frame_interval": 3,
  "body_sub_task":"3,5",
  "abnormal_sub_task":"1",
  "violate_operate_sub_task":"1",
  "dup_alarm_gap": 5,
  "invade_area": "",
  "fireExtinguisher_area": "",
  "leave_area": "",
  "over_boundary_direction": 0,
  "over_boundary_line": "",
  "crowd_area": "",
  "crowd_threshold": 10,
  "liveness_threshold": 0,
  "liveness_switch": 0
}
```

Response example

```
{
  "code": 0,
  "data":
    {  "channel": 1
  },
  "msg": ""
}
```

b. Request parameter list for modifying a camera in onvif video mode:

Request parameter

Parameter	Type	Description	Required	Example
msg_id	string	“515”	Yes	
ip	string	Camera IP address after modification	No	
port	int	Camera port number after modification	No	
channel	int	Channel; value range:[1,16] of SenseNebula-M4s, [1,32] of SenseNebula-M8s	Yes	
position	string	Camera position after modification	No	
camera_user	string	Camera login username after modification	No	
camera_pwd	string	Camera login password after modification	No	
lib_ids	string	IDs of camera alert deployment databases after modification in video mode.	No	
vehicle_lib_ids	string	IDs of camera alert deployment databases after modification in video mode for vehicle detection.	No	
camera_roi	string	Camera Region of interest after modification	No	
description	string	Camera description after modification	No	
threshold	int	Similarity threshold after modification; value range:[0,100]	No	
camera_name	string	Camera name after modification	No	
snap_mode	int	Capture strategy (1:Accurate	No	

		mode, 2:Timing mode, 3:Realtime mode)		
facesize_min	int	Minimum face size that SenseNebula-M can capture and recognize. The range is [30,facesize_max], Default to 60 when the value is invalid.	No	
facesize_max	int	Maximum face size that SenseNebula-M can capture and recognize. The range is [facesize_min, 1000], Default to 1000 when the value is invalid.	No	
task_type	string	Task type. Default to face,body,vehicle,cyclist,abnormal,violate operate when the value is invalid	No	face,body,vehicle means simultaneously verifying face, body and vehicle
frame_interval	int	frame interval; value range:[3,25]	No	
liveness_switch	int	liveness switch (0:disable, 1:enable)	No	
liveness_threshold	int	liveness threshold; value range:[0,100]	No	
body_sub_task	string	body task, (3:smoke, 4:doze, 5:call, 6:no helmet 7: no work cloth, 10:invade, 11:leave, 12: over boundary, 14:crowd)	No	3,5
abnormal_sub_task	string	abnormal task, (1:firesmoke)	No	1
violate_operate_sub_task	string	violate operate task, (1:fire extinguisher)	No	1
dup_alarm_gap	int	alarm gap, default:5 seconds	No	5
invade_area	string	body invade area: x1*y1,x2*y2,x3*y3	No	0*0,1000*0,1000*500, 0*500
leave_area	string	body leave area: x1*y1,x2*y2,x3*y3	No	0*0,1000*0,1000*500, 0*500
over_boundary_line	string	over boundary line: x1*y1,x2*y2	No	0*500,1000*500
over_boundary_direction	int	over boundary direction: 1:left to right 2： right to left 3:double-sided	No	1
crowd_area	string	crowd area: x1*y1,x2*y2,x3*y3	No	0*0,1000*0,1000*500, 0*500
crowd_threshold	int	crowd threshold, default:5	No	10

smoke_alarm_gap	int	smoke alarm gap	No	
call_alarm_gap	int	call alarm gap	No	
helmet_alarm_gap	int	helmet alarm gap	No	
work_cloth_alarm_gap	int	work cloth alarm gap	No	
invade_alarm_gap	int	invade alarm gap	No	
leave_alarm_gap	int	leave alarm gap	No	
doze_alarm_gap	int	doze alarm gap	No	
crowd_alarm_gap	int	crowd alarm gap	No	
firesmoke_alarm_gap	int	firesmoke alarm gap	No	
fireExtinguisher_alarm_gap	int	fireExtinguisher alarm gap	No	
fireExtinguisher_area	string	fireExtinguisher roi area: x1*y1,x2*y2,x3*y3	No	0*0,1000*0,1000*500,0*500

Response parameter

Parameter	Type	Description
code	int	Result code (0:successful)
data	json	Returned data
msg	string	Result description

Field information(data)

Parameter	Type	Description
channel	int	Channel; value range:[1,16] of SenseNebula-M4s, [1,32] of SenseNebula-M8s

Example

Request example

```
{
  "channel":2,
  "description":"test",
  "ip":"10.5.4.178",
  "lib_ids":"1",
  "port":80,
  "position":"test",
  "camera_pwd":"admin123",
  "camera_user":"admin",
  "threshold":1,  "msg_id":"515",
  "camera_name":"test",
  "camera_roi":"0,130,1020,700",
```

```
  "snap_mode":1,  "facesize_min":60,
  "facesize_max":1000,  "task_type":
  "face,body,vehicle",
  "frame_interval": 3,
  "liveness_threshold": 0,
  "liveness_switch": 0
}
```

Response example

```
{
  "code": 0,
  "data":
    {  "channel": 2
  },
  "msg": ""
}
```

c. Request parameter list for modifying a camera in GB28181 video mode:

Request parameter

Parameter	Type	Description	Required	Example
msg_id	string	“515”	Yes	
ip	string	SIP server IP	No	
port	int	SIP server port, usually is 5060	No	
server_sip_id	string	SIP server SIP ID, the length should be less than 21	No	
camera_sip_id	string	Camera SIP ID, the length should be less than 21	No	
channel	int	Channel; value range:[1,16] of SenseNebula-M4s, [1,32] of SenseNebula-M8s	Yes	
position	string	Camera position	No	
camera_pwd	string	GB28181 authentication password for SIP server	No	
lib_ids	string	IDs of camera alert deployment databases to be added in video mode.	No	
vehicle_lib_ids	string	IDs of camera alert deployment databases after modification in video mode for vehicle detection.	No	
description	string	Camera description	No	
threshold	int	Threshold; value range:[0,100]	No	

camera_name	string	Camera name	No	
snap_mode	int	Capture strategy (1:Accurate mode, 2:Timing mode, 3:Realtime mode)	No	
facesize_min	int	Minimum face size that SenseNebula-M can capture and recognize. The range is [30,facesize_max], Default to 60 when the value is invalid.	No	
facesize_max	int	Maximum face size that SenseNebula-M can capture and recognize. The range is [facesize_min, 1000], Default to 1000 when the value is invalid.	No	
task_type	string	Task type. Default to face,body,vehicle,cyclist,abnormal,fire extinguisher when the value is invalid	No	face,body,vehicle means simultaneously verifying face, body and vehicle
frame_interval	int	frame interval; value range:[3,25]	No	
liveness_switch	int	liveness switch (0:disable, 1:enable)	No	
liveness_threshold	int	liveness threshold; value range:[0,100]	No	
body_sub_task	string	body task, (3:smoke, 4:doze, 5:call, 6:no helmet 7: no work cloth, 10:invade, 11:leave, 12: over boundary, 14:crowd)	No	3,5
abnormal_sub_task	string	abnormal task, (1:firesmoke)	No	1
violate_operate_sub_task	string	violate operate task, (1:fire extinguisher)	No	1
dup_alarm_gap	int	alarm gap, default:5 seconds	No	5
invade_area	string	body invade area: x1*y1,x2*y2,x3*y3	No	0*0,1000*0,1000*500,0*500
leave_area	string	body leave area: x1*y1,x2*y2,x3*y3	No	0*0,1000*0,1000*500,0*500
over_boundary_line	string	over boundary line: x1*y1,x2*y2	No	0*500,1000*500
over_boundary_direction	int	over boundary direction: 1:left to right 2：right to left 3:double-sided	No	1
crowd_area	string	crowd area: x1*y1,x2*y2,x3*y3	No	0*0,1000*0,1000*500,0*500

crowd_threshold	int	crowd threshold, default:5	No	10
smoke_alarm_gap	int	smoke alarm gap	No	
call_alarm_gap	int	call alarm gap	No	
helmet_alarm_gap	int	helmet alarm gap	No	
work_cloth_alarm_gap	int	work cloth alarm gap	No	
invade_alarm_gap	int	invade alarm gap	No	
leave_alarm_gap	int	leave alarm gap	No	
doze_alarm_gap	int	doze alarm gap	No	
crowd_alarm_gap	int	crowd alarm gap	No	
firesmoke_alarm_gap	int	firesmoke alarm gap	No	
fireExtinguisher_alarm_gap	int	fireExtinguisher alarm gap	No	
fireExtinguisher_area	string	fireExtinguisher roi area: x1*y1,x2*y2,x3*y3	No	0*0,1000*0,1000*500,0*500

Response parameter

Parameter	Type	Description
code	int	Result code (0:successful)
data	json	Returned data
msg	string	Result description

Field information(data)

Parameter	Type	Description
channel	int	Channel; value range:[1,16] of SenseNebula-M4s, [1,32] of SenseNebula-M8s

Example

Request example

```
{
  "channel":2,
  "description":"test",
  "ip":"10.5.4.178",
  "lib_ids":"18",
  "port":80,
  "position":"test",
  "camera_pwd":"admin123",
  "server_sip_id":"34020000002000000001",
  "camera_sip_id":"34020000001320000001",
  "protocol":"gb28181",
  "threshold":1,
```

```
  "msg_id":"515",
  "camera_name":"test",
  "snap_mode":1,  "facesize_min":60,
  "facesize_max":1000,  "task_type":
  "face,body,vehicle",
  "frame_interval": 3,
  "liveness_threshold": 0,
  "liveness_switch": 0
}
```

Response example

```
{
  "code": 0,
  "data":
    {  "channel": 2
  },
  "msg": ""
}
```

d. Request parameter list for modifying a camera in vcn video mode:

Request parameter

Parameter	Type	Description	Required	Example
msg_id	string	“515”	Yes	
camera_sip_id	string	Camera SIP ID, the length should be less than 21	No	
channel	int	Channel; value range:[1,16] of SenseNebula-M4s, [1,32] of SenseNebula-M8s	Yes	
position	string	Camera position	No	
trd_login_user	string	vcn platform UserId	Yes	nablaworks
trd_login_pwd	string	vcn platform Password	Yes	nablaworks
trd_login_url	string	vcn platform login url	Yes	http://ip:port/rtsp/login
trd_fetch_stream_url	string	vcn platform rtsp stream fetch url	Yes	http://ip:port/rtsp/url
lib_ids	string	IDs of camera alert deployment databases to be added in video mode.	No	
vehicle_lib_ids	string	IDs of camera alert deployment databases after modification in video mode for vehicle	No	

		detection.		
description	string	Camera description	No	
threshold	int	Threshold; value range:[0,100]	No	
camera_name	string	Camera name	No	
snap_mode	int	Capture strategy (1:Accurate mode, 2:Timing mode, 3:Realtime mode)	No	
facesize_min	int	Minimum face size that SenseNebula-M can capture and recognize. The range is [30,facesize_max], Default to 60 when the value is invalid.	No	
facesize_max	int	Maximum face size that SenseNebula-M can capture and recognize. The range is [facesize_min, 1000], Default to 1000 when the value is invalid.	No	
task_type	string	Task type. Default to face,body,vehicle,cyclist,abnormal,violate operate when the value is invalid	No	face,body,vehicle means simultaneously verifying face, body and vehicle
frame_interval	int	frame interval; value range:[3,25]	No	
liveness_switch	int	liveness switch (0:disable, 1:enable)	No	
liveness_threshold	int	liveness threshold; value range:[0,100]	No	
body_sub_task	string	body task, (3:smoke, 4:doze, 5:call, 6:no helmet 7: no work cloth, 10:invade, 11:leave, 12: over boundary, 14:crowd)	No	3,5
abnormal_sub_task	string	abnormal task, (1:firesmoke)	No	1
violate_operate_sub_task	string	violate operate task, (1:fire extinguisher)	No	1
dup_alarm_gap	int	alarm gap, default:5 seconds	No	5
invade_area	string	body invade area: x1*y1,x2*y2,x3*y3	No	0*0,1000*0,1000*500, 0*500
leave_area	string	body leave area: x1*y1,x2*y2,x3*y3	No	0*0,1000*0,1000*500, 0*500
over_boundary_line	string	over boundary line: x1*y1,x2*y2	No	0*500,1000*500
over_boundary_direct	int	over boundary direction: 1:left	No	1

ion		to right 2： right to left 3:double-sided		
crowd_area	string	crowd area: x1*y1,x2*y2,x3*y3	No	0*0,1000*0,1000*500,0*500
crowd_threshold	int	crowd threshold, default:5	No	10
smoke_alarm_gap	int	smoke alarm gap	No	
call_alarm_gap	int	call alarm gap	No	
helmet_alarm_gap	int	helmet alarm gap	No	
work_cloth_alarm_gap	int	work cloth alarm gap	No	
invade_alarm_gap	int	invade alarm gap	No	
leave_alarm_gap	int	leave alarm gap	No	
doze_alarm_gap	int	doze alarm gap	No	
crowd_alarm_gap	int	crowd alarm gap	No	
firesmoke_alarm_gap	int	firesmoke alarm gap	No	
fireExtinguisher_alarm_gap	int	fireExtinguisher alarm gap	No	
fireExtinguisher_area	string	fireExtinguisher roi area: x1*y1,x2*y2,x3*y3	No	0*0,1000*0,1000*500,0*500

Response parameter

Parameter	Type	Description
code	int	Result code (0:successful)
data	json	Returned data
msg	string	Result description

Field information(data)

Parameter	Type	Description
channel	int	Channel; value range:[1,16] of SenseNebula-M4s, [1,32] of SenseNebula-M8s

Example

Request example

```
{
  "channel":2,
  "description":"test",
  "ip":"10.5.4.178",
  "lib_ids":"18",
  "port":80,
  "position":"test",
  "camera_pwd":"admin123",
```

```
  "server_sip_id":"34020000002000000001",
  "camera_sip_id":"34020000001320000001",
  "protocol":"gb28181",
  "threshold":1,  "msg_id":"515",
  "camera_name":"test",
  "snap_mode":1,
  "facesize_min":60,
  "facesize_max":1000,
  "task_type": "face,body,vehicle",
  "frame_interval": 3,
  "liveness_threshold": 0,
  "liveness_switch": 0
}
```

Response example

