

Content

1. Foreword.....	1
2. Introduction.....	1
3. Secret Key Query.....	1
4. Calling Conventions.....	1
4.1 Value of Digital Signature (Sign) in Client.....	1
4.2 Parameter Submitted by the Client to the Server.....	2
5. Time Format.....	2
6. Character Encoding.....	2
7. Corresponding Code of API Error.....	3
8. Instructions for API.....	4
8.1 API Related to Device.....	4
8.1.1 Remote Opening (/api/device/open).....	3
8.1.2 Time Synchronization (/api/device/syncTime).....	8
8.1.3 json Command (/api/device/jsonCmd).....	10
8.2 API Related to User.....	18
8.2.1 User Object (UserObject).....	18
8.2.2 Add User (/api/user).....	19
8.2.3 Update User (/api/uer/update/{userId}).....	20
8.2.4 Delete User (/api/user/delete/{userId}).....	21
8.2.5 Query User (/api/user/list).....	22
8.3 API Related to Record.....	24
8.3.1 Recognition Record Object (SignObject).....	24
8.3.2 Recognition Record Acquisition (/api/record).....	25
8.3.3 Recognition Record Push (/api/record).....	27
8.3.4. Recognition Record POST (HTTP).....	29

1. Foreword

This document is used to providing OpenAPI call guidelines for face recognition device for developers.

2. Introduction

The open platform is an open service platform based on face recognition device. The multi-scene services with interface capabilities such as user and record can be obtained after the third-party system is connected to the face recognition device.

No domain name or IP is restricted to access to the OpenAPI in face recognition device, and all IP requests that can access to this device are considered legitimate.

3. Secret Key Query

Log in face recognition device to query the login password, which defaults to 123456.

Note: Host in this document is device IP address, all the Port value is 8899.

4. Calling Conventions

4.1 Value of Digital Signature (Sign) in Client

Calculation method: $\text{sign} = \text{md5}(\text{timestamp}\#\text{password})$, e.g.:

Password = 123456

timestamp = 1573539353795 (timestamp of GMT (ms))

$\text{sign} = \text{md5}(1573539353795\#123456) =$

188ef735093645c802b0e4741bd3e9a1

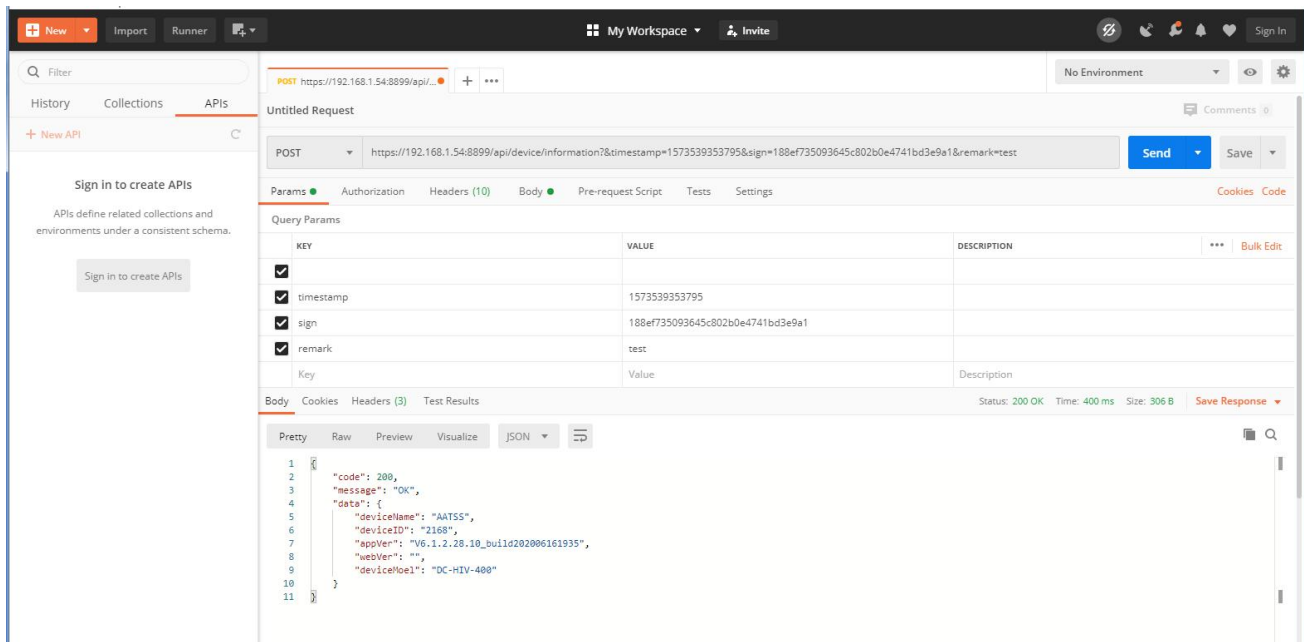
4.2 Parameter Submitted by the Client to the Server

