HTTP Protocol For Face Recognition All-in-One Machine
V2.0.3

1、Documentation

This document is mainly used to interface with the HTTP method of the face recognition all-in-one machine, and is suitable for system developers and maintainers.

2、Interface Description

This data interface specification uses the standard HTTP protocol for communication with the following requirements:

2.1 Request Method

The request method uses the `POST` method, and the download and upload data are placed in the message body (entity-body)。

2.2 Data Types And Formats

1. All strings should be encoded in the `UTF-8` character set.
2. The data exchanged between the two parties is uniformly packaged in `JSON` format, with the following basic format:

```
{
    "operator": "<Operator>",
    "info": {<Info>},
    ……
}
```

See the interface-specific test cases below for details.

3. All image data is encoded using `base64` encoding or image `URI`.
4. All HTTP interactions must contain `Content-Length` or `Transfer-Encoding` in the header, noting that it is case-sensitive.

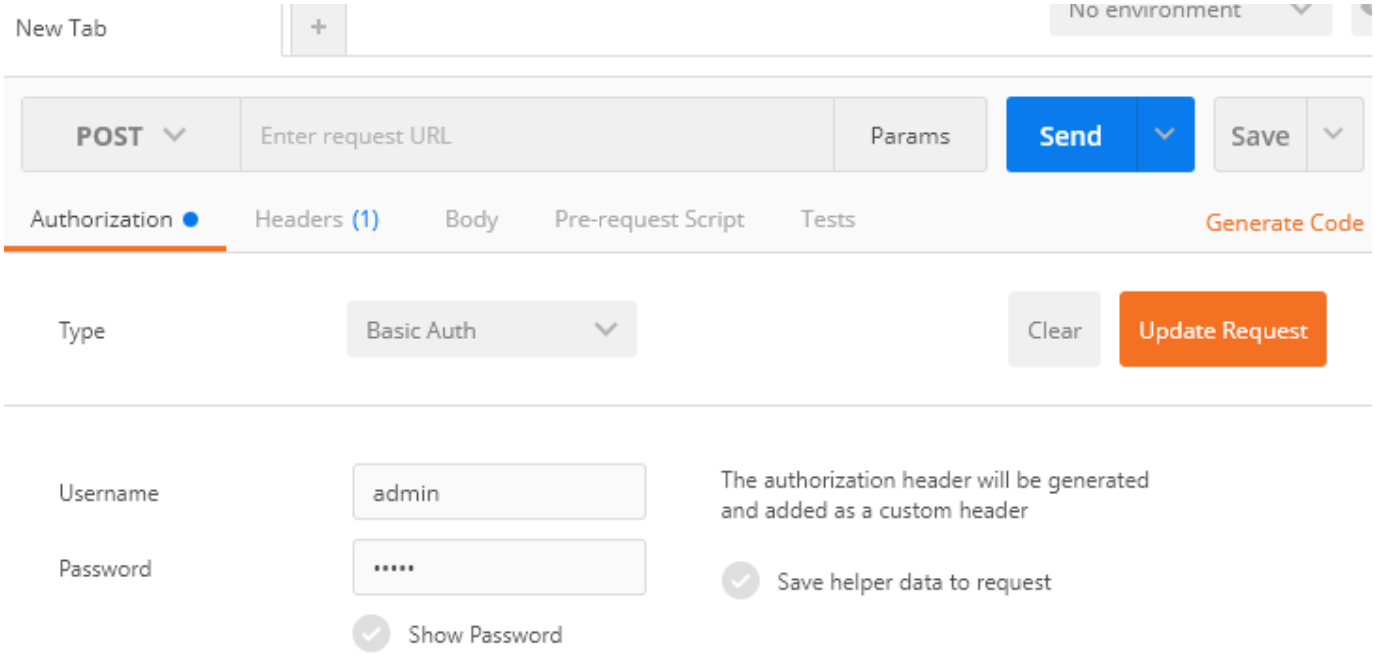
2.3 Grammatical

The format of the requested `URI`:`http://<server ipaddr>/action/<Operator>`。 Where`<server ipaddr>`:the ip address of the requested Face Recognition All-in-One Machine；`<Operator>`:the content of the action. See the interface-specific test cases below for details.