```
{
  "code": 0,
  "data":
    {  "channel": 2
  },
  "msg": ""
}
```

e. Request parameter list for adding a camera in image mode:

Request parameter

Parameter	Type	Description	Required	Example
msg_id	string	“515”	Yes	
ip	string	Camera IP address after modification	No	
port	int	Camera port number after modification	No	
channel	int	Channel; value range:[1,16] of SenseNebula-M4s, [1,32] of SenseNebula-M8s	Yes	
position	string	Camera position after modification	No	
camera_us	strin	Camera login	No	

er	g	username after modification		
camera_pwd	string	Camera login password after modification	No	
lib_ids	string	IDs of camera alert deployment databases after modification	No	
description	string	Camera description after modification	No	
threshold	int	Similarity threshold after modification; value range: [1,100]	No	
vendor	string	Camera manufacturers	No	Support hikvision,sensedlc11,sensedlct,sensedlcd,dahuas,dahua; No or wrong filling is default to dahua.
camera_name	string	Camera name after modification	No	
task_type	string	Task type. Default to face,body,vehicle,cyclist when the value is invalid	No	face,body,vehicle means simultaneously verifying face, body and vehicle

Response parameter

Parameter	Type	Description
code	int	Result code (0:successful)
data	json	Returned data
msg	string	Result description

Field information(data)

Parameter	Type	Description
channel	int	Channel; value range:[1,16] of SenseNebula-M4s, [1,32] of SenseNebula-M8s

Example

Request example

```
{
  "channel":1,
```

```
   "description":"test",
   "ip":"10.5.4.177",
   "lib_ids":"1",
   "port":80,
   "position":"test",
   "camera_pwd":"admin123",
   "camera_user":"admin",
   "threshold":1,
   "vendor": "dahua",
   "msg_id":"515",
   "camera_name":"test",
   "task_type": "face,body,vehicle"
}
```

Response example

```
{
   "code": 0,
   "data":
      {  "channel": 1
   },
   "msg": ""
}
```

Note 1: Currently, cameras of video streams only support RTSP, ONVIF and GB28181. The decoder and protocol parameters are temporarily reserved.
Note 2: The lib_ids parameter is in the format "lib1,lib2,lib3,...". Separate multiple IDs with commas (,). Note 3: The channel parameter value ranges from 1 to 8 for SenseNebula-M4s and from 1 to 16 for SenseNebula-M8s.
Note 4: When the entered camera IP address is incorrect or invalid, wait until the server responds. Note 5: If it is not required, the original information will be retained.
Note 6: The vendor field can be filled with hikvision, sensedlc11, sensedlct, sensedlcd, dahuas, dahua, which respectively represent the brand of capture cameras of Hikvision, SenseDLC 11 , SenseDLC T series, SenseDLC D series , and Dahua/SenseDLC AA series.

4.4 Deleting a camera Interface Description

Interface description

Interface url	http://${ip}:${port}/api/json
Request Method	POST
Request Parameter Format	JSON
Interface Description	Deletes a camera

Request parameter

Parameter	Type	Description	Required	Example
msg_id	string	“514”	Yes	
channel	int	Channel; value range:[1,16] of SenseNebula-M4s, [1,32] of SenseNebula-M8s	Yes	

Response parameter

Parameter	Type	Description
code	int	Result code (0:successful)
data	json	NULL
msg	string	Result description

Example

Request example

```
{
  "msg_id": "514",
  "channel": 16
}
```

Response example

```
{
  "code":0,
  "data":"",
  "msg":""
}
```

4.5 Querying SIP server ID

Interface description

Interface url	http://${ip}:${port}/api/json
Request Method	POST
Request Parameter Format	JSON
Interface Description	Queries SIP server id

Request parameter

Parameter	Type	Description	Required	Example
msg_id	string	“1310”	Yes	

Response parameter

Parameter	Type	Description
code	int	Result code (0:successful)
data	json	NULL
msg	string	Result description

Field information(data)

Parameter	Type	Description
server_sip_id	string	SIP server id

Example

Request example

```
{
  "msg_id":"1310"
}
```

Response example

```
{
  "code": 0,
  "data": {
    "server_sip_id": "34020000002000000001"
  },
  "msg": ""
}
```

5 Face function-related interfaces

5.1 Detecting a single face image

Interface description

Interface url	http://${ip}:${port}/api/form
Request Method	POST
Request Parameter Format	FORM
Interface Description	Detects a single face image

Request parameter

Parameter	Type	Description	Required	Example
msg_id	string	“769”	Yes	
img	file	Face image	Yes	

Response parameter

Parameter	Type	Description
code	int	Result code (0:successful)
data	json	Returned data
msg	string	Result description

Field information(data)

Parameter	Type	Description
quality	int	Quality score, value range:[0,100]
coordinate	string	Coordinate of face in the face image. (x1,y1,x2,y2 indicate the coordinates of the left, top, right, and bottom boundaries of the rectangle, respectively.)
attribute	json	Face attributes, for details see in appendix B

Example

Request example

```
{
  "msg_id":"769", "img":test.jpg
}
```

Response example

```
{
  "code": 0,
  "data": {
    "attribute": {
      "cap_style": "hat_style_type_none", "gender_code":
      "male", "glass_style":"glasses_style_type_none",
      "mustache_style": "mustache_style_type_none",
      "st_respirator": "st_respirator_mouth",
      "respirator_color": "color_type_none", "st_age":
      "st_adult",
      "st_age_value": "24.000000",
      "st_expression": "st_calm"
    },
    "quality": 67,
    "coordinate": "56,10,432,678"
```

```
  },
  "msg": ""
}
```

5.2 Detecting multi face image

Interface description

Interface url	http://${ip}:${port}/api/form
Request Method	POST
Request Parameter Format	FORM
Interface Description	Detects multi face image

Request parameter

Parameter	Type	Description	Required	Example
msg_id	string	“782”	Yes	
img	file	Face image	Yes	

Response parameter

Parameter	Type	Description
code	int	Result code (0:successful)
data	json list	Returned data
msg	string	Result description

Field information(data)

Parameter	Type	Description
quality	int	Quality score, value range:[0,100]
coordinate	string	Coordinate of face in the face image. (x1,y1,x2,y2 indicate the coordinates of the left, top, right, and bottom boundaries of the rectangle, respectively.)
img_path	string	Path of the face image
attribute	json	Face attributes, for details see in appendix B

Example

Request example

```
{
  "msg_id":"782",  "img":test.jpg
```

```
}
```

Response example


```
{
  "code": 0,
  "data": [
    {
      "attribute": {
        "cap_style": "hat_style_type_none",
        "gender_code": "male",
        "glass_style": "glasses_style_type_none",
        "mustache_style": "mustache_style_type_none",
        "st_respirator": "st_respirator_mouth",
        "respirator_color": "color_type_none",   "st_age":
        "st_adult",
        "st_age_value": "24.000000",
        "st_expression": "st_calm"
      },
      "quality": 67,
      "img_path": "tmp/943f95bc-c7df-408f-bae8-c3285557f3ef.jpg",
      "coordinate": "56,10,432,678"
    }
  ],
  "msg": ""
}
```


5.3 Detecting a single face image(base64 encoding)

Interface description

Interface url	http://${ip}:${port}/api/json
Request Method	POST
Request Parameter Format	JSON
Interface Description	Detects a single face image

Request parameter

Parameter	Type	Description	Required	Example
msg_id	string	“779”	Yes	
img	json	Face image	Yes	

Field information(img)

Parameter	Type	Description
filename	string	file name
data	string	base64 encoding data

Response parameter

Parameter	Type	Description
code	int	Result code (0:successful)
data	json	Returned data
msg	string	Result description

Field information(data)

Parameter	Type	Description
quality	int	Quality score, value range:[0,100]
coordinate	string	Coordinate of face in the face image. (x1,y1,x2,y2 indicate the coordinates of the left, top, right, and bottom boundaries of the rectangle, respectively.)
attribute	json	Face attributes, for details see in appendix B

Example

Request example


```
{
  "msg_id":"779",
  "img": {
    "filename":"haha.jpg", "data":""
  }
}
```


Response example


```
{
  "code": 0,
  "data": {
    "attribute": {
      "cap_style": "hat_style_type_none", "gender_code":
      "male", "glass_style":"glasses_style_type_none",
      "mustache_style": "mustache_style_type_none",
      "st_respirator": "st_respirator_mouth",
      "respirator_color": "color_type_none",  "st_age":
      "st_adult",
      "st_age_value": "24.000000",
```

```
        "st_expression": "st_calm"
      },
      "quality": 67,
      "coordinate": "56,10,432,678"
    },
    "msg": ""
}
```


5.4 1:N face image compare

Interface description

Interface url	http://${ip}:${port}/api/form
Request Method	POST
Request Parameter Format	FORM
Interface Description	You can enter a single image and compare it with the data in the specified face database. The returned results include the top 1 to 50 matched records, face coordinates, face quality score, and attributes (name, age, gender, and ID) of N persons.

Request parameter

Parameter	Type	Description	Required	Example
msg_id	string	“772”	Yes	
threshold	int	Comparison threshold; value range:[0,100]; default value:0	No	
img	file	Format of the face image to be compared, which is xxx.jpg	Yes	
lib_ids	string	Target face database for comparison You can enter up to 50 database IDs and separate them with commas (,).	Yes	
topk	int	Number of top N records to be returned; a	No	

		maximum of top 50 records can be returned; value range:[1,50]; default value: 1		
n_topk	int	The nth of the top N returned records; value range:[1,top k]; default value: 1	No	

Response parameter

Parameter	Type	Description
code	int	Result code (0:successful)
data	json	Returned data
msg	string	Result description

Field information(data)

Parameter	Type	Description
n_topn	json	The nth of the top N returned records, which is determined by the n_topk parameter
topn	json list	The first top N returned records
In_pic	json	Input face in Original attribute

Field information(n_topn)

Parameter	Type	Description
img_id	string	ID of the face in the source database that maps the ranking
img_path	string	Path of the face image in the source database that maps the ranking
lib_id	int	ID of the database that maps the ranking
lib_name	string	Name of the database that maps the ranking
lib_type	int	Type of the face database maps the ranking
ranking	int	Ranking
similarity	int	Similarity coefficient of the face image that maps the ranking; value range:[0,100]

id	string	ID of the face in the source database that maps the ranking
name	string	Name of the face in the source database that maps the ranking
gender	string	Gender of the face in the source database that maps the ranking (0:female, 1:male)
age	string	Age of the face in the source database that maps the ranking
address	string	Address of the face in the source database that maps the ranking
wanderdeviceid	string	The stranger corresponding to the rank captures the channel number
wandertrigger	string	The capture time of strangers corresponding to the ranking

Field information(topn)

Parameter	Type	Description
img_id	string	ID of the face in the source database that maps the ranking
img_path	string	Path of the face image in the source database that maps the ranking
lib_id	int	ID of the database that maps the ranking
lib_name	string	Name of the database that maps the ranking
lib_type	int	Type of the face database maps the ranking
ranking	int	Ranking
similarity	int	Similarity coefficient of the face image that maps the ranking; value range:[0,100]
id	string	ID of the face in the source database that maps the ranking
name	string	Name of the face in the source database that maps the ranking
gender	string	Gender of the face in the source database that maps the ranking (0:female, 1:male)
age	string	Age of the face in the source

		database that maps the ranking
address	string	Address of the face in the source database that maps the ranking
wanderdeviceid	string	The stranger corresponding to the rank captures the channel number
wandertrigger	string	The capture time of strangers corresponding to the ranking

Field information(In_pic)

Parameter	Type	Description
pos_top	int	Top coordinate of the input face in the source image
pos_bottom	int	Bottom coordinate of the input face in the source image
pos_left	int	Left coordinate of the input face in the source image
pos_right	int	Right coordinate of the input face in the source image
quality	int	Quality score, value range:[0,100]

Example

Request example

```
Input : files
{
  'img':  name='1542712163434643.jpg'
}
  "lib_ids":"1",
  "threshold":0,
  "topk":3,
  "msg_id":"772",
  "n_topk":1
```

Response example

```
{
  "code": 0,
  "data": {
    "in_pic":
      {  "pos_top":255,
      "pos_bottom":400,
      "pos_left":100,
      "pos_right":300,
```

```
    "quality":95,
  },
  "n_topn": {
    "img_id": "deab1032-b90a-4d02-9f9f-a1a25907b467",   "img_path":
    "img/1_deab1032-b90a-4d02-9f9f-a1a25907b467.jpg",  "lib_id": 1,
    "lib_name": "blacklib",
    "lib_type": 1,
    "ranking": 1,  "similarity":
    36,
    "id":4259541481123111,
    "name":zhangsan,
    "gender":0,
    "age":18,
    "address":address,
    "wanderdeviceid": ""
    "wandertrigger": ""
  },
  "topn": [
    {
      "img_id":   "deab1032-b90a-4d02-9f9f-a1a25907b467",   "img_path":
       "img/1_deab1032-b90a-4d02-9f9f-a1a25907b467.jpg",  "lib_id": 1,
       "lib_name": "blacklib",
       "lib_type": 1,
       "ranking": 1,  "similarity":
       36,
       "id":4259541481123111,
       "name":zhangsan,
       "gender":0,
       "age":18,
       "address":address1
       "wanderdeviceid": ""
       "wandertrigger": ""
    },
    {
       "img_id": "a068b878-95cb-4aa7-aba6-af95cd5b36b3",   "img_path":
       "img/1_a068b878-95cb-4aa7-aba6-af95cd5b36b3.jpg",  "lib_id": 1,
       "lib_name": "blacklib",
       "lib_type": 1,
       "ranking": 2,
       "similarity": 33,
```

```
            "id":4259541481123111,
            "name":lisi,
            "gender":0,
            "age":18,
            "address":address2
            "wanderdeviceid": ""
            "wandertrigger": ""
        },
        {
            "address": "******"
            "age": ""
            "gender": ""
            "id": "******"
            "img_id": "v214_1626681187_1214_0"
            "img_path": ""
            "lib_id": 2147483647
            "lib_name": ""
            "lib_type": 0  "name":
            ""  "ranking": 1
            "similarity": 40
            "wanderdeviceid": "111"
            "wandertrigger": "2021-07-19 15:53:07"
        }
    ]
  },
  "msg": ""
}
```


5.5 1:1 Face image compare

Interface description

Interface url	http://${ip}:${port}/api/form
Request Method	POST
Request Parameter Format	FORM
Interface Description	Compares a single face with a single face image.

Request parameter

Parameter	Type	Description	Required	Example
msg_id	string	“771”	Yes	
img_1	file	The first input face	Yes	

img_2	file	The second input face	Yes	

Response parameter

Parameter	Type	Description
code	int	Result code (0:successful)
data	json	Returned data
msg	string	Result description

Field information(data)

Parameter	Type	Description
FaceImg1to1Rsp_Score	int	Similarity between the two faces

Example

Request example

```
Input : files
{
  "msg_id"：  "771"
  'img_1': name='1542712163434643.jpg'
  'img_2': name='54646644544635.jpg'
}
```

Response example