Submit the two parameters of timestamp and sign to the server together with the BizParam, e.g.:

`http://host:port/api/device/open?timestamp=1573539353795&sign=188ef735093645c802b0e4741bd3e9a1&remark=test`

Thereinto “remark” indicates BizParam. The server compares the current timestamp and the timestamp in client request, it will be deemed as a fillback if the difference is **more than 5 minutes**.

POSTMAN Screenshot:



5. Time Format

The time format is the timestamp of GMT (s), unless otherwise specified.

6. Character Encoding

UTF-8 character is adopted for all character encoding, unless otherwise specified.

7. Corresponding Code of API Error

Code	Message	Reason
10001	Password Not Exist	Password Not Exist
10002	Password Expired	Password Expired
10003	Request Expired	Request Expired
10004	IP Invalid	IP Invalid
10005	Signature Failed	Signature Failed
20001	User Not Exists	User Not Exists
20002	User Not Match	User Not Match
30001	Param Invalid	Param Invalid
30002	No Face Detected In The Image	No Face Detected In The Image
30003	Missing Param	Missing Param
30004	Avatar File Illegal	Avatar File Illegal
40001	Not Support	Not Support
50001	RPC Failed	RPC Failed, Timestamp Illegal
50002	Internal Server Error	Internal Server Error
50003	Multi Request Error	Multi Request Error
50004	Request Method Not Allowed	Request Method Not Allowed
99999	Unknown Error	Unknown Error
50008	Search List Null	Search List Null
50009	Database Full	Database Full

8. Instructions for API

8.1 API Related to Device

8.1.1 Device Information Object (DeviceObject)

Attribute Name	Type	Required or Not	Instructions
deviceID	string	Y	Unique deviceID, with maximum length of 32
deviceName	string	Y	Device name, with maximum length of 32
appVer	string	Y	App version of device, with maximum length of 32
webVer	string	Y	Web version, with maximum length of 32
deviceMoel	string	Y	Device model, with maximum length of 32

8.1.2 Network Parameter Object (NetworkObject)

Attribute Name	Type	Required or Not	Instructions
dhcpEnable	integer	Y	DHCP enable switch; 0 indicates closed, and 1 indicates open
ipAddress	string	Y	IP address of device
subMask	string	Y	Subnet mask of device
DNS1	string	Y	Preferred DNS server
DNS2	string	Y	Alternate DNS server
defRout	string	Y	Gateway

8.1.3 Audio Parameter Object (AudioObject)

Attribute Name	Type	Required or Not	Instructions
aencEnable	integer	Y	Audio enable switch, 0 indicates closed, 1 indicates open
aencType	integer	Y	Type of audio encoding:

			G726 = 1, G722 = 2, G711A =3, ADPCM = 4, MP3 = 5, PCM = 6, G711U = 7, AACLC =8,
sampleRate	integer	Y	Audio sampling rate of device
aencbitrate	integer	Y	Audio code rate of device
micLine	integer	Y	Microphone input 0; microphone input 1
volumeIn	integer	Y	Input volume of device: 0 to 15
volumeOut	integer	Y	Output volume of device: 0 to 15