2.4 Authenticate

In order to ensure the security of data transmission, authentication is performed during formal data transmission, and the `basic` method is used for the time being. Typical data of `Authorization header` is `"Authorization: Basic jdhaHY0="`, where `Basic` means basic authentication, and `jdhaHY0=` is the base64-encoded `user:passwd` string. For example, if the username of the Face Recognition All-in-One Machine is "admin" and the password is 'admin', then the base64 encoding of "admin:admin" in the header of the sent data is "admin:admin", and the base64 encoding of "admin:admin" is "admin:admin", and the base64 encoding is "admin:admin". The base64 encoding of "admin:admin" in the header of the sent data is "YWRtaW46YWRtaW4=", and the ID authentication of the sent data is as follows: "YWRtaW46YWRtaW4=".

If you are using `postman` as an example to debug the interface, select `Basic Auth` under `Authorization`, enter your username and password, click `Update Request`, and `postman` will encrypt your username and password and add them to the Headers, as shown in the figure:



After setting up, select `POST` request method, enter the corresponding function interface URL, and click `Send`.

3、System Interfaces

3.1 Personnel Management

Personnel management mainly involves the management of personnel interactions between the platform and the Face Recognition All-in-One Machine.

3.1.1 List Editing and Modification

1. Description

   Adding and modifying individual list, the parameters to be modified are optional, if they do not need to be modified, the corresponding field values should not appear in the data sent. Add or modify the list according to `CustomizeID` or `PersonUUID`, if `CustomizeID` or `PersonUUID` already exists, then modify the list, if it doesn't exist, then add the list which corresponds to `CustomizeID` or `PersonUUID`; this interface can only choose one or the other of `CustomizeID` or `PersonUUID`. `CustomizeID` or `PersonUUID` can only choose one or the other. The image of the corresponding issued list of people can be issued with a field value of `picinfo` filled with base64 image data, or with a field value of `picURI` filled with the URI address of the image, and the corresponding image data will be obtained by the Face Recognition All-in-One Machine. **These two ways are optional**, when `picinfo` and `picURI` are passed in at the same time, the default is to get the data corresponding to the `picinfo`, **If it is to modify the list of people, it is not involved. If you are modifying the list of people, and there is no image replacement involved, you can not pass in the field values `picURI` and `picinfo`.**

2. API Description

Items	Description
Operator	`EditPersonNew`
Request URL	`http://<server ipaddr>/action/EditPersonNew` (Where `<server ipaddr>` is the device IP，for example:192.168.1.10)

3. Explanation of the Paragraph of the Requested Message

   Parameter information(Note: `Optional` is optional):

Key	Type	Values	Description
operator	string	EditPersonNew	Individual personnel additions or modifications
info	json object		Concrete content
DeviceID	int		ID number of the device to be operated See 6.8.1 Getting Device Information
IdType	int	0/2	Type of ID to be operated 0:CustomizeID 2:PersonUUID
CustomizeID	unsigned int **Must be filled in when IdType=0**		User-defined ID You can manage the corresponding personnel information such as modification and deletion by this ID number. Note: You can not use the same ID to add people repeatedly, otherwise it will lead to errors in modifying and deleting operations.
PersonUUID	string **Must be filled in when IdType=2**		User-defined UUID This UUID number can be used to manage personnel information such as modification and deletion. Note: You can not use the same ID to add people repeatedly, otherwise it will lead to errors in operations such as modification and deletion (not more than 48 bytes)
PersonType	int (optional)	0~1	Type of list 0: white list 1: black list **Default 0(white list): when adding people**
Name	string (optional)		Name Maximum length is 32 characters (including terminator) **Default “” (empty):when adding a new person**