```
{
  "code": 0,
  "data":
    { "FaceImg1to1Rsp_Score":
    30
  },
  "msg": ""
}
```

5.6 1:N face image compare(base64 encoding)

Interface description

Interface url	http://${ip}:${port}/api/json
Request Method	POST
Request Parameter Format	JSON
Interface Description	You can enter a single image and compare it with the data in the specified face database. The returned results include the top 1 to 50 matched records, face coordinates, face quality score, and

	attributes (name, age, gender, and ID) of N persons.

Request parameter

Parameter	Type	Description	Required	Example
msg_id	string	"780"	Yes	
threshold	int	Comparison threshold; value range:[0,100]; default value:0	No	
img	json	Face image to be compared	Yes	
lib_ids	string	Target face database for comparison You can enter up to 50 database IDs and separate them with commas (,).	Yes	
topk	int	Number of top N records to be returned; a maximum of top 50 records can be returned; value range:[1,50]; default value: 1	No	
n_topk	int	The nth of the top N returned records; value range:[1,top k]; default value: 1	No	

Field information(img)

Parameter	Type	Description
filename	string	file name
data	string	base64 encoding data

Response parameter

Parameter	Type	Description
code	int	Result code (0:successful)
data	json	Returned data

msg	string	Result description

Field information(data)

Parameter	Type	Description
n_topn	json	The nth of the top N returned records, which is determined by the n_topk parameter
topn	json list	The first top N returned records
In_pic	json	Input face in Original attribute

Field information(n_topn)

Parameter	Type	Description
img_id	string	ID of the face in the source database that maps the ranking
img_path	string	Path of the face image in the source database that maps the ranking
lib_id	int	ID of the database that maps the ranking
lib_name	string	Name of the database that maps the ranking
lib_type	int	Type of the face database maps the ranking
ranking	int	Ranking
similarity	int	Similarity coefficient of the face image that maps the ranking; value range:[0,100]
id	string	ID of the face in the source database that maps the ranking
name	string	Name of the face in the source database that maps the ranking
gender	string	Gender of the face in the source database that maps the ranking (0:female, 1:male)
age	string	Age of the face in the source database that maps the ranking
address	string	Address of the face in the source database that maps the ranking

Field information(topn)

Parameter	Type	Description
img_id	string	ID of the face in the source

		database that maps the ranking
img_path	string	Path of the face image in the source database that maps the ranking
lib_id	int	ID of the database that maps the ranking
lib_name	string	Name of the database that maps the ranking
lib_type	int	Type of the face database maps the ranking
ranking	int	Ranking
similarity	int	Similarity coefficient of the face image that maps the ranking; value range:[0,100]
id	string	ID of the face in the source database that maps the ranking
name	string	Name of the face in the source database that maps the ranking
gender	string	Gender of the face in the source database that maps the ranking (0:female, 1:male)
age	string	Age of the face in the source database that maps the ranking
address	string	Address of the face in the source database that maps the ranking

Field information(In_pic)

Parameter	Type	Description
pos_top	int	Top coordinate of the input face in the source image
pos_bottom	int	Bottom coordinate of the input face in the source image
pos_left	int	Left coordinate of the input face in the source image
pos_right	int	Right coordinate of the input face in the source image
quality	int	Quality score, value range:[0,100]

Example

Request example

Input : files

```
{
   "msg_id":"780",
   "lib_ids":"1",
   "threshold":0,
   "topk":3,
   "n_topk":1,
   "img": {
            "filename":"haha.jpg",
            "data":""
   }
}
```

Response example

```
{
   "code": 0,
   "data": {
      "in_pic":
         {  "pos_top":255,
         "pos_bottom":400,
         "pos_left":100,
         "pos_right":300,
         "quality":95,
      },
      "n_topn": {
         "img_id":  "deab1032-b90a-4d02-9f9f-a1a25907b467",   "img_path":
         "img/1_deab1032-b90a-4d02-9f9f-a1a25907b467.jpg",  "lib_id": 1,
         "lib_name": "blacklib",
         "lib_type": 1,
         "ranking": 1,
         "similarity": 36,
         "id":4259541481123111,
         "name":zhang,
         "gender":0,
         "age":18,
         "address":add,
      },
      "topn": [
         {
           "img_id":  "deab1032-b90a-4d02-9f9f-a1a25907b467",   "img_path":
            "img/1_deab1032-b90a-4d02-9f9f-a1a25907b467.jpg",  "lib_id": 1,
            "lib_name": "blacklib",
```

```
        "lib_type": 1,
        "ranking": 1,  "similarity":
        36,
        "id":4259541481123111,
        "name":zhang,
        "gender":0,
        "age":18,
        "address":add1
    },
    {
        "img_id":  "a068b878-95cb-4aa7-aba6-af95cd5b36b3",   "img_path":
        "img/1_a068b878-95cb-4aa7-aba6-af95cd5b36b3.jpg",  "lib_id": 1,
        "lib_name": "blacklib",
        "lib_type": 1,
        "ranking": 2,  "similarity":
        33,
        "id":4259541481123111,
        "name":si,
        "gender":0,
        "age":18,
        "address":add2
    },
    {
        "img_id":  "fb7dcd8b-bfee-4e47-b7cd-5cc4e6d09d60",   "img_path":
        "img/1_fb7dcd8b-bfee-4e47-b7cd-5cc4e6d09d60.jpg",  "lib_id": 1,
        "lib_name": "blacklib",
        "lib_type": 1,
        "ranking": 3,  "similarity":
        31,  "pos_top":255,
        "pos_bottom":400,
        "pos_left":100,
        "pos_right":300,
        "quality":95,
        "id":4259541481123111,
        "name":wu,
        "gender":0,
        "age":18,
        "address":add3
    }
]
```

```
  },
  "msg": ""
}
```

5.7 1:1Face image compare(base64 encoding)

Interface description

Interface url	http://${ip}:${port}/api/json
Request Method	POST
Request Parameter Format	JSON
Interface Description	Compares a single face with a single face image.

Request parameter

Parameter	Type	Description	Required	Example
msg_id	string	“781”	Yes	
img_1	json	The first input face img	Yes	
img_2	json	The second input face img	Yes	

Field information(img)

Parameter	Type	Description
filename	string	file name
data	string	base64 encoding data

Response parameter

Parameter	Type	Description
code	int	Result code (0:successful)
data	json	Returned data
msg	string	Result description

Field information(data)

Parameter	Type	Description
FaceImg1to1Rsp_Score	int	Similarity between the two faces

Example

Request example

```
Input : files
{
  "msg_id"：  "781",
```

```
  "img_1": {
            "filename":"haha1.jpg",
            "data":""
  },
  "img_2": {
            "filename":"haha2.jpg",
            "data":""
  }
}
```

Response example

```
{
  "code": 0,
  "data":
    { "FaceImg1to1Rsp_Score":
    30
  },
  "msg": ""
}
```

6 Stranger clustering-related interfaces

6.1 Modifying stranger clustering related settings

Interface description

Interface url	http://${ip}:${port}/api/json
Request Method	POST
Request Parameter Format	JSON
Interface Description	Modifies the stranger clustering related settings

Request parameter

Parameter	Type	Description	Required	Example
msg_id	string	“1333”	Yes	
interval	int	Deduplicate period (Unit： minites), deduplication period options are 0, 1, 3, 5, 10, 30, 60 minutes, and the default value is 1 minute	Yes	10

threshold	int	Recognition threshold, from [65,100] and the default value is 80	Yes	70
enable	int	Enable switch , 1:enabled, 0:disabled	Yes	1

Response parameter

Parameter	Type	Description
code	int	Result code (0:successful)
data	json	Returned data
msg	string	Result description

Example

Request example

```
{
   "msg_id":"1333",
   "interval":10,
   "threshold":80,
   "enable":1
}
```

Response example

```
{
   "code": 0,
   "data": "",
   "msg": ""
}
```

6.2 Querying stranger clustering settings

Interface description

Interface url	http://${ip}:${port}/api/json
Request Method	POST
Request Parameter Format	JSON
Interface Description	Queries stranger clustering settings

Request parameter

Parameter	Type	Description	Required	Example
msg_id	string	“1334”	Yes	

Response parameter

Parameter	Type	Description
code	int	Result code (0:successful)
data	json	Returned data
msg	string	Result description

Field information(data)

Parameter	Type	Description
enable	int	Enable switch , 1:enabled, 0:disabled
interval	int	Deduplicate period (Unit：minites), deduplication period options are 0, 1, 3, 5, 10, 30, 60 minutes, and the default value is 1 minute
threshold	int	Recognition threshold, from [65,100] and the default value is 80

Example

Request example

```
{{
   "msg_id":"1334"
}}
```

Response example

```
{
   "code": 0,
   "data": {
      "enable":1,
      "interval":10,
      "threshold":80
      },
   "msg": ""
}
```

6.3 Setting the wandering alarm threshold

Interface description

Interface url	http://${ip}:${port}/api/json
Request Method	POST
Request Parameter Format	JSON
Interface Description	Sets the wandering alarm threshold

Request parameter

Parameter	Type	Description	Required	Example
msg_id	string	“1335”	Yes	
channels	string	The set wandering places for the wandering alarm. Only valid channels can be set.	Yes	
appeared_times	int	Number of wanderings, the threshold of occurrences number of a stranger at the selected wandering places. It should be larger than 0.	Yes	
enable	int	Enable switch , 1:enabled, 0:disabled	Yes	
interval	int	The set wandering time, the time period when strangers wander. The options are 3, 5, 7, 15 which means Nearly 3 days, Nearly 5 days, Nearly 7 days and Nearly 15 days, respectively. The default value is 3.	Yes	

Response parameter

Parameter	Type	Description
code	int	Result code (0:successful)
data	string	Verification code string
msg	string	Result description

Example

Request example

```
{
   "msg_id":"1335",
   "channels":"1,2,3,4,5",
   "appeared_times":1,
   "enable":0,  "interval":3
}
```

Response example

```
{
   "code": 0,
   "data": "",
   "msg": ""
}
```

6.4 Querying wandering alarm threshold

Interface description

Interface url	http://${ip}:${port}/api/json
Request Method	POST
Request Parameter Format	JSON
Interface Description	Queries wandering alarm threshold

Request parameter

Parameter	Type	Description	Required	Example
msg_id	string	“1336”	Yes	

Response parameter

Parameter	Type	Description
code	int	Result code (0:successful)
data	string	Verification code string
msg	string	Result description

Field information(data)

Parameter	Type	Description
enable	int	Enable switch , 1:enabled, 0:disabled
interval	int	The set wandering time, the

		time period when strangers wander. The options are 3, 5, 7, 15 which means Nearly 3 days, Nearly 5 days, Nearly 7 days and Nearly 15 days, respectively. The default value is 3.
channels	string	The set wandering places for the wandering alarm
appeared_times	int	Number of wanderings, the threshold of occurrences number of a stranger at the selected wandering places. It should be larger than 0.

Example

Request example

```
{
  "msg_id":"1336"
}
```

Response example

```
{
  "code": 0,
  "data":
    { "channels":"1,2,3,4,5"
    , "appeared_times":1,
    "enable":0, "interval":1
    },
  "msg": ""
}
```

7 Personnel profile-related interfaces

7.1 Paging query for details of Personnel profil

Interface description

Interface url	http://${ip}:${port}/api/json
Request Method	POST
Request Parameter Format	JSON
Interface Description	According to the personnel's ID in the stranger database, query the archive details by page

Request parameter

Parameter	Type	Description	Required	Example
msg_id	string	“1065”	Yes	
person_id	string	personnel id in the person database	Yes	
start_no	int	Start position (Default value:1)	No	1
qry_len	int	Number of query items [1,50] (Default value:10)	No	10
channels	string	A comma-separated list of channel numbers, empty means to query all channels	No	1,3,6
start_time	string	Start time (Time format：YYYY-MM-DD HH:MM:SS）	No	2019-09-02 10:10:10
stop_time	string	End time (Time format：YYYY-MM-DD HH:MM:SS）	No,When start_time is filled in, stop_time must also be filled in	2019-09-02 20:20:20

Response parameter

Parameter	Type	Description
code	int	Result code (0:successful)
data	json	Returned data
msg	string	Result description

Field information(data)

Parameter	Type	Description
record	json list	Returned data
result_num	int	Number of query results
total_num	int	Total number of records

Field information(record)

Parameter	Type	Description
camera_name	string	Camera name

channel	int	Channel; value range:[1,16] of SenseNebula-M4s, [1,32] of SenseNebula-M8s
face_attr	json	Face attributes, refer to Appendix B
img_id	string	Stranger id in the stranger database
lib_id	int	Stranger database id
lib_name	string	Null
lib_type	int	Null
alive_type	int	Alive type，0：Unknown；1：Non alive；2：Alive
appear_count	int	Current occurrences
img_path	string	personnel database image path
alarm_type	int	Face alarm type (0: no alarm, 1: key personnel/allowlist alarm, 2: stranger alarm, 3: hit stranger db alarm)
event_type	int	Event alarm type (0: no alarm, 1: wandering alarm, 2: staying alarm)
stranger_appear_channel	int	The capture channel of the stranger database image
wander_channels	string	The set wandering places for the wandering alarm
wander_deviceID	int	Camera name of the triggered wandering alarm
wander_trigger	int	The trigger time for wandering alarm
position	string	Camera position
ranking	int	1 indicates the highest ranking in the comparison results
similarity	int	Similarity coefficient of the face image that maps the ranking; value range:[0,100]
snap_id	string	Capture ID
snap_path	string	Capture path
threshold	int	Threshold; value range:[0,100]
trigger	string	Capture time

Example

Request example


```
{
  "msg_id": "1065",
  "person_id": "78e0f17b-4fc4-4b22-993b-8669b9a44ce5",
  "qry_len": 50,
  "start_no": 1
}
```


Response example