8.1.4 Smart Face Parameter Object (SmartFaceObject)

Attribute Name	Type	Required or Not	Instructions
faceEnable	integer	Y	Enable switch of face algorithm; 0 indicates closed, and 1 indicates open
faceCmpEnable	integer	Y	Enable switch of face comparison; 0 indicates closed, and 1 indicates open
faceThre	integer	Y	Detection threshold: 0 to 10
faceMinsize	integer	Y	Minimize pixel of face capture: 30 to 300
faceMaxsize	integer	Y	Maximum pixel of face capture: 300 to 500
faceInterval	integer	Y	Detection interval: 6s to 20s
imgType	integer	Y	Image type, 0 indicates face; 1 indicates original image, and 2 indicates face and original image
maskDetecEnable	integer	Y	Enable switch of mask detection; 0 indicates closed, and 1 indicates open
tempDetecEnable	integer	Y	Enable switch of temperature detection; 0 indicates closed, and 1 indicates open

TempUnit	integer	Y	Unit of temperature; 0 indicates degree centigrade, and 1 indicates degree Fahrenheit
tempWarnThre	double	Y	Warning threshold of temperature: 37.0 to 37.5
ftpPutImgEnable	integer	Y	Enable switch of FTP image put; 0 indicates closed, and 1 indicates open
faceFrame	integer	Y	Enable switch of face frame overlay detection; 0 indicates closed, and 1 indicates open
StrTempCkEnable	integer	Y	Enable switch of strange face check; 0 indicates closed, and 1 indicates open
Hacknessenble	integer	Y	Enable switch of bio-assay

8.1.5 Peripheral Parameter Object (PerParaObject)

Attribute Name	Type	Required or Not	Instructions
detecTemMod	integer	Y	Temperature detection model
alarmType	integer	N	Alarm model; 0 indicates normal alarm, and 1 indicates abnormal alarm
voiceProswitch	integer	Y	Voice prompt switch; 0 indicates closed, and 1 indicates open
tempNormal	integer	Y	Normal temperature or not
maskNormal	integer	Y	Wearing mask normally or not
cmpNormal	integer	Y	Normal comparison or not
RecordOpen	integer	Y	Opening of open record; 0 indicates closed, and 1 indicates open
unlockControl	integer	Y	Unlock control; 0 indicates normally open, and 1 indicates normally closed
unlockOuttime	integer	Y	Unlock output time (1-254)*30ms

8.1.6 Production Configuration Parameter Object (ProductionObject)

Attribute Name	Type	Required or Not	Instructions
deviceType	integer	Y	Device type

devSerialUUID	string	N	Device serial No., with maximum length of 32
deviceName	string	Y	Device name, with maximum length of 16
sensorType	integer	Y	Sensor type
screenSizeType	integer	Y	Screen size type
deviceCpuType	integer	Y	Device CPU type
wifiFlage	integer	Y	Support WFIF or not; 0 indicates no, and 1 indicates yes
wifiDefaultMode	integer	Y	WIFI type 0 staute 1AP
defaultIPAddr	integer	Y	Default IP
languageType	integer	Y	Language type; 0 indicates Chinese, and 1 indicates English
GB28181Flage	integer	Y	Support national standard or not; 0 indicates no, and 1 indicates yes
P2PType	integer	Y	Support mobile monitoring or not; 0 indicates no, and 1 indicates yes // 1G

8.1.7 Remote Opening (/api/device/open)

1. Functions Description

Give opening command to the device

2. Request Address Example

<https://HOST:PORT/api/device/open>

3. Sample

<https://192.168.1.100:8899/api/device/open?timestamp=1593324328000&sign=8793a72812e36c5e974e14b5fddecf35>

4. Request Method

POST: form-data

5. Request Parameter

Parameter Name	Type	Required or Not	Instructions
remark	string	N	Remarks of opening with maximum length of 255; written in url
timestamp	string	Y	Current timestamp (ms) (ms), which is written in url
sign	string	Y	Parameter signature, which is written in url; refer to document for calculation rules

6. Normal Return Response

```
{
  "code": 200,
  "message": "OK",
  "desc": ""
}
```

7. Abnormal Return Response

Code	Message	Desc	Error Description
40001	Not Support	device-not- support	Device-not-support

8.1.8 Time Synchronization (/api/device/syncTime)

1. Functions Description

Synchronize the time command to the device

2. Request Address Example

<https://HOST:PORT/api/device/syncTime>

3. Sample

https://192.168.1.100:8899/api/device/syncTime?timestamp=1593327458000&sign=3314f754d6d1618a37bbd9444e7e7294

4. Request Method

POST: form-data

5. Request Parameter

Parameter Name	Type	Required or Not	Instructions
timestamp	string	Y	Current timestamp (ms) (ms), which is written in url; synchronize the time to the device
sign	string	Y	Parameter signature, which is written in url; refer to document for calculation rules

6. Normal Return Response

```
{
  "code": 200, "message":
  "OK",
  "desc": ""
}
```

7. Abnormal Return Response

Code	Message	Desc	Error Description
40001	Not Support	device-not- support	Device-not-support

8.1.9 Remote Device Reboot (/api/device/reboot)

8. Functions Description

Give opening command to device

9. Request Address Example

<https://HOST:PORT/api/device/reboot>

10. Sample

https://192.168.1.100:8899/api/device/reboot?timestamp=1593327458000&sign=3314f754d6d1618a37bbd9444e7e7294

11. Request Method

POST: form-data

12. Request Parameter

Parameter Name	Type	Required or Not	Instructions
timestamp	string	Y	Current timestamp (ms) (ms), which is written in url
sign	string	Y	Parameter signature, which is written in url; refer to document for calculation rules

13. Normal Return Response

```
{
  "code": 200,
  "message": "OK",
  "desc": ""
}
```

14. Abnormal Return Response

Code	Message	Desc	Error Description
40001	Not Support	device-not- support	Device-not-support

8.1.10 json Command (/api/device/jsonCmd)

1. Functions Description

Give json command to the device

2. Request Address Example

https://HOST:PORT/api/device/jsonCmd

3. Request Method

POST: form-data

4. Request Parameter

Parameter Name	Type	Required or Not	Instructions
json object	json	Y	Written in Content
timestamp	string	Y	Current timestamp (ms) (ms), which is written in url; synchronize the time to the device
sign	string	Y	Parameter signature, which is written in url; refer to document for calculation rules

5. Normal Return Response

```
{
  "code": 200, "message":
  "OK",
  "desc": "",
  "json": {} // detailed result
}
```

6. Abnormal Return Response

Code	Message	Desc	Error Description
40001	Not Support	device-not- support	Device-not-support

8.1.11 Device Information Acquisition (/api/device/information)

7. Functions Description

Acquire device information from the device

8. Request Address Example

<https://HOST:PORT/api/device/information>

9. Sample

https://192.168.1.100:8899/api/device/information?timestamp=1593324328000&sign=8793a72812e36c5e974e14b5fddecf35

10. Request Method

POST: form-data

11. Request Parameter

Parameter Name	Type	Required or Not	Instructions
timestamp	string	Y	Current timestamp (ms) (ms), which is written in url; synchronize the time to the device
sign	string	Y	Parameter signature, which is written in url; refer to document for calculation rules

12. Normal Return Response

```
{
  "code": 200,
  "message": "OK",
  "desc": "",
  "data": DeviceObject
}
```

13. Abnormal Return Response

Code	Message	Desc	Error Description
40001	Not Support	device-not- support	Device-not-support

8.1.12 Network Parameter Acquisition (/api/device/network)

14. Functions Description

Acquire network parameter information from the device

15. Request Address Example

<https://HOST:PORT/api/device/network>

16. Sample

https://192.168.1.100:8899/api/device/network?timestamp=1593324328000&sign=8793a72812e36c5e974e14b5fddecf35

17. Request Method

POST: form-data

18. Request Parameter

Parameter Name	Type	Required or Not	Instructions
timestamp	string	Y	Current timestamp (ms) (ms), which is written in url; synchronize the time to the device
sign	string	Y	Parameter signature, which is written in url; refer to document for calculation rules

19. Normal Return Response

```
{  
  "code": 200,  
  "message": "OK",  
  "desc": "",  
  "data": NetworkObject
```

}

20. Abnormal Return Response

Code	Message	Desc	Error Description
40001	Not Support	device-not- support	Device-not-support