```
{
  "code": 0,
  "data": {
    "record": [
      {
        "camera_name": "121",
        "channel": 4,
        "face_attr": {"cap_style": "hat_style_type_none", "gender_code": "male", "glasses_style":
"glasses_style_type_none", "mustache_style": "mustache_style_none", "respirator_color":  "color_type_none",
"st_age": "st_adult", "st_age_value": "30.000000", "st_expression": "st_calm"},
        "img_id": "78e0f17b-4fc4-4b22-993b-8669b9a44ce5",
        "lib_id": 2147483647,
        "lib_name": "",
        "lib_type": 0,
        "position": "",
        "ranking": 1,
        "similarity": 0,
        "snap_id":  "78e0f17b-4fc4-4b22-993b-8669b9a44ce5",
        "snap_path": "record/channel4/face/78e0f17b-4fc4-4b22-993b-8669b9a44ce5.jpg",
        "threshold": 85,
        "trigger": "2020-03-31 13:52:49"
      }
    ],
    "result_num": 1
    "total_num": 1
  },
  "msg": ""
}
```

7.2 Exporting the Personnel profile details in pagination mode

Interface description

Interface url	http://${ip}:${port}/api/json
Request Method	POST
Request Parameter Format	JSON
Interface Description	Exports the Personnel profile details in pagination mode

Request parameter

Parameter	Type	Description	Required	Example
msg_id	string	“2051”	Yes	
person_id	string	personnel id in the personnel database	Yes	
start_no	int	Start position (Default value:1)	No	1
qry_len	int	Number of query items [1,50] (Default value:10)	No	10
channels	string	A comma-separated list of channel numbers, empty means to query all channels	No	1,3,6
start_time	string	Start time (Time format：YYYY-MM-DD HH:MM:SS）	No	2019-09-02 10:10:10
stop_time	string	End time (Time format：YYYY-MM-DD HH:MM:SS）	No,When start_time is filled in, stop_time must also be filled in	2019-09-02 20:20:20

Response parameter

Parameter	Type	Description
code	int	Result code (0:successful)
data	json	Returned data
msg	string	Result description

Field information(data)

Parameter	Type	Description
export_url	string	Exported file path
result_num	int	Number of query results
total_num	int	Total number of records
zippasswd	string	Compressed password

Example

Request example

```
{
  "msg_id": "2051",
  "person_id": "78e0f17b-4fc4-4b22-993b-8669b9a44ce5",
  "start_no": 1,
  "qry_len": 50,
  "channels": "1,3,6",
  "start_time":"2019-10-16 13:20:00",
  "stop_time":"2019-10-16 18:00:00"
}
```

Response example

```
{
  "code": 0,
  "data": {
    "export_url": "tmp/greylist_detail_2019-01-23-12-09-42.zip",
    "result_num": 10,
    "total_num": 108,
                    "zippasswd":"xxxx"
  },
  "msg": ""
}
```

7.3 Exporting selected Personnel profile details

Interface description

Interface url	http://${ip}:${port}/api/json
Request Method	POST
Request Parameter Format	JSON
Interface Description	Exports selected Personnel profile details

Request parameter

Parameter	Type	Description	Required	Example
msg_id	string	“2052”	Yes	

person_id	string	personnel id in the personnel database	Yes	
snap_ids	string	A set of capture images IDs, which can be obtained through query (Separate image IDs with commas [,]. The image quantity ranges from 1 to 50.)	Yes	

Response parameter

Parameter	Type	Description
code	int	Result code (0:successful)
data	json	Returned data
msg	string	Result description

Field information(data)

Parameter	Type	Description
export_url	string	Exported file path
failed_img_ids	string	ID of the image that cannot be exported
result_num	int	Number of query results
total_num	int	Total number of exported images
zippasswd	string	Compressed password

Example

Request example

```
{
  "msg_id":"2052",
  "snap_ids":"74a199aa-dad7-4d3a-95f0-f74dbb818974,f12ed623-87d6-49e0-939a-97262a0407e9",
  "preson_id":"74a199aa-dad7-4d3a-95f0-f74dbb818974"
}
```

Response example

```
{
  "code": 0,
  "data": {
    "export_url": "tmp/catch_2019-01-23-12-44-37.zip",
    "failed_img_ids": "",
```

```
    "result_num": 5,
    "total_num": 5,
                    "zippasswd":"xxxx"
  },
  "msg": ""
}
```


7.4 Querying the Personnel database image based on the capture record

Interface description

Interface url	http://${ip}:${port}/api/json
Request Method	POST
Request Parameter Format	JSON
Interface Description	Queries the Personnel database image based on the capture record

Request parameter

Parameter	Type	Description	Required	Example
msg_id	string	“1066”	Yes	
snap_id	string	Capture ID	Yes	

Response parameter

Parameter	Type	Description
code	int	Result code (0:successful)
data	json	Returned data
msg	string	Result description

Field information(data)

Parameter	Type	Description
record	json list	Returned data
result_num	int	Number of queried results，Here the value is 1
total_num	int	Total number of records，Here the value is 1

Field information(record)

Parameter	Type	Description
camera_name	string	Camera name
channel	int	Channel; value range:[1,16] of

		SenseNebula-M4s, [1,32] of SenseNebula-M8s
alarm_type	int	Face alarm type (0: no alarm, 1: key personnel/allowlist alarm, 2: stranger alarm, 3: hit stranger db alarm)
event_type	int	Event alarm type (0: no alarm, 1: wandering alarm, 2: staying alarm)
img_id	string	Personnel id in the Personnel database
face_attr	json	Face capture attributes, for details, see the appendix B
lib_id	int	Personnel database id
lib_name	string	Null
lib_type	int	Null
position	string	Camera position
ranking	int	1 indicates the highest ranking in the comparison results
similarity	int	Similarity score; value range:[0,100]
snap_id	string	Capture id
snap_path	string	Storage path of the capture image
threshold	int	Threshold; value range:[0,100]
trigger	string	Capture time

Example

Request example


```
{
  "msg_id": "1066",
  "snap_id":  "78e0f17b-4fc4-4b22-993b-8669b9a44ce5"
}
```


Response example


```
{
  "code": 0,
  "data": {
    "record": [
      {
        "camera_name": "121",
        "channel": 4,
        "face_attr": {"cap_style": "hat_style_type_none", "gender_code": "male", "glasses_style":
```

"glasses_style_type_none", "mustache_style": "mustache_style_none", "respirator_color": "color_type_none", "st_age": "st_adult", "st_age_value": "30.000000", "st_expression": "st_calm"},
"lib_id": 2147483647,
"lib_name": "",
"lib_type": 0,
"position": "",
"ranking": 1,
"similarity": 0,
"snap_id": "78e0f17b-4fc4-4b22-993b-8669b9a44ce5",
"snap_path": "record/channel4/face/78e0f17b-4fc4-4b22-993b-8669b9a44ce5.jpg",
"threshold": 85,
"trigger": "2020-04-20 08:51:52"
}
],
"result_num": 1
"total_num": 1
},
"msg": ""
}

8 System configuration interfaces

8.1 Querying NTP configuration

Interface description

Interface url	http://${ip}:${port}/api/json
Request Method	POST
Request Parameter Format	JSON
Interface Description	Queries NTP time synchronization configuration.

Request parameter

Parameter	Type	Description	Required	Example
msg_id	string	“1291”	Yes	

Response parameter

Parameter	Type	Description
code	int	Result code (0:successful)
data	json	Returned data
msg	string	Result description

Field information(data)

Parameter	Type	Description
ntp_cycle	int	Synchronization cycle, the rage in [1min,30mins]; default value: 10 minutes
ntp_disable	string	Whether to disable NTP; valid values: 0: Not disable NTP; 1: Disable NTP
port	int	Port number of the NTP server
url	string	URL of the NTP server

Example

Request example

```
{
  "msg_id":"1291"
}
```

Response example

```
{
  "code": 0,
  "data": { "ntp_cycle":
    10, "ntp_disable":
    "0", "port": 123,
    "url": "edu.ntp.org.cn"
  },
  "msg": ""
}
```

8.2 Modifying NTP configuration

Interface description

Interface url	http://${ip}:${port}/api/json
Request Method	POST
Request Parameter Format	JSON
Interface Description	Modifies NTP time synchronization configuration.

Request parameter

Parameter	Type	Description	Required	Example
msg_id	string	“1290”	Yes	
url	string	URL of the NTP server; default	No	

		value: pool.ntp.org		
ntp_cycle	string	Synchronization cycle, the rage in [1min,30mins]; default value:10 minutes	No	10
ntp_disable	string	Whether to disable NTP; valid values (0:not disable NTP, 1:disable NTP); default value:1	No	

Response parameter

Parameter	Type	Description
code	int	Result code (0:successful)
data	json	NULL
msg	string	Result description

Example

Request example

```
{
  "msg_id":"1290",
  "ntp_disable":"0"
}
```

Response example

```
{
  "code": 0,
  "data": "",
  "msg": ""
}
```

8.3 Querying version information

Interface description

Interface url	http://${ip}:${port}/api/json
Request Method	POST
Request Parameter Format	JSON
Interface Description	Queries version information

Request parameter

Parameter	Type	Description	Required	Example
msg_id	string	“1281”	Yes	

Response parameter

Parameter	Type	Description
code	int	Result code (0:successful)
data	json	Returned data
msg	string	Result description

Field information(data)

Parameter	Type	Description
device_id	string	Device ID
serial_id	string	SN
display_version	string	Server version for display
display_web_version	string	Web version for display
version	string	Server version
web_version	string	Web version

Example

Request example

```
{
  "msg_id": "1281"
}
```

Response example

```
{
  "code": 0,
  "data": {  "device_id":
    "666",
    "serial_id": "STSNMAWX08A118110001",
    "display_version": "SenseNebula AIE V2.2.0-20210825",
    "display_web_version": "SenseNebula AIE Web V2.2.0-20210825",
    "version": "SenseNebula-AIE-V1.0.2-20190122",
    "web_version":  "SenseNebulaWeb-AIE-V1.0.2-web-20190122"
  },
  "msg": ""
}
```

8.4 Querying the storage strategy

Interface description

Interface url	http://${ip}:${port}/api/json
Request Method	POST
Request Parameter Format	JSON
Interface Description	Queries the storage strategy

Request parameter

Parameter	Type	Description	Required	Example
msg_id	string	“1288”	Yes	

Response parameter

Parameter	Type	Description
code	int	Result code (0:successful)
data	json	Returned data
msg	string	Result description

Field information(data)

Parameter	Type	Description
cur_storage	int	Occupied storage space
max_storage	int	Maximum storage
storage_strategy	int	Storage strategy; valid values (1:refill, 2:freeze)

Example

Request example

```
{
  "msg_id":"1288"
}
```

Response example

```
{
  "code": 0,
  "data": {  "cur_storage":
    0,  "max_storage":
    2048,
    "storage_strategy": 1
  },
  "msg": ""
}
```

8.5 Modifying the storage strategy

Interface description

Interface url	http://${ip}:${port}/api/json
Request Method	POST
Request Parameter Format	JSON
Interface Description	Modifies the storage strategy.

Request parameter

Parameter	Type	Description	Required	Example
msg_id	string	“1287”	Yes	
storage_strategy	int	Storage strategy; valid values (1:refill, 2:freeze)	Yes	1 or 2

Response parameter

Parameter	Type	Description
code	int	Result code (0:successful)
data	json	NULL
msg	string	Result description

Example

Request example

```
{
  "msg_id":"1287",
  "storage_strategy":1
}
```

Response example

```
{
  "code": 0,
  "data": "",
  "msg": ""
}
```

8.6 Querying the system time

Interface description

Interface url	http://${ip}:${port}/api/json
Request Method	POST
Request Parameter Format	JSON
Interface Description	Queries the system time.

Request parameter

Parameter	Type	Description	Required	Example
msg_id	string	“1283”	Yes	

Response parameter

Parameter	Type	Description
code	int	Result code (0:successful)
data	json	Returned data
msg	string	Result description

Field information(data)

Parameter	Type	Description
sys_time	string	System time
time_format	string	Time format
time_zone	string	Time zone

Example

Request example

```
{
   "msg_id": "1283"
}
```

Response example

```
{
   "code": 0,
   "data": {
      "sys_time": "2019-01-23 14:51:18",
      "time_format": "yyyy-MM-dd HH:mm:ss",
      "time_zone": "GMT+08:00"
   },
   "msg": ""
}
```

8.7 Setting the system time

Interface description

Interface url	http://${ip}:${port}/api/json
Request Method	POST
Request Parameter Format	JSON
Interface Description	Sets the system time.

Request parameter

Parameter	Type	Description	Required	Example
msg_id	string	“1282”	Yes	
sys_time	string	System time	Yes	2019-01-14 12:23:45
time_zone	string	Time zone	Yes	GMT+08:00
time_format	string	Time format, which may be yyyy-mm-dd, dd-mm-yyyy, or mm-dd-yyyy	No	yyyy-MM-dd HH:mm:ss

Response parameter

Parameter	Type	Description
code	int	Result code (0:successful)
data	json	NULL
msg	string	Result description

Example

Request example

```
{
  "msg_id": "1282",
  "sys_time":"2019-01-14 12:23:45",
  "time_zone":"GMT+08:00",
  "time_format":"yyyy-MM-dd HH:mm:ss"
}
```

Response example

```
{
  "code": 0,
  "data": "",
  "msg": ""
}
```

8.8 Querying network configuration

Interface description

Interface url	http://${ip}:${port}/api/json
Request Method	POST
Request Parameter Format	JSON
Interface Description	Queries network configuration

Request parameter

Parameter	Type	Description	Required	Example
msg_id	string	“1297”	Yes	

Response parameter

Parameter	Type	Description
code	int	Result code (0:successful)
data	json	Returned data
msg	string	Result description

Field information(data)

Parameter	Type	Description
default_info	json	Returned data
ethernet	json list	Network configuration

Field information(default_info)

Parameter	Type	Description
dns_method	string	DNS acquisition method
ethname	string	Network interface card (NIC) name
first_dns	string	Primary DNS
second_dns	string	Secondary DNS

Field information(ethernet)

Parameter	Type	Description
ethname	string	Network interface card (NIC) name
gateway	string	Gateway
hw_addr	string	MAC address
ip	string	IP
ip_method	string	IP address acquisition method, which may be static or DHCP
ip_version	string	IP version
netmask	string	Subnet mask

Example

Request example

```
{
  "msg_id": "1297"
}
```

Response example

```
{
  "code": 0,
  "data":
    {  "default_info":
    {
      "dns_method": "static",
      "ethname": "eth3",
      "first_dns": "10.5.4.207",
      "second_dns": "10.5.4.205"
    },
    "ethernet": [
      {
        "ethname": "eth0",
        "gateway": "10.5.2.1",
        "hw_addr": "00:04:4b:a5:4d:b5",
        "ip": "10.5.2.56",
        "ip_method": "static",
        "ip_version": "IPv4",
        "netmask": "255.0.0.0"
      },
      {
        "ethname": "eth1",
        "gateway": "",
        "hw_addr": "00:07:32:4e:68:44",
        "ip": "192.168.2.10",
        "ip_method": "static",
        "ip_version": "IPv4",
        "netmask": "255.255.255.0"
      },
      {
        "ethname": "eth2",
        "gateway": "",
        "hw_addr": "00:07:32:4e:68:45",
        "ip": "192.168.3.10",
        "ip_method": "static",
        "ip_version": "IPv4",
        "netmask": "255.255.255.0"
      },
      {
        "ethname": "eth3",
        "gateway": "",
        "hw_addr": "00:07:32:4e:68:46",
        "ip": "192.168.4.10",
        "ip_method": "static",
```