8.1.13 Network Parameter Setting (/api/device/networkcfg)

21. Functions Description

Set network parameter information for the device

22. Request Address Example

<https://HOST:PORT/api/device/networkcfg>

23. Sample

https://192.168.1.100:8899/api/device/networkcfg?timestamp=1593324328000&sign=8793a72812e36c5e974e14b5fddecf35

```
{
  "dhcpEnable": 0,
  "ipAddress": "192.168.1.73",
  "subMask": "255.255.0.0",
  "defRout": "192.168.0.1",
  "DNS1": "202.96.134.133",
  "DNS2": "8.8.8.8"
}
```

(Note: {}Data should be written in Content)

24. Request Method

POST: form-data

25. Request Parameter

Parameter Name	Type	Required or Not	Instructions
----------------	------	-----------------	--------------

json object	<u>NetworkObject</u>	Y	Written in Content
timestamp	string	Y	Current timestamp (ms) (ms), which is written in url; synchronize the time to the device
sign	string	Y	Parameter signature, which is written in url; refer to document for calculation rules

26. Normal Return Response

```
{
  "code": 200,
  "message": "OK",
  "desc": ""
}
```

27. Abnormal Return Response

Code	Message	Desc	Error Description
40001	Not Support	device-not- support	Device-not-support

8.1.14 Smart Face Parameter Acquisition (/api/device/smartface)

28. Functions Description

Acquire smart face parameter information from the device

29. Request Address Example:

<https://HOST:PORT/api/device/smartface>

30. Sample

<https://192.168.1.100:8899/api/device/smartface?timestamp=1593324328000&sign=8793a72812e36c5e974e14b5fddecf35>

31. Request Method

POST: form-data

32. Request Parameter

Parameter Name	Type	Required or Not	Instructions
timestamp	string	Y	Current timestamp (ms) (ms), which is written in url; synchronize the time to the device
sign	string	Y	Parameter signature, which is written in url; refer to document for calculation rules

33. Normal Return Response

```
{
  "code": 200,
  "message": "OK",
  "desc": "",
  "data": SmartFaceObject
}
```

34. Abnormal Return Response

Code	Message	Desc	Error Description
40001	Not Support	device-not- support	Device-not-support

8.1.15 Smart Face Parameter Setting (/api/device/smartfacecfg)

35. Functions Description

Set smart face parameter information for the device

36. Request Address Example

<https://HOST:PORT/api/device/smartfacecfg>

37. Sample

<https://192.168.1.100:8899/api/device/smartfacecfg?timestamp=1593324328000&sign=8793a72812e36c5e974e14b5fddecf35>

```
{
  "faceEnable": 1,
  "faceCmpEnable": 1,
  "faceThre": 99,
  "faceMinsize": 300,
  "faceMaxsize": 500,
  "faceInterval": 4,
  "imgType": 0,
  "maskDetecEnable": 0,
  "tempDetecEnable": 0,
  "tempWarnThre": 37.500000,
  "ftpPutImgEnable": 1,
  "faceFrame": 1,
  "Hacknessenble": 1,
  "TempUnit": 0,
  "StrTempCkEnable": 1
}
```

(Note: {} Data should be written in Content)

38. Request Method

POST: form-data

39. Request Parameter

Parameter Name	Type	Required or Not	Instructions
json object	<u>SmartFaceObject</u>	Y	Written in Content
timestamp	string	Y	Current timestamp (ms) (ms), which is written in url; synchronize the time to the device
sign	string	Y	Parameter signature, which is written in url; refer to document for calculation rules

40. Normal Return Response

```
{  
  "code": 200,  
  "message": "OK",  
  "desc": ""  
}
```

41. Abnormal Return Response

Code	Message	Desc	Error Description
40001	Not Support	device-not- support	Device-not-support

8.1.16 Peripheral Parameter Acquisition (/api/device/perpara)

42. Functions Description

Acquire peripheral parameter information from the device

43. Request Address Example

<https://HOST:PORT/api/device/perpara>

44. Sample

<https://192.168.1.100:8899/api/device/perpara?timestamp=1593324328000&sign=8793a72812e36c5e974e14b5fddecf35>

45. Request Method

POST: form-data

46. Request Parameter

Parameter Name	Type	Required or Not	Instructions
timestamp	string	Y	Current timestamp (ms) (ms), which is written in url; synchronize the time to the device
sign	string	Y	Parameter signature, which is written in url; refer to document for calculation rules

47. Normal Return Response

```
{
  "code": 200,
  "message": "OK",
  "desc": "",
  "data": PerParaObject
}
```

48. Abnormal Return Response

Code	Message	Desc	Error Description
40001	Not Support	device-not- support	Device-not-support

8.1.17 Peripheral Parameter Setting (/api/device/perparacfg)

49. Functions Description

Set peripheral parameter information for the device

50. Request Address Example

<https://HOST:PORT/api/device/perparacfg>

51. Sample

<https://192.168.1.100:8899/api/device/perparacfg?timestamp=1593324328000&sign=8793a7281>

2e36c5e974e14b5fddecf35

52. Request Method

POST: form-data

53. Request Parameter

Parameter Name	Type	Required or Not	Instructions
json object	<u>PerParaObject</u>	Y	Written in Content
timestamp	string	Y	Current timestamp (ms) (ms), which is written in url; synchronize the time to the device
sign	string	Y	Parameter signature, which is written in url; refer to document for calculation rules

54. Normal Return Response

```
{
  "code": 200, "message":
  "OK",
  "desc": ""
}
```

55. Abnormal Return Response

Code	Message	Desc	Error Description
40001	Not Support	device-not- support	Device-not-support

8.2 API Related to User

8.2.1 User Object (UserObject)

Json object:

Attribute Name	Type	Required or Not	Instructions
userId	string	Y	User ID with maximum length of 16
userType	integer	Y	User type (1 indicates staff, 2 indicates visitor, 3 indicates blacklist, and -1 indicates stranger)
photo	string	N	Base64 encoding of base photo, with required content type of image or jpeg
icNumber	string	Y	IC number, with maximum length of 40, and blank string is available
jobNumber	string	Y	Job number, with maximum length of 40, and blank string is available
mobile	string	Y	Mobile phone number, with maximum length of 20, and blank string is available
name	string	Y	Name with maximum length of 30
remark	string	Y	Remark, with maximum length of 255, and blank string is available
department	string	N	Department, with maximum length of 80
birthday	integer	N	Birthday
entryTime	integer	N	Entry time
mail	string	N	Email, with maximum length of 40
title	string	N	Title, with maximum length of 40
position	string	N	Work position, with maximum length of 100
idNumber	string	N	ID number, with maximum length of 40
gender	integer	N	Gender; 1 indicates male, and 2 indicates female
prompt	string	N	Customized welcome words can be displayed in device side, with maximum length of 100

startTime	integer	N	Start time
endTime	integer	N	End time
purpose	string	N	Visiting purpose, with maximum length of 40
interviewee	string	N	Interviewee, with maximum length of 80
comeFrom	string	N	Visiting unit, with maximum length of 200
description	string	N	Description, with maximum length of 80

8.2.2 Add User (/api/user)

1. Functions Description

Add staff.

2. Request Address Example

<https://HOST:PORT/api/user>

3. Sample

<https://192.168.1.100:8899/api/user?timestamp=1593324328000&sign=8793a72812e36c5e974e14b5fddecf35>

```
{
  "birthday": 0,
  "comeFrom": "",
  "department": "",
  "description": "",
  "endTime": 0,
  "entryTime": 0,
  "gender": 0,
  "icNumber": "",
  "idNumber": "",
  "interviewee": "",
  "jobNumber": "",
  "mail": "",
  "mobile": "1",
```

```

    "name": "A001",
    "photo": "/9j/4AAxxxxxx(JPG format picture with base64 encoding)",
    "position": "1",
    "purpose": "",
    "remark": "",
    "startTime": 0,
    "title": "",
    "userId": "0011",
    "userType": 1
}

```

(Note: {}Data should be written in Content)

4. Request Method

POST: application/json

5. Request Parameter

Parameter Name	Type	Required or Not	Instructions
json object	UserObject	Y	Written in Content
timestamp	string	Y	Current timestamp (ms) (ms), which is written in url
sign	string	Y	Parameter signature, which is written in url; refer to document for calculation rules

6. Normal Return Response

```

{
  "code": 200,
  "message": "OK",
  "desc": "",
  "data": UserObject
}

```


}

8.2.3 Update User (/api/user/update/{userId})

1. Functions Description

Update the staff information.