```
            "ip_version": "IPv4",
            "netmask": "255.255.255.0"
          }
        ]
    },
    "msg": ""
}
```

8.9 Configuring network settings

Interface description

Interface url	http://${ip}:${port}/api/json
Request Method	POST
Request Parameter Format	JSON
Interface Description	Configures network settings

Request parameter

Parameter	Type	Description	Required	Example
msg_id	string	“1296”	Yes	
ethname	string	Network interface card (NIC) name	Yes	
ip	string	IP address	Yes	
ip_method	string	IP acquisition method	Yes	Static or DHCP
netmask	string	Subnet mask	Yes	
gateway	string	Gateway	Yes	
ip_version	string	IP version	Yes	IPv4 or IPv6

Response parameter

Parameter	Type	Description
code	int	Result code (0:successful)
data	json	NULL
msg	string	Result description

Example

Request example

```
{
    "msg_id":"1296",
    "ethname":"eth3",
    "ip":"192.168.133.120",
    "ip_method":"static",
```

```
   "netmask":"255.255.255.0",
   "gateway":"192.168.133.1",
   "ip_version":"IPv4"
}
```

Response example

```
{
   "code": 0,
   "data": "",
   "msg": ""
}
```

8.10 Setting the default network port

Interface description

Interface url	http://${ip}:${port}/api/json
Request Method	POST
Request Parameter Format	JSON
Interface Description	Sets the default network port

Request parameter

Parameter	Type	Description	Required	Example
msg_id	string	“1295”	Yes	
dns_method	string	DNS acquisition method	Yes	Static or DHCP
first_dns	string	Primary DNS	No	
second_dns	string	Secondary DNS	No	
ethname	string	Network interface card (NIC) name	Yes	

Response parameter

Parameter	Type	Description
code	int	Result code (0:successful)
data	json	NULL
msg	string	Result description

Example

Request example

```
{
   "msg_id":"1295",
   "dns_method":"static",
   "first_dns":"10.5.4.207",
```

```
   "second_dns":"10.5.4.205",
   "ethname":"eth0"
}
```

Response example

```
{
   "code": 0,
   "data": "",
   "msg": ""
}
```

8.11 Querying the storage configuration

Interface description

Interface url	http://${ip}:${port}/api/json
Request Method	POST
Request Parameter Format	JSON
Interface Description	Query the storage configuration

Request parameter

Parameter	Type	Description	Required	Example
msg_id	string	“1900”	Yes	

Response parameter

Parameter	Type	Description
code	int	Result code (0:successful)
data	json	Returned data
msg	string	Result description

Field information(data)

Parameter	Type	Description
capture	string	whether to enter the database and save the picture (the 0th character means to enter the database, the first character means to save the picture. 0 is not saved, 1 is saved)
face	string	Whether the facebase save the image (0 is not saved, 1 is saved)

Example

Request example

```
{
   "msg_id":"1900"
}
```

Response example

```
{
   "code": 0,
   "data":{  "capture"
      : "11",
      "face": "1",
   }
   "msg": ""
}
```

8.12 Setting storage configuration

Interface description

Interface url	http://${ip}:${port}/api/json
Request Method	POST
Request Parameter Format	JSON
Interface Description	Set storage configuration

Request parameter

Parameter	Type	Description	Required	Example
msg_id	string	“1901”	Yes	
capture	string	whether to enter the database and save the picture (11 means enter the database and save the picture,00 means not save.)	Yes	11
face	string	whether to enter the database and save the picture (1 means enter the database and save the picture,0 means not save.)	Yes	1

Response parameter

Parameter	Type	Description
code	int	Result code (0:successful)
data	json	NULL
msg	string	Result description

Example

Request example

```
{
   "msg_id":"1901",
   "capture":"11",
   "face":"1"
}
```

Response example

```
{
   "code": 0,
   "data": "",
   "msg": ""
}
```

8.13 Upgrading the software

Interface description

Interface url	http://${ip}:${port}/api/form
Request Method	POST
Request Parameter Format	FORM
Interface Description	Upgrades the software

Request parameter

Parameter	Type	Description	Required	Example
msg_id	string	“1292”	Yes	
file	file	file	Yes	

Response parameter

Parameter	Type	Description
code	int	Result code (0:successful)
data	json	NULL
msg	string	Result description

Note: Wait until the system restarts automatically after the software is successfully upgraded.

Example

Request example

```
POST /form HTTP/1.1
Host: 10.5.2.53:8080
Content-Type: multipart/form-data; boundary=     WebKitFormBoundary7MA4YWxkTrZu0gW
cache-control: no-cache
Postman-Token:  37201329-0164-414c-85d3-65e821456ea2

Content-Disposition: form-data; name="file";
filename="/home/nablaworks/liuyang1_vendor/Downloads/install/nebula.tar.bz2
Content-Disposition: form-data; name="msg_id"

1292
------WebKitFormBoundary7MA4YWxkTrZu0gW--
```

Response example

```
{
  "code": 0,
  "data": "",
  "msg": ""
}
```

8.14 System restart

Interface description

Interface url	http://${ip}:${port}/api/json
Request Method	POST
Request Parameter Format	JSON
Interface Description	System restart

Request parameter

Parameter	Type	Description	Required	Example
msg_id	string	“1294”	Yes	

Response parameter

Parameter	Type	Description
code	int	Result code (0:successful)
data	json	Returned data
msg	string	Result description

Example

Request example

```
{
  "msg_id":"1294"
}
```

Response example

```
{
  "code": 0,
  "data": "",
  "msg": ""
}
```

8.15 Shut down the system

Interface description

Interface url	http://${ip}:${port}/api/json
Request Method	POST
Request Parameter Format	JSON
Interface Description	Shut down the system

Request parameter

Parameter	Type	Description	Required	Example
msg_id	string	“1337”	Yes	

Response parameter

Parameter	Type	Description
code	int	Result code (0:successful)
data	json	Returned data
msg	string	Result description

Example

Request example

```
{
  "msg_id":"1337"
}
```

Response example

```
{
  "code": 0,
  "data": "",
  "msg": ""
}
```

8.16 Set Image Data Push Switch

Interface description

Interface url	http://${ip}:${port}/api/json
Request Method	POST
Request Parameter Format	JSON
Interface Description	Set Image Data Push Switch

Request parameter

Parameter	Type	Description	Required	Example
msg_id	string	“1360”	Yes	
push_snap_img	int	Cutting Snap Image, 0 off, 1 on, Default is 1	No	
push_frame_img	int	Original Sanp Image, 0 off, 1 on, Default is 1	No	
push_compared_img	int	lib Image, 0 off, 1 on, Default is 1	No	

Response parameter

Parameter	Type	Description
code	int	Result code (0:successful)
data	json	Returned data
msg	string	Result description

Example

Request example

```
{
  "msg_id":"1360",
  "push_snap_img":0,
  "push_frame_img":0,
  "push_compared_img":0
}
```

Response example

```
{
  "code": 0,
  "data": "",
  "msg": ""
}
```

8.17 Querying Image Data Push Switch

Interface description

Interface url	http://${ip}:${port}/api/json
Request Method	POST
Request Parameter Format	JSON
Interface Description	QUERY Image Data Push Switch

Request parameter

Parameter	Type	Description	Required	Example
msg_id	string	“1361”	Yes	

Response parameter

Parameter	Type	Description
code	int	Result code (0:successful)
data	json	Returned data
msg	string	Result description

Field information(data)

Parameter	Type	Description
push_snap_img	int	Cutting Snap Image, 0 off, 1 on, Default is 1
push_frame_img	int	Original Sanp Image, 0 off, 1 on, Default is 1
push_compared_img	int	lib Image, 0 off, 1 on, Default is 1

Example

Request example

```
{
   "msg_id":"1361"
}
```

Response example

```
{
   "code": 0,
   "data": {  "push_snap_img":0,
      "push_frame_img":0,
      "push_compared_img":0
   },
   "msg": ""
```

```
}
```

8.18 Setting AES Crypto Algorithm Switch

Interface description

Interface url	http://${ip}:${port}/api/json
Request Method	POST
Request Parameter Format	JSON
Interface Description	Set AES Crypto Algorithm Switch

Request parameter

Parameter	Type	Description	Required	Example
msg_id	string	“1362”	Yes	
aes_crypto_flag	int	0 AES128, 1 AES256, Default is 0	No	

Response parameter

Parameter	Type	Description
code	int	Result code (0:successful)
data	json	Returned data
msg	string	Result description

Example

Request example

```
{
  "msg_id":"1362",
  "aes_crypto_flag":0
}
```

Response example

```
{
  "code": 0,
  "data": "",
  "msg": ""
}
```

8.19 Querying AES Crypto Algorithm Switch

Interface description

Interface url	http://${ip}:${port}/api/json
Request Method	POST

Request Parameter Format	JSON
Interface Description	Query AES Crypto Algorithm Switch

Request parameter

Parameter	Type	Description	Required	Example
msg_id	string	“1363”	Yes	

Response parameter

Parameter	Type	Description
code	int	Result code (0:successful)
data	json	Returned data
msg	string	Result description

Field information(data)

Parameter	Type	Description
aes_crypto_flag	int	AES Crypto Flag

Example

Request example

```
{
  "msg_id":"1363"
}
```

Response example

```
{
  "code": 0,
  "data":
    { "aes_crypto_flag":0
  },
  "msg": ""
}
```

8.20 Querying the device ID

Interface description

Interface url	http://${ip}:${port}/api/json
Request Method	POST
Request Parameter Format	JSON
Interface Description	Queries the device ID

Request parameter

Parameter	Type	Description	Required	Example
msg_id	string	“1301”	Yes	

Response parameter

Parameter	Type	Description
code	int	Result code (0:successful)
data	json	Returned data
msg	string	Result description

Field information(data)

Parameter	Type	Description
device_id	string	Device ID

Example

Request example

```
{
  "msg_id":"1301"
}
```

Response example

```
{
  "code": 0,
  "data": {
    "device_id": "test_device"
  },
  "msg": ""
}
```

8.21 Modifying the device ID

Interface description

Interface url	http://${ip}:${port}/api/json
Request Method	POST
Request Parameter Format	JSON
Interface Description	Modifies the device ID.

Request parameter

Parameter	Type	Description	Required	Example
msg_id	string	“1300”	Yes	
device_id	string	Device ID	Yes	

Response parameter

Parameter	Type	Description
code	int	Result code (0:successful)
data	json	NULL
msg	string	Result description

Example

Request example

```
{
  "msg_id":"1300",
  "device_id":"test_device"
}
```

Response example

```
{
  "code": 0,
  "data": "",
  "msg": ""
}
```

8.22 Restoring default configuration

Interface description

Interface url	http://${ip}:${port}/api/json
Request Method	POST
Request Parameter Format	JSON
Interface Description	Restores the default configuration (except general settings, network settings and security settings, other models are restored to default configuration)

Request parameter

Parameter	Type	Description	Required	Example
msg_id	string	“1317”	Yes	

Response parameter

Parameter	Type	Description
code	int	Result code (0:successful)
data	string	NULL
msg	string	Result description

Example

Request example

```
{
  "msg_id":"1317"
}
```

Response example

```
{
  "code": 0,
  "data": "",
  "msg": ""
}
```

8.23 Opening web-access function

Interface description

Interface url	http://${ip}:${port}/api/json
Request Method	POST
Request Parameter Format	JSON
Interface Description	Opens the Web-access funtion

Request parameter

Parameter	Type	Description	Required	Example
msg_id	string	“1315”	Yes	

Response parameter

Parameter	Type	Description
code	int	Result code (0:successful)
data	string	NULL
msg	string	Result description

Example

Request example

```
{
  "msg_id":"1315"
}
```

Response example

```
{
  "code": 0,
  "data": "",
  "msg": ""
```

```
}
```

8.24 Closing web-access function

Interface description

Interface url	http://${ip}:${port}/api/json
Request Method	POST
Request Parameter Format	JSON
Interface Description	Closes the Web-access function

Request parameter

Parameter	Type	Description	Required	Example
msg_id	string	“1316”	Yes	

Response parameter

Parameter	Type	Description
code	int	Result code (0:successful)
data	string	NULL
msg	string	Result description

Example

Request example

```
{
  "msg_id":"1316"
}
```

Response example

```
{
  "code": 0,
  "data": "",
  "msg": ""
}
```

8.25 Querying CPU information

Interface description

Interface url	http://${ip}:${port}/api/json
Request Method	POST
Request Parameter Format	JSON
Interface Description	Quries CPU information

Request parameter

Parameter	Type	Description	Required	Example
msg_id	string	“1313”	Yes	

Response parameter

Parameter	Type	Description
code	int	Result code (0:successful)
data	json	Returned data
msg	string	Result description

Field information(data)

Parameter	Type	Description
cpu_type	string	CPU Type
cpu_usage	double	CPU usage
kernel_num	int	CPU kernel number

Example

Request example

```
{
  "msg_id":"1313"
}
```

Response example

```
{
  "code": 0,
  "data": {
    "cpu_type": "ARMv8 Processor rev 3 (v8l)",
    "cpu_usage": 14.450867052023122,
    "kernel_num": 6
  },
  "msg": ""
}
```

8.26 Querying memory Information

Interface description

Interface url	http://${ip}:${port}/api/json
Request Method	POST
Request Parameter Format	JSON
Interface Description	Queries memory information

Request parameter

Parameter	Type	Description	Required	Example
msg_id	string	“1314”	Yes	

Response parameter

Parameter	Type	Description
code	int	Result code (0:successful)
data	json	Returned data
msg	string	Result description

Field information(data)

Parameter	Type	Description
memory_usage	double	Memory usage
total_memory	int	Total memory

Example

Request example

```
{
  "msg_id":"1314"
}
```

Response example

```
{
  "code": 0,
  "data": {
    "memory_usage": 78.615106141826942,
    "total_memory": 8232378368
  },
  "msg": ""
}
```

8.27 Set User Data Stored Period

Interface description

Interface url	http://${ip}:${port}/api/json
Request Method	POST
Request Parameter Format	JSON
Interface Description	Set User Data Stored Period

Request parameter

Parameter	Type	Description	Required	Example
msg_id	string	“1902”	Yes	

store_obj	int	Stored Object Type. 1 Image Libraries, 33 Snap Images.	Yes	
saved_time	int	Stored During of the Images. [0~360)	Yes	

Response parameter

Parameter	Type	Description
msg_id	string	Msg id
result	int	Result of Setting User Data Stored Period. 0 OK, other Error.

Example

Request example

```
{
   "msg_id":"1902",
            "store_obj":1,
            "saved_time":30
}
```

Response example