2. Request Address Example

<https://HOST:PORT/api/user/update/{userId}>

3. Sample

https://192.168.1.100:8899/api/user/update/0011?timestamp=1593324328000&sign=8793a72812e36c5e974e14b5fddecf35

```
{
  "birthday": 0,
  "comeFrom": "",
  "department": "",
  "description": "",
  "endTime": 0,
  "entryTime": 0,
  "gender": 0,
  "icNumber": "",
  "idNumber": "",
  "interviewee": "",
  "jobNumber": "",
  "mail": "",
  "mobile": "1",
  "name": "A001",
  "photo": "/9j/4AAxxxxx(JPG format picture with base64 encoding)",
  "position": "1",
  "purpose": "",
  "remark": "",
  "startTime": 0,
  "title": "",
  "userId": "0011",
  "userType": 1
}
```

(Note: {}Data should be written in Content)

4. Request Method

POST: application/json

5. Request Parameter

Parameter Name	Type	Required or Not	Instructions
json object	UserObject	Y	Written in Content
timestamp	string	Y	Current timestamp (ms) (ms), which is written in url
sign	string	Y	Parameter signature, which is written in url; refer to document for calculation rules

6. Normal Return Response

```
{
  "code": 200,
  "message": "OK",
  "desc": "",
  "data": UserObject
}
```

8.2.4 Delete User (/api/user/delete/{userId})

1. Functions Description

Delete user.

2. Request Address Example

<https://HOST:PORT/api/user/delete/{userId}>

3. Sample

https://192.168.1.100:8899/api/user/delete/0011?timestamp=1593324328000&sign=8793a72812e36c5e974e14b5fddecf35

4. Request Method

GET

Parameter Name	Type	Required or Not	Instructions
id	string	Y	User ID, which is written in url; “id = all” indicates clear all user information
timestamp	string	Y	Current timestamp (ms) (ms), which is written in url
sign	string	Y	Parameter signature, which is written in url; refer to document for calculation rules

5. Normal Return Response

```
{
  "code": 200,
  "message": "OK",
  "desc": "",
  "data": UserObject
}
```

6. Instructions to Return Fields

Parameter Name	Type	Instructions
code	integer	Return code

message	string	Return information
----------------	--------	--------------------

8.2.5 Query User (/api/user/list)

1. Functions Description

Acquire all user information of staff, visitor, and black list. Filter search with specific fields is supported.

2. Request Address Example

<https://HOST:PORT/api/user/list>

3. Sample

<https://192.168.1.100:8899/api/user/list?type=1&page=1&size=10×tamp=1593324328000&sign=8793a72812e36c5e974e14b5fddecf35>

4. Request Method

GET

Parameter Name	Type	Required or Not	Instructions
userId	string	N	User ID, which is written in url
type	integer	Y	User type, which is written in url; 1 indicates staff, 2 indicates visitor, 3 indicates blacklist, -1 indicates stranger, and return to all users in case of no parameter passed
name	string	N	User name, which is written in url
jobNumber	string	N	Job number, which is written in url
mobile	string	N	Mobile phone number, which is written in url
mail	string	N	Email, which is written in url
icNumber	string	N	IC number, which is written in url

idNumber	string	N	ID number, which is written in url
department	string	N	Department, which is written in url
page	integer	Y	Page number (default to 1), which is written in url
size	integer	Y	Data number (default to 10) on each page, which is written in url, with maximum pieces of 10; 10 indicates 10 pieces or more.
timestamp	string	Y	Current timestamp (ms) (ms), which is written in url
sign	string	Y	Parameter signature, which is written in url; refer to document for calculation rules

5. Normal Return Response

```
{
  "code": 200,
  "message": "OK",
  "desc": "",
  "data": {
    "total": 14, // total number of records
    "pageNum": 1, // page No.
    "size": 3, // current number of records
    "pageSize": 3, // size of each page
    "totalPage": 5, // total pages
    "data": [
      UserObject,
      UserObject,

```

```

        UserObject
    ]
}
}

```

8.3 API Related to Record

8.3.1 Recognition Record Object (SignObject)

Json object. Add the following recognition record attributes to the user object (UserObject):

Attribute Name	Type	Required or Not	Instructions
id	integer	Y	ID of recognition record
signAvatar	string	N	Base64 encoding of captured recognition photo
signTime	integer	Y	Recognition time in timestamp format
type	integer	Y	User type (1 indicates staff, 2 indicates visitor, 3 indicates blacklist, and -1 indicates stranger)
inputType	integer	Y	Recognition method (1 indicates face recognition, 2 indicates QR code recognition, 3 indicates card recognition, and 4 indicates UUID)
similarity	integer	N	Similarity with the scope from 0 to 100
temperature	float	N	Temperature
temperatureState	integer	N	Temperature status (0 indicates normal, -1 indicates hypothermia, and 1 indicates hyperthermia)
wearMask	boolean	N	Wearing mask or not
wearMaskAlarm	boolean	N	Alarm for non-wearing of mask

deviceMark	string	N	Device mark
Temperatureunit	integer	N	0 indicates degree centigrade, and 1 indicates degree Fahrenheit

When a stranger is recognized, the userType of UserObject shows -1.

8.3.2 Recognition Record Acquisition (/api/record)

1. Functions Description

Acquire the list of recognition records in reverse chronological order by default

2. Request Address Example

<https://HOST:PORT/api/record>

3. Sample

<https://192.168.1.100:8899/api/record?type=0&startTime=1593273600&endTime=1593313901&page=1&size=6×tamp=1593333615000&sign=a3cf7f874a51765f9fdabd16d201173f>

4. Request Method

GET

5. Request Parameter

Parameter Name	Type	Required or Not	Instructions
type	integer	Y	User type (1 indicates staff, 2 indicates visitor, 3 indicates blacklist, and -1 indicates stranger), which is written in url, and 0 indicates returning to all User
userId	string	N	User ID, which defaults to 0 and is written in url
startTime	integer	Y	Start time with the timestamp that is accurate to second, which defaults to current day and is written in url
endTime	integer	Y	End time with the timestamp that is accurate to second, which defaults to current day and

			is written in url
username	string	N	User name, which defaults to blank string, is written in url
page	integer	Y	Page number (defaults to 1), which is written in url
size	integer	Y	Data number (default to 20) on each page, which is written in url, with maximum pieces of 1,000; 1,000 indicates 1,000 pieces or more
timestamp	string	Y	Current timestamp (ms) (ms), which is written in url
sign	string	Y	Parameter signature, which is written in url; refer to document for calculation rules

6. Normal Return Response

```
{
  "code": 200,
  "message": "OK",
  "desc": "",
  "data": {
    "total": 14, // total number of records
    "pageNum": 1, // page No.
    "size": 3, // current number of records
    "pageSize": 3, // size of each page
    "totalPage": 5, // total pages
    "data": [
      SignObject,
```



```

        SignObject,
        SignObject
    ]
}
}

```

8.3.3 Recognition Record Push (/api/record)

1. Functions Description

Receive the real-time recognition record push (WebSocket) from the device, with service client numbered 9000. The client sends Heat Beat every ten seconds and the chain shall be considered broken if there is no any information interaction within 33 seconds.

2. Request Address Example

ws://HOST:9000/api/record?timestamp={timestamp}&sign={sign}

3. Request Method

GET

4. Request Parameter

Parameter Name	Type	Required or Not	Instructions
timestamp	string	Y	Current timestamp (ms) (ms), which is written in url
sign	string	Y	Parameter signature, which is written in url; refer to document for calculation rules

5. Normal Return Response

Successful access:

```
{
```

```
    "code": 20000, "message":  
    "Connected!",  
    "desc": "Connected!"  
}
```

Failed access:

```
{  
    "code": 20002,  
    "message": "Permission Denied!",  
    "desc": "app key not exist"  
}
```

Push-body of Information:

```
{  
    "code": 30000,  
    "message": "push record",  
    "desc": "",  
    "data": SignObject  
}
```

9. Help and Support

Please feel free to contact us in case of any doubt when using this product:

www.richtech-ai.com

API FOR AATSS MODEL V2 PRO

service@richtech-ai.com