```
{
   "msg_id": "1902",
   "result": 0
}
```

8.28 Query User Data Stored Period

Interface description

Interface url	http://${ip}:${port}/api/json
Request Method	POST
Request Parameter Format	JSON
Interface Description	Query User Data Stored Period

Request parameter

Parameter	Type	Description	Required	Example
msg_id	string	“1903”	Yes	
store_obj	int	Stored Object Type. 1 Image	Yes	

		Libraries, 33 Snap Images.		

Response parameter

Parameter	Type	Description
msg_id	string	“1903”
result	int	Result of Query User Data Stored Period. 0 OK, other Error.
store_obj	int	Stored Object Type. 1 Image Libraries, 33 Snap Images.
saved_time	int	Stored During of the Images. [0~360)

Example

Request example

```
{
   "msg_id":"1903",
            "store_obj":1
}
```

Response example

```
{
   "msg_id":"1903",
            "result":0,
            "store_obj":1,
            "saved_time":30
}
```

9 Http Configuration Interfaces

9.1 Querying http configuration

Interface description

Interface url	http://${ip}:${port}/api/json
Request Method	POST
Request Parameter Format	JSON
Interface Description	Queries Http configuration

Request parameter

Parameter	Type	Description	Required	Example
msg_id	string	“1299”	Yes	

Response parameter

Parameter	Type	Description
code	int	Result code (0:successful)
data	json	Returned data
msg	string	Result description

Field information(data)

Parameter	Type	Description
key	string	Http Key
url	string	Http Url

Example

Request example

```
{
  "msg_id":"1299"
}
```

Response example

```
{
  "code": 0,
  "data": {
    "key":  "db504129-85e0-49ca-8626-f639b8d09846",
    "url": "http://sgdzpic.3322.org:8088/agbox/device/snap"
  },
  "msg": ""
}
```

9.2 Modifying http configuration

Interface description

Interface url	http://${ip}:${port}/api/json
Request Method	POST
Request Parameter Format	JSON
Interface Description	Modifies HTTP configuration

Request parameter

Parameter	Type	Description	Required	Example
msg_id	string	“1298”	Yes	
url	string	http url	Yes	google.com
key	string	http key	Yes	google-key

Response parameter

Parameter	Type	Description
code	int	Result code (0:successful)
data	json	NULL
msg	string	Result description

Example

Request example

```
{
  "msg_id":"1298",
  "url":"google.com",
  "key":"google-key"
}
```

Response example

```
{
  "code": 0,
  "data": "",
  "msg": ""
}
```

9.3 Pushing http heartbeat information

Interface description

Interface url	URL of the HTTP server that receives pushed heartbeat information
Interface Description	Pushes heartbeat information to the server with the specified IP address.

Response parameter

Parameter	Type	Description
msg_id	string	“775”
data	json	Returned data

Field information(data)

Parameter	Type	Description
trigger	string	Time of heartbeat information
device_id	string	Device ID that maps heartbeat information

Response example

```
{
```

```
  "data" : {
    "device_id" : "test_device",
    "trigger" : "2019-01-24 15:05:05"
  },
  "msg_id" : 775
}
```

9.4 Pushing http alert deployment results

Interface description

Interface url	URL of the HTTP server that receives pushed results
Interface Description	Pushes alert deployment results to the server with the specified IP address

Response parameter

Parameter	Type	Description
key	string	Used in verification of docking equipment
json	json	Returned data
snap	file	Face/Body capture image
img	file	Matched image in the portrait database
snap_frame	file	Large scene corresponding to Face/Body image

Field information(json)

Parameter	Type	Description
msg_id	string	“774”
data	json	Returned data

Field information(data)

Parameter	Type	Description
camera_name	string	Camera name
device_id	string	Device ID
channel	int	Channel; value range:[1,16] of SenseNebula-M4s, [1,32] of SenseNebula-M8s
position	string	Camera position
threshold	int	Comparison threshold (camera-bound); value range:[0,100].
img_id	string	ID of the matched face image in the portrait database
img_path	string	Path of the image in the portriat database (This parameter is reserved and

		indicates the image name.) Concatenate the URL as follows when downloading the image: http://${ip}:${port}/ws/img_path
lib_id	int	Database ID
lib_name	string	Database name
lib_type	int	Database type (1:key personnel database, 2:whitelist database)
person_addr	string	Address
person_age	string	Age
person_gender	string	Gender (0:female, 1:male)
person_idcard	string	Identity code
person_name	string	Name
snap_id	string	Capture ID
similarity	int	Score of similarity with the image in the source database; value range:[0,100]; the default value is –1.
quality	int	Quality score; value range:[0,100]
pos_top	int	Y-coordinate of the face in the upper-left position
pos_bottom	int	Y-coordinate of the face in the lower-right position
pos_left	int	X-coordinate of the face in the upper-left position
pos_right	int	X-coordinate of the face in the lower-right position
snap_feat	string	Features of the capture
snap_path	string	Path of the capture (This parameter is reserved and indicates the image name) Concatenate the URL as follows when downloading the image: http://${ip}:${port}/ws/snap_path
trigger	string	Trigger (capture time)
obj_label	int	Label (1:face, 2:body, 8:firesmoke, 9:fire extinguisher)
alive_type	int	Alive type, 0：Unknown; 1: Non alive; 2: Alive
face_attr	json	Face attributes with obj_label=1, for details see in appendix B
body_attr	json	Body attributes with obj_label=2, for details see in appendix C
ranking	int	Similarity coefficient of the face image that maps the ranking; value range:[0,100]
alarm_type	int	Face alarm type (0: no alarm, 1: key personnel/allowlist alarm, 3: hit stranger db alarm)
appear_count	int	Current occurrences
event_type	int	Event alarm type (0: no alarm, 1: wandering alarm, 2: staying alarm)
wander_channels	string	The set wandering places for the wandering alarm
wander_thresHold	int	The threshold of occurrences number
wander_devi	string	The trigger place for wandering alarm

ceID		
wander_trigger	string	Wandering time for wandering alarm
stranger_appear_channel	int	Camera channel of the captured wandering stranger
events_type	string	alarm events, (387:smoke, 388:doze, 389:call, 390: no helmet,391:no work cloth,394:invade,395:leave396:over boundary,398:crowd,641:fire,705:fire extinguisher)
over_boundary_num1	int	over boundary direction 1 num
over_boundary_num2	int	over boundary direction 2 num
crowd_num	int	crowd num
crowd_threshold	int	crowd threshold
leave_time_threshold	int	leave time threshold
leave_num_threshold	int	leave num threshold
helmet_type	string	helmet type: bonnet,cap,bucket_hat,st_hard_hat
work_cloth_color	string	work cloth color:black,white,gray,red,yellow,blue,green,purple
work_cloth_type	string	work cloth type:st_worker_uniform,st_common_clothing,st_office_uniform,st_chef_uniform,st_medical_uniform,st_police_uniform,st_firefighter_uniform,st_reflective_clothing

Note 1: Configure the corresponding HTTP server before receiving alert deployment results.

Note 2: In the pushed results, img, snap, and snap_frame are files. To download the image in the source database, concatenate the download URL based on img_path. To download the capture, concatenate the download URL based on snap_path.

Response example


```
{
  "key":"",
  "snap":"",
  "img":"",
  "snap_frame":"",
  "json": {
    "data": {
      "face_attr": {
        "cap_style": "hat_style_type_none", "gender_code": "male",
        "glass_style": "glasses_style_type_none",
```

```
            "mustache_style": "mustache_style_type_none",
            "st_respirator": "st_respirator_mouth",
            "respirator_color":  "color_type_none",   "st_age":
            "st_adult",
            "st_helmet_style": "st_helmet_style_type_none",  "st_age_value":
            "24.000000", "st_expression": "st_calm",  "st_expression": "st_calm"
         },
         "camera_name": "test", "device_id": "13",
         "channel": 19,
         "img_id": "fa48b56f-9aed-43bc-ad20-9ee164873473",
         "img_path": "img_pic",
         "lib_id": 3,  "lib_name":
         "Facelib",  "lib_type": 2,
         "obj_label":1,
         "person_addr": "",
         "person_age": "",
         "person_gender": "",
         "person_idcard": "",
         "person_name": "lousongya_000000000000000000_1",
         "position": "",
         "similarity": 96,
         "quality":1,
         "pos_top":1000,
         "pos_bottom":100,
         "pos_left":100,
         "pos_right":1000,
         "snap_feat":
"124.000000,391.000000,280.000000,327.000000,239.000000,239.000000,242.000000,244.000000,",
         "snap_id": "92ed1a18-2aaf-4362-b5b4-1fd6750e1dd9",
         "snap_path": "snap_pic",
         "threshold": 30,
         "trigger": "2019-01-24 13:43:32",
         "events_type":"387,389",
         "alive_type": 2
      },
      "msg_id": 774
   }
}
```

9.5 Interface of Sending vehicle message by HTTP

Interface description

Interface url	http server address for receiving message
Interface Description	Interface of Sending vehicle message by HTTP

Response parameter

Parameter	Type	Description
key	string	key
json	json	result data
snap	file	snap image
img	file	reserved
snap_frame	file	frame
vehicle_num_img	file	license plate image

Field information(json)

Parameter	Type	Description
msg_id	string	“774”
data	json	Returned data

Field information(data)

Parameter	Type	Description
camera_name	string	camera name
device_id	string	device ID
channel	int	channel；M4s：[1,16]，M8s：[1,32]
position	string	camera position
threshold	int	reserved, threshould of similarity：[0,100]
notify	int	is alarm message（0: No，1: Yes）
img_id	string	the uuid of license plate in database which has the same license number with the snapped one
img_path	string	reserved
lib_id	int	database id
lib_name	string	database name
owner_age	string	owner age
owner_gender	string	owner gender（0: female，1:male）
owner_idcard	string	owner id card

owner_name	string	owner name
quality	int	Quality score, value range:[0,100]
pos_top	int	top of bounding box
pos_bottom	int	bottom of bounding box
pos_left	int	left of bounding box
pos_right	int	right of bounding box
vehicle_num_left	int	license plate coordinate in the vehicle image
vehicle_num_top	int	license plate coordinate in the vehicle image
vehicle_num_right	int	license plate coordinate in the vehicle image
vehicle_num_bottom	int	license plate coordinate in the vehicle image
snap_id	string	Capture ID
snap_path	string	snap image path
vehicle_number	string	license number
trigger	string	trigger time of snaping
obj_label	int	3：non-motor vehicle, 4：motor vehicle
vehicle_attr	json	vehicle attribute，Appendix D

Note 1：Configure the corresponding HTTP server before receiving alert deployment results.

Note 2：in this message，img、snap、snap_frame are files, and img, snap can be download according to img_path and snap_path.

Response example


```
{
  "key":"",
  "snap":"",
  "img":"",
  "snap_frame":"",
  "json": {
    "data":
      { "vehicle_attr":
      {
        "Brand":"丰田",
        "CarDirection":"Front",
        "CarFilter":"Easy",
        "SubType":"丰田-凯美瑞",
        "Type":"Car",
        "vehicle_class": "suv",
        "vehicle_color": "black"
```

```
        },
        "camera_name": "test",
        "device_id": "13",
        "channel": 19,
        "img_id": "fa48b56f-9aed-43bc-ad20-9ee164873473",
        "img_path": "img_pic",
        "lib_id": 3,
        "lib_name": "Vehiclelib",
        "obj_label":3,
        "owner_age": "",
        "owner_gender": "",
        "owner_idcard": "",
        "owner_name": "lousongya_000000000000000000_1",
        "position": "",
        "quality":85,
        "pos_top":1000,
        "pos_bottom":100,
        "pos_left":100,
        "pos_right":1000,"snap_id": "92ed1a18-2aaf-4362-b5b4-1fd6750e1dd9",
        "snap_path": "snap_pic",
        "verhicle_number":"YNLV880",
        "threshold": 80,
        "trigger": "2020-01-24 13:43:32"
    },
    "msg_id": 774
  }
}
```


10 Https Configuration Interfaces

10.1 Uploading the Https certificate and key

Interface description

Interface url	http://${ip}:${port}/api/form
Request Method	POST
Request Parameter Format	FORM
Interface Description	Uploads the Https certificate and key

Request parameter

Parameter	Type	Description	Required	Example
msg_id	string	“1302”	Yes	

file	file	https file	Yes	
file_type	string	https file type; crt: Certificate; key: key	Yes	

Response parameter

Parameter	Type	Description
code	int	Result code (0:successful)
data	json	NULL
msg	string	Result description

Example

Request example

```
Input Files
{
  'file':name='server.crt'
}
data
{
  "msg_id":"1302",
  "file_type":"crt"
}
```

Response example

```
{
  "code": 0,
  "data": "",
  "msg": ""
}
```

Example

Request example

```
Input Files
{
  'file':name='server.key'
}
data
{
  "msg_id":"1302",
  "file_type":"key"
}
```

Response example

```
{
  "code": 0,
  "data": "",
  "msg": ""
}
```

10.2 Deleting the https certificate and key

Interface description

Interface url	http://${ip}:${port}/api/json
Request Method	POST
Request Parameter Format	JSON
Interface Description	Deletes the HTTPS certificate and key

Request parameter

Parameter	Type	Description	Required	Example
msg_id	string	“1303”	Yes	
file_type	string	HTTPS file type crt: certificate; key: key	Yes	“crt” or “key”

Response parameter

Parameter	Type	Description
code	int	Result code (0:successful)
data	json	NULL
msg	string	Result description

Example

Request example

```
{
  "msg_id": "1303",
  "file_type":"crt"
}
```

Response example

```
{
  "code": 0,
  "data": "",
  "msg": ""
}
```

Example

Request example

```
{
  "msg_id": "1303",
  "file_type":"key"
}
```

Response example

```
{
  "code": 0,
  "data": "",
  "msg": ""
}
```

10.3 Querying the HTTPS certificate and key

Interface description

Interface url	http://${ip}:${port}/api/json
Request Method	POST
Request Parameter Format	JSON
Interface Description	Queries the HTTPS certificate and key

Request parameter

Parameter	Type	Description	Required	Example
msg_id	string	“1304”	Yes	

Response parameter

Parameter	Type	Description
code	int	Result code (0:successful)
data	json	Returned data
msg	string	Result description

Field information(data)

Parameter	Type	Description
https_crt	string	Certificate that is issued after HTTPS authentication
https_key	string	HTTPS private key

Example

Request example

```
{
```

```
  "msg_id": "1304"
}
```

Response example

```
{
  "code": 0,
  "data":
    { "https_crt":server.crt,
    "https_key":server.key
  }
  "msg": ""
}
```

10.4 Switching to HTTPS or HTTP

Interface description

Interface url	http://${ip}:${port}/api/json
Request Method	POST
Request Parameter Format	JSON
Interface Description	Switches to HTTPS or HTTP

Request parameter

Parameter	Type	Description	Required	Example
msg_id	string	“1305”	Yes	
prot_mode	string	Switching to HTTPS or HTTP mode	Yes	“http” or “https”

Response parameter

Parameter	Type	Description
code	int	Result code (0:successful)
data	json	NULL
msg	string	Result description

Note: Wait until the system restarts automatically after it switches to HTTPS or HTTP mode.

Example

Request example

```
{
  "msg_id": "1305",
  "prot_mode":"http"
}
```

Response example

```
{
  "code": 0,
  "data": ""
  "msg": ""
}
```

10.5 Querying the current mode (HTTPS or HTTP)

Interface description

Interface url	http://${ip}:${port}/api/json
Request Method	POST
Request Parameter Format	JSON
Interface Description	Queries the current mode (HTTPS or HTTP)

Request parameter

Parameter	Type	Description	Required	Example
msg_id	string	“1306”	Yes	

Response parameter

Parameter	Type	Description
code	int	Result code (0:successful)
data	json	Returned results
msg	string	Result description

Field information(data)

Parameter	Type	Description
https_stat	string	http or https

Example

Request example

```
{
  "msg_id": "1306"
}
```

Response example

```
{
  "code": 0,
  "data":
    { "https_stat":"http"
  },
  "msg": ""
}
```

11 Interfaces related to alert deployment result push through WebSocket

11.1 Querying the key for alert deployment result push through WebSocket

Interface description

Interface url	http://${ip}:${port}/api/json
Request Method	POST
Request Parameter Format	JSON
Interface Description	Queries the key for alert deployment result push through WebSocket

Request parameter

Parameter	Type	Description	Required	Example
msg_id	string	“1286”	Yes	

Response parameter

Parameter	Type	Description
code	int	Result code (0:successful)
data	json	Returned data
msg	string	Result description

Field information(data)

Parameter	Type	Description
key	string	Subscription key

Example

Request example

```
{
  "msg_id":"1286"
}
```

Response example

```
{
  "code": 0,
  "data": {
    "key":  "04af251e-91d4-43bd-b8b0-8c8c6b3245bb"
```

```
  },
  "msg": ""
}
```

11.2 Pushing alert deployment results through WebSocket

Interface description

Interface url	ws://${ip}:${port}/ws/
Interface Description	Pushes alert deployment results through WebSocket.

Request parameter

Parameter	Type	Description	Required	Example
msg_id	string	“776”	Yes	
key	string	Subscription key	Yes	

Response parameter

Parameter	Type	Description
code	int	Result code (0:successful)
data	json	Returned data
msg_id	string	777 (unique identifier of an image push message)
msg	string	Result description

Field information(data)

Parameter	Type	Description
camera_name	string	Camera name
device_id	string	Device ID
channel	int	Channel; value range:[1,16] of SenseNebula-M4s, [1,32] of SenseNebula-M8s
threshold	int	Comparison threshold (camera-bound); value range:[0,100].
img_id	string	ID of the matched face image in the database
img	str	Image in the source database (base64-transcoded)

	ing	
img_path	string	Path of the image in the source database, concatenate the URL as follows when downloading the image: http://${ip}:${port}/ws/img_path
lib_id	int	Database ID
lib_name	string	Database name
lib_type	int	Database type (1:key personnel database, 2:whitelist database)
person_addr	string	Address
person_age	string	Age
person_gender	string	Gender (0:female; 1:male)
person_idcard	string	Identity code
person_name	string	Name
position	string	Camera position
ranking	int	Similarity ranking
similarity	int	Similarity score; value range:[0,100]; The default value is –1.
quality	int	Quality score; value range:[0,100]
pos_top	int	Y-coordinate of the face in the upper-left position
pos_bottom	int	Y-coordinate of the face in the lower-right position
pos_left	int	X-coordinate of the face in the upper-left position
pos_right	int	X-coordinate of the face in the lower-right position
snap_id	string	Capture ID
snap_buf	string	Face/Body capture image (base64-transcoded)
snap_feat	str	Features of the capture

	ing	
snap_path	string	Capture path Concatenate the URL as follows when downloading the image: http://${ip}:${port}/ws/snap_path
snap_frame	string	Large scene corresponding to Face/Body image (base64-transcoded)
trigger	string	Trigger time
obj_label	int	Label (1:face; 2:body, 8:firesmoke, 9:fire extinguisher)
alive_type	int	Alive type, 0： Unknown; 1: Non alive; 2: Alive
face_attr	json	Face attributes with obj_label=1, for details see in appendix B
body_attr	json	Body attributes with obj_label=2, for details see in appendix C
alarm_type	int	Face alarm type (0: no alarm, 1: key personnel/allowlist alarm, 3: hit stranger db alarm)
appear_count	int	Current occurrences
event_type	int	Event alarm type (0: no alarm, 1: wandering alarm, 2: staying alarm)
wander_channels	string	The set wandering places for the wandering alarm
wander_thresHold	int	The threshold of occurrences number
wander_deviceID	string	The trigger place for wandering alarm
wander_trigger	string	Wandering time for wandering alarm
stranger_appear_channel	int	Camera channel of the captured wandering stranger
events_type	string	alarm events, (387:smoke, 388:doze, 389:call, 390: no helmet,391:no work cloth,394:invade,395:leave396:over boundary,398:crowd,641:fire,705:fire extinguisher)
over_boundary_num1	int	over boundary direction 1 num
over_bound	int	over boundary direction 2 num

ary_num2		
crowd_num	int	crowd num
crowd_threshold	int	crowd threshold
leave_time_threshold	int	leave time threshold
leave_num_threshold	int	leave num threshold
helmet_type	string	helmet type: bonnet,cap,bucket_hat,st_hard_hat
work_cloth_color	string	work cloth color:black,white,gray,red,yellow,blue,green,purple
work_cloth_type	string	work cloth type:st_worker_uniform,st_common_clothing,st_office_uniform,st_chef_uniform,st_medical_uniform,st_police_uniform,st_firefighter_uniform,st_reflective_clothing

Note 1: In the pushed results, img, snap, and snap_frame are base64- transcoded. To download the image in the source database, concatenate the download URL based on img_path. To download the capture, concatenate the download URL based on snap_path.

Example

Request example

```
{
  "key":"7410342e-e4e8-47de-a92f-2fd918eb3d6a",
  "msg_id":"776"
}
```

Response example

```
{
  "code": 0,
  "msg": ""
}
{
  "code": 0,
  "data": {  "camera_name":
    "test",  "device_id":
    "123",  "channel": 1,
    "img_id":  "3c370627-d082-4c53-b156-e9380163d22f",
    "img_path": "img/12_3c370627-d082-4c53-b156-e9380163d22f.jpg",
    "lib_id": 12,
```

```
      "lib_name": "Marry",
      "lib_type": 1,
      "person_addr": "",
      "person_age": "",
      "person_gender": "",
      "person_idcard": "",
      "person_name": "1201020000000589932",
      "position": "",
      "ranking": 1,
      "similarity": 38,
      "quality":1,
      "pos_top":1000,
      "pos_bottom":100,
      "pos_left":100,
      "pos_right":1000,
      "snap_buf": "",  "snap_feat":
      "315.000000,32.000000,",
      "snap_id": "5672fb61-74ab-4037-8a37-a3c0abf417a3",   "snap_path":
      "record/5672fb61-74ab-4037-8a37-a3c0abf417a3.jpg",  "threshold":
      60,
      "trigger": "2019-01-24 14:09:21",
      "snap_frame":"",
      "obj_label":1,
      "alive_type": 2,
      "events_type":"387,389",
      "face_attr": {
         "cap_style": "hat_style_type_none",
         "gender_code": "female",
         "glass_style": "glasses_style_type_none",
         "mustache_style": "mustache_style_type_none",
         "st_respirator": "st_respirator_mouth",
         "respirator_color":  "color_type_none",   "st_age":
         "st_adult",
         "st_age_value": "26.000000",
         "st_expression": "st_angry"
      },
   },
   "msg": "",
   "msg_id": "777"
}
```

11.3 Interface of Sending vehicle message by WebSocket

Interface description

Interface url	ws://${ip}:${port}/ws/
Interface Description	Interface of Sending vehicle message by WebSocket

Request parameter

Parameter	Type	Description	Required	Example
msg_id	string	"776"	Yes	
key	string	subscribe key	Yes	

Response parameter

Parameter	Type	Description
code	int	error code（0:successfully）
data	json	result data
msg_id	string	"777"（message uuid）
msg	string	result description

Field information(data)

Parameter	Type	Description
camera_name	string	camera name
device_id	string	device ID
channel	int	channel；M4s：[1,16]，M8s：[1,32]
position	string	camera position
threshold	int	reserved, threshould of similarity：[0,100]
notify	int	is alarm message（0: No，1: Yes）
img_id	string	the uuid of license plate in database which has the same license number with the snapped one
img_path	string	reserved
img	string	reserved
lib_id	int	database id
lib_name	string	database name
owner_age	string	owner age
owner_gender	string	owner gender（0: female，1:male）
owner_idcard	string	owner id card

owner_name	string	owner name
is_vip	int	is VIP（0:No，1:Yes）
quality	int	Quality score, value range:[0,100]
pos_top	int	top of bounding box
pos_bottom	int	bottom of bounding box
pos_left	int	left of bounding box
pos_right	int	right of bounding box
vehicle_num_left	int	license plate coordinate in the vehicle image
vehicle_num_top	int	license plate coordinate in the vehicle image
vehicle_num_right	int	license plate coordinate in the vehicle image
vehicle_num_bottom	int	license plate coordinate in the vehicle image
snap_id	string	Capture ID
snap_path	string	snap image path
snap_buf	string	vehicle image (base64 encoded)
snap_frame	string	frame(base64 encoded)
vehicle_num_img	string	license plate image(base64 encoded)
vehicle_number	string	license number
trigger	string	trigger time of snaping
obj_label	int	3：non-motor vehicle, 4：motor vehicle
vehicle_attr	json	vehicle attribute，Appendix C1

Note：img、snap、snap_frame are encoded with base64 in this message。and the snap image and database image can be downloaded according to snap_path and img_path.

Example

Request example

```
{
  "key":"7410342e-e4e8-47de-a92f-2fd918eb3d6a",
  "msg_id":"776"
}
```

Response example

```
{
  "code": 0,
  "data": {
```

```
        "vehicle_attr": {  "Brand":"
           丰田",
           "CarDirection":"Front",
           "CarFilter":"Easy",
           "SubType":"丰田-凯美瑞",
           "Type":"Car",
           "vehicle_class": "suv",
           "vehicle_color": "black"
        },
        "camera_name": "test",
        "device_id": "13",
        "channel": 19,
        "img_id": "fa48b56f-9aed-43bc-ad20-9ee164873473",
        "img_path": "img_pic",
        "lib_id":  3,   "lib_name":
        "车辆底库",
        "obj_label":3,
        "owner_age": "",
        "owner_gender": "",
        "owner_idcard":  "",
        "owner_name": "娄松亚_000000000000000000_1",
        "position": "",
        "quality":85,
        "pos_top":1000,
        "pos_bottom":100,
        "pos_left":100,
        "pos_right":1000,"snap_id": "92ed1a18-2aaf-4362-b5b4-1fd6750e1dd9",
        "snap_path": "snap_pic",
        "verhicle_number":"豫 NLV880",
        "threshold": 80,
        "trigger": "2020-01-24 13:43:32"
    },
    "msg": "",
    "msg_id": "777"
}
```


11.4 Pushing statistics results through WebSocket

Interface description

Interface url	ws://${ip}:${port}/ws/
Interface Description	Pushing statistics results through WebSocket

Request parameter

Parameter	Type	Description	Required	Example
msg_id	string	“778”	Yes	
key	string	Subscription key	Yes	

Response parameter

Parameter	Type	Description
code	int	Result code (0:successful)
data	json	Returned data
msg_id	string	“778”(unique identifier of an image push message)
msg	string	Result description

Field information(data)

Parameter	Type	Description
key_person_cnt	int	key person count
white_list_cnt	int	white list person count
stranger_cnt	int	stranger count

Note 1: count results.

Example

Request example

```
{
  "key":"7410342e-e4e8-47de-a92f-2fd918eb3d6a",
  "msg_id":"778"
}
```

Response example

```
{
  "code": 0,
  "msg": ""
}
{
  "code": 0,
  "data":
    { "key_person_cnt": 0,
    "white_list_cnt": 0,
    "stranger_cnt": 0,
  },
  "msg": "recvive websocket msg"  "msg_id":
  "778"
}
```

12 Events management interfaces

12.1 Query the binding relationship

Interface description

Interface url	http://${ip}:${port}/api/json
Request Method	POST
Request Parameter Format	JSON
Interface Description	Queries the binding relationship between camera and controller

Request parameter

Parameter	Type	Description	Required	Example
msg_id	string	“1537”	Yes	

Response parameter

Parameter	Type	Description
code	int	Result code (0:successful)
data	json list	Returned data
msg	string	Result description

Field information(data)

Parameter	Type	Description
id	int	Relationship id, unique identification of the binding relation between camera and controller
camera_channel	int	Camera channel number
camera_name	string	Camera name
dev_ip	string	Controller ip
dev_port	int	Controller port
dev_name	string	Controller name
dev_delay_time	int	Controller status delay, value:[1, 10]
dev_tag	string	Reserved fields
dev_type	string	Controller Type, Relay controller corresponds to type 'relay'
work_status	int	Relay online status (0:offline, 1:online)

Example

Request example


```
{
  "msg_id":"1537"
}
```


Response example


```
{
  "code": 0,
  "data": [
    {
      "camera_channel": 1,  "camera_name":
      "subway1_1.264",  "dev_ip":
      "127.0.0.2",
      "dev_name": "test3",
      "dev_port": 5679,
      "dev_tag": "1",
                    "dev_delay_time":"1",
      "dev_type": "relay",  "work_status":1,
      "id": 2
    }
  ],
  "msg": ""
}
```


12.2 Creating binding relationship

Interface description

Interface url	http://${ip}:${port}/api/json
Request Method	POST
Request Parameter Format	JSON
Interface Description	Creates binding relationship between camera and controller

Request parameter

Parameter	Type	Description	Required	Example
msg_id	string	“1538”	Yes	
camera_channel	int	Camera channel number	Yes	
dev_ip	string	Controller ip	Yes	10.0.0.2
dev_port	int	Controller port 0-65535	Yes	1234

dev_delay_time	int	Controller status delay, value:[1, 10]	Yes	1
dev_name	string	Controller name	No	
dev_tag	string	Reserved fields	No	

Response parameter

Parameter	Type	Description
code	int	Result code (0:successful)
data	json	Returned data
msg	string	Result description

Field information(data)

Parameter	Type	Description
id	int	Relationship id, unique identification of the binding relation between camera and controller
camera_channel	int	Camera channel number
camera_name	string	Camera name
dev_ip	string	Controller ip
dev_port	int	Controller port
dev_delay_time	int	Controller status delay, value:[1, 10]
dev_name	string	Controller name
dev_tag	string	Reserved fields
dev_type	string	Controller Type，Relay controller corresponds to type 'relay'

Example

Request example

```
{
  "msg_id":"1538",
  "camera_channel":1,
  "dev_ip":"10.0.0.2",
  "dev_port":1234,
  "dev_name":"test3",
        "dev_delay_time":"1"
}
```

Response example

```
{
  "code": 0,
  "data": {
     "camera_channel": 1,
     "camera_name": "xxx",
     "dev_ip": "127.0.0.2",
     "dev_name": "test3",
     "dev_port": 5679,
                    "dev_delay_time":"1",
     "dev_tag": "1",
     "dev_type": "relay",
     "id": 1
  },
  "msg": ""
}
```


12.3 Deleting the binding relationship

Interface description

Interface url	http://${ip}:${port}/api/json
Request Method	POST
Request Parameter Format	JSON
Interface Description	Deleting the binding relationship between camera and controller

Request parameter

Parameter	Type	Description	Required	Example
msg_id	string	“1539”	Yes	
id	int	Relationship id, unique identification of the binding relation between camera and controller	Yes	

Response parameter

Parameter	Type	Description
code	int	Result code (0:successful)
data	string	
msg	string	Result description

Example

Request example

```
{
   "msg_id":"1539",
   "id":1
}
```

Response example

```
{
   "code": 0,
   "data":"",
   "msg": ""
}
```

Appendix A: Error code

-1	Unknown error
-105	Data does not exist
-108	User does not exist
-109	Camera does not exist
-110	No face image in the database
-113	Username or password error
-114	User already exists
-115	The channel already has a camera
-116	The face image already exists
-117	The face database does not exist
-118	The number of face database has reached the limit
-119	The face database already exists
-122	The number of images has reached the limit
-123	The database has been bound to the camera please untie it before Deleting
-126	No such face image in the database
-610	Snap mode switch failed
-140	User does not have permission to operate
-141	User is not logged in
-142	User password too weak
-143	Role id not found
-144	User limit exceed
-145	Current user name or password incorrect
-146	Role cannot be deleted cause users exist in the role
-147	Role has already exist
-148	Role limit exceed
-151	Insufficient computing power
-153	the vehicle number already exists
-154	the vehicle number not exists
-155	The vehicle database does not exist
-1100	'img_id' exists.
-1101	No face verified
-1102	The face image quality is too low
-1103	inner broken
-1104	Incorrect image format or broken image
-1105	unknown lib

-1106	none lib in the message
-1107	empty lib
-1108	unknown lib
-1109	multiface in the image
-1110	the image size out of limmit
-2100	No content to export
-2754	The img_id is too long
-2824	Current user cannot delete himself
-3596	Parameter input error
-3599	'msg_id' error
-3603	Upgrade failed
-3606	Network configuration error
-3607	The current system is already in this mode
-3608	Switch mode failed
-3614	NTP server access error
-3616	The crt or key file cannot be imported in HTTPS mode
-2617	The crt or key file cannot be deleted in HTTPS mode
-3619	Toggle https/http failed
-3620	The capture strategy cannot be modified in Image mode
-3524	System internal error
-3625	crt or key file not yet imported

Note:

The reasons for -3606 network configuration errors include:

1. Being in the same subnet with other network ports;
2. DHCP failure;
3. Gateway configuration error;
4. Subnet mask configuration error

Appendix B: Face attribute description

Face Attribute	Description	Enum	Description
st_age	age grouping	st_old, st_adult, st_child	
mustache_style	mustache	mustache_style_type_none, whiskers	No mustache; Mustache
st_respirator	mask	st_respirator_full, st_respirator_nose, st_respirator_mouth	Wear correctly; Wear uncorrectly; Not worn
respirator_color	mask color	color_type_none, color_type_other	without respirator; other color respirator
glass_style	glasses	glasses_style_type_none, transparent_glasses, sunglasses	without glasses; transparent glasses; sunglasses; ordinary glasses
gender_code	gender	female, male	female; male
st_expression	facial expression	st_angry, st_happy, st_sorrow, st_calm, st_surprised, st_scared, st_disgust, st_yawn	angry; happy; sorrow; calm; surprised; scared; disgust; yawn
cap_style	hat style	hat_style_type_none, cap	without hat; with hat
st_age_value	age value		

Appendix C: Body attribute description

Body Attribute	Description	Enum	Description
gender_code	gender	male, female	male; female
st_age	age grouping	st_old, st_adult, st_child	the eldly; adult; child
coat_length	coat style	st_bareback, short_sleeve, long_sleeve	bareback; short sleeve; long sleeve
pants_length_type	pants type	trousers, shorts, st_skirt	trousers; shorts; skirt
st_pedestrian_angle	orientation	st_front, st_side, st_back	positive; side; back
st_coat_pattern	coat pattern	st_pure, st_stripe, st_design, st_joint, st_lattic	solid color; stripe; pattern; splicing; lattice
trousers_color	trousers color	black, white, gray, red, yellow, blue, green, purple	black; white; gray; red; yellow; blue; green; purple
st_umbrella	umbrella	st_umbrella	
umbrella_color	umbrella color	black, white, gray, red, yellow, blue, green, purple,	black; white; gray; red; yellow; blue; green; purple;
		transparent	transparent

st_trousers_pattern	trousers pattern	st_pure, st_stripe, st_design, st_joint, st_lattic	solid color； stripe； pattern； splicing； lattice
st_hat	hat	st_hat	
cap_style	hat style	hat_style_type_none, bonnet, cap, bucket_hat, st_hard_hat	No hat； Beanie； Peaked cap； fisherman's hat； helmat
cap_color	hat color	black, white, gray, red, yellow, blue, green, purple	black； white； gray； red； yellow； blue； green； purple
coat_style	coat style	long_coat, jacket, t_shirt, sweater, shirt, dress, business_suit, coat_style_type_other	long coat； jacket； T-shirt； sweater； short sleeve； dress； suit； other
coat_color	coat color	black, white, gray, red, yellow, blue, green, purple	black； white； gray； red； yellow； blue； green； purple
hair_style	hair sytle	st_short, long, bald	short hair； long hair； bald
hair_color	hair color	black, white,	black； white；
		yellow	yellow

shoes_style	shoes style	leather_shoes, boots, walking_shoes, sandal	leather shoes; boots; casual shoes; sandals
shoes_color	shoes color	black, white, gray, red, yellow, blue, green, purple	black; white; gray; red; yellow; blue; green; purple
st_bag	bag	st_bag	
bag_type	bag type	hand_bag, shoulder_bag, backpack, trolley, bag_style_type_none, waist_pack	hand_bag; shoulder_bag; backpack; trolley; bag_style_type_none; waist_pack
bag_color	bag color	black, white, gray, red, yellow, blue, green, purple	black; white; gray; red; yellow; blue; green; purple
st_respirator	mask	st_respirator	st_respirator
respirator_color	mask color	black, white, gray, blue	black ; white ; gray; blue
glasses_style_type	glasses type	glasses_style_type_none, st_ordinary_glasses, sunglasses	No glasses; Ordinary glasses; Sunglasses
st_glove	glove	st_glove	
st_status_type	pedestrian status	st_normal, st_hold_object_in_front, st_play_phone, st_call, st_smoke	Walk normally; Hold something in front of your chest; play cell phone; Call;

			smokes
st_uniform	uniform	st_common_clothing, st_office_uniform, st_worker_uniform, st_chef_uniform, st_medical_uniform, st_police_uniform, st_firefighter_uniform, st_reflective_clothing	Ordinary clothing； Office uniform； Worker uniform； Chef uniform； Medical uniform； Police uniform； Firefighter uniform； Reflective vests

Appendix C1: Body alarmtype description

Eventtype Num	Alarmtype
0	no alarm
385	fight
386	tumble
387	smoke
388	doze
389	play phone
390	no helmet
391	no work cloth
394	invade
395	leave post
396	over boundary
398	crowded
641	fire smoke
705	fire extinguisher

Appendix D: Vehicle attribute description

Vehicle Attribute	Description	Enum	Description
vehicle_color	color of the vehicle	gray, white, red, black, blue, green, brown, yellow, purple,	gray, white, red, black, blue, green, brown, yellow, purple,
		pink	pink
vehicle_class	vehicle type	st_car, st_van, st_small_truck, st_big_truck, st_suv, st_big_bus, st_med_bus	car, truck, small truck, big truck, SUV, Bus, CMB

Appendix D1: Cyclist attribute description

Cyclist Attribute	Description	Enum	Description
vehicle_color	color of the vehicle	gray, white, red, black, blue, green, brown, yellow, purple, pink	gray, white, red, black, blue, green, brown, yellow, purple, pink
st_vehicle_class	vehicle type	st_motor, st_ebike, st_bicycle, st_tricycle	car, truck, small truck, big truck, SUV, Bus, CMB

Appendix F: Http/Websocket example of push receiving

1. httpserver Example

```
import tornado.ioloop
import tornado.web
import pprint
 import sys
 import os
 import time
 import json
 import logging
 import socket
 import datetime

 file_handler = logging.FileHandler('/tmp/http_server.log', 'a', encoding='utf-8')  formatter
 =  logging.Formatter("%(asctime)-15s  %(message)s")  file_handler.setFormatter(formatter)
 logger = logging.getLogger()
 logger.addHandler(file_handler)
 logger.setLevel(logging.INFO)

 # the ip of httpserver
 local_IP = socket.gethostbyname(socket.gethostname())  class

 MyDumpHandler(tornado.web.RequestHandler):

    def  post(self)   logger.info("="
       * 100 + "\n")
       logger.info(datetime.datetime.now())
       logger.info("\n[BODY]")
       try:
          body = self.request.body.decode('utf', 'backslashreplace')
          logger.info("try1:\n%s" % body)
          # logger.info('BODY: \n' + body)
       except BaseException as e:
          logger.info("BaseException1 %s" % e)  try:
             body = self.request.body.decode('utf-8')
             logger.info("try2:\n%s" % body)
             # logger.info('BODY: \n' + body)
          except BaseException as e:
             logger.info("BaseException2 %s" % e)
             pass

       os.makedirs('/tmp/images',    exist_ok=True)   ip
       = self.request.remote_ip  # the ip of nebula-m  file
       = self.request.files
       rsp_json  =  self.request.body_arguments["json"][0].decode("utf-8")
```

```
    # the push data field  js_dict
    = json.loads(rsp_json)  if
    js_dict["msg_id"] == 774:
      response_774_dict = ["camera_name", "device_id", "channel", "position", "img_id", "img_path",
"lib_name", "lib_type", "person_addr", "person_age", "person_gender",  "person_idcard","person_name",
"snap_id", "similarity", "quality", "snap_feat", "snap_path", "trigger"]
      self.assert_resp_in_data(response_774_dict, js_dict["data"])  elif
    js_dict["msg_id"] == 775:
      response_775_dict = ["device_id", "trigger"]
      self.assert_resp_in_data(response_775_dict, js_dict["data"])
    else:
      print(js_dict["msg_id"])

    # write the pushing picture  t
    = str(time.time()).replace('.', '')
    if file:
      logger.info(file['snap'][0]['filename']  +  '  saved')  fobj
      = open('/tmp/images/' + ip + 'snap' + t + '.jpg', "wb")
      fobj.write(file['snap'][0]['body'])
      fobj.close()

      logger.info(file['snap_frame'][0]['filename']  +  '  saved')  fobj
      = open('/tmp/images/' + ip + 'snap_frame' + t + '.jpg', "wb")
      fobj.write(file['snap_frame'][0]['body'])
      fobj.close()
    if 'img' in file:
      logger.info(file['img'][0]['filename']  +  '  saved')  fobj
      = open('/tmp/images/' + ip + 'image' + t + '.jpg', "wb")
      fobj.write(file['img'][0]['body'])
      fobj.close()

  @staticmethod
  def assert_resp_in_data(resp_dict, resp_data):  for
    item in resp_dict:
      assert item in resp_data, print("lost item %s" %item)

if __name__ == "__main__":
  try:
    print("\nHttp server listen on http://" + local_IP +  ":8880
      now ... (/tmp/http_server.log, /tmp/images)")
    tornado.web.Application([(r"/.*", MyDumpHandler), ]).listen(8080)
    tornado.ioloop.IOLoop.instance().start()
  except BaseException as e:
```

```
        print(e)
        pass
```

2. Websocket Example

```
#!/usr/bin/env python3
import       time   import
websocket  import
ssl   import   base64
import         sys   import
logging
IP = '10.5.2.81'   # the ip of nebula-m

key = ' '

file_handler =
   logging.FileHandler(  '/tmp/ws.log',
   'a', encoding='utf-8')
formatter = logging.Formatter("%(asctime)-15s %(message)s")
file_handler.setFormatter(formatter)
logger = logging.getLogger()
logger.addHandler(file_handler)
logger.setLevel(logging.INFO)

def on_message(ws, message):
   logger.info(message)      d
   = json.loads(message)
   print(d['msg'])
   if 'data' in d:
      # print(dir(d['data']))
      # print(pretty_json(d['data']['img']))
      t = str(time.time()).replace('.', '')
      # print(json.dumps(d))

      response_777_dict = ["camera_name", "device_id", "channel", "img_id", "img_path",  "lib_name","lib_type",
"person_addr", "person_age", "person_gender", "person_idcard",
       "person_name", "position", "ranking", "similarity", "quality", "snap_id", "snap_buf", "snap_feat",
"snap_path", "snap_frame", "trigger"]
      assert_resp_in_data(response_777_dict, d['data'])  if
      d['data']['img'] != -1:
         data = d['data']['img'].split(',', 1)
         f     =     imgdata     =     base64.b64decode(data[1])    fobj
         = open('/tmp/wsimages/' + IP + 'snap' + t + '.jpg', "wb")
         fobj.write(f)
```

```
        fobj.close()

def assert_resp_in_data(resp_dict, resp_data):  for
   item in resp_dict:
      assert item in resp_data, print(item)

 def on_error(ws, error):
   print(error)

def on_close(ws):
   print("### closed ###")

# Subscribe the push messages
def on_open(ws):
   def run(*args):
      ws.send('{"key":"%s", "msg_id":"776"}' % key)
      print("thread terminating...")
   thread.start_new_thread(run,  ())

if __name__ == "__main__":
   websocket.enableTrace(True)
   ws = websocket.WebSocketApp('ws://' + IP + '/ws/',
                           on_message=on_message,
                           on_error=on_error,  on_close=on_close,
                           on_open=on_open)
   ws.run_forever(sslopt={"cert_reqs":  ssl.CERT_NONE})
```

Version: 2024/05/20

NABLA WORKS

Business Inquiries
03 - 6721 - 8000
(Mon-Fri 09:30-18: 00)

Business Partnership
info@nablaworks.co.jp