Key	Type	Values	Description
Gender	int (optional)	0~1	Genders 0: male 1: female **Default 0(male): when adding people**
Nation	int (optional)	Reserve	Reserve
CardType	int (optional)		ID Type 0:ID **Default 0(ID): when adding people**
IdCard	string (optional)		ID Number Maximum length is 32 characters (including terminator) **Default “” (empty):when adding a new person**
Birthday	string (optional)	YYYY-MM-DD	Birthday e.g., 2018-1-1 **Default 1970-00-00:when adding a new person**
Telnum	string (optional)		Telephone Number Maximum length is 32 characters (including terminator) **Default “” (empty):when adding a new person**
Native	string (optional)		Native Maximum length is 32 characters (including terminator) **Default “” (empty):when adding a new person**
Address	string (optional)		Address Maximum length is 72 characters (including terminator) **Default “” (empty):when adding a new person**
Notes	string (optional)		Notes Maximum length is 64 characters (including terminator) **Default “” (empty):when adding a new person**

Key	Type	Values	Description
MjCardFrom	int (optional)	0~3	Wiegand Card Number Generation Method 0: public number 1: automatic generation 2: manual input 3: do not use access card numbers
WiegandType	int (optional)	0~1 or 6~7	The Wiegand protocols that form the basis for the Wiegand card numbers **Must be filled in when MjCardFrom=2** ; 0: 26 bits 1: 34 bits 6:26 bits(8+16 facility code+userid (fill in separately); 7:34 bits(8+24 facility code+userid(fill in separately)) **Default:1(34 bits)**
WGFacilityCode	int (optional)		Facility code **Must be filled in when WiegandType= 6 or 7** Use with MjCardNo. **Not required WiegandType=0 or 1**. See 3.1.6 Wiegand Access Card Number and Built-in Swipe ID Card Number
MjCardNo	unsigned int (optional)		Wiegand Access Card Number(userid) **Must be filled in when WiegandType= 6 or 7** , Use with WGFacilityCode. Separate composition of Wiegand card numbers **when WiegandType=0 or 1** , . See 3.1.6 Wiegand Access Card Number and Built-in Swipe ID Card Number

Key	Type	Values	Description
CardMode	unsigned int (optional)	0~1	Component Access Card Number Model 0: Decimal Composition Card Number 1: Hexadecimal Composition Card Number **Default 0: Decimal Composition Card Number。** See 3.1.6 Wiegand Access Card Number and Built-in Swipe ID Card Number
RFCardMode	unsigned int (optional)	0~1	Component RF (ID) Card Number Model **For built-in card machine type** 0: Decimal Composition Card Number 1: Hexadecimal Composition Card Number **Default 1: Hexadecimal Composition Card Number**
RFIDCard	string (optional)	If RFCardMode=0 , Fill in the decimal string (“1369406761”) ; If RFCardMode=1 , Fill in the hexadecimal string (“519F7D29”)	ID Card Number Maximum length is 18 characters,**for built-in card machine type**(including terminator) See 3.1.6 Wiegand Access Card Number and Built-in Swipe ID Card Number

Key	Type	Values	Description
Tempvalid	int (optional)	0~3	Is it a temporary list 0: Permanent list 1: Temporary list 1 (starting and ending time periods) 2: Temporary list 2 (daily time slots, supported by new database version) 3: Temporary list 3 (number of times valid, supported by new database version) 4: Temporary List 4 (a combination of Temporary List 2 and Temporary List 3, supported by the new database version and after neutral version 9.4)
ValidBegin	string (optional)	YYYY-MM-DDThh:mm:ss	In the case of Temporary List 1, the start time of the entry into force of the Temporary List，e.g., 2018-03-12T09:09:20 If Temporary list 2 or Temporary list 4，e.g., 2018-03-12T09:09:20 2018-03-12 Year/Month/Day for the commencement of the entry into force of the Temporary list 09:09:20 Hour/Minute/Second for the commencement of the entry into force of the Temporary list；
ValidEnd	string (optional)	YYYY-MM-DDThh:mm:ss	In the case of Temporary List 1, the end of the effective date of Temporary List 1，e.g., 2018-03-12T09:10:00 If Temporary list 2 or Temporary list 4，e.g., 2018-03-12T09:09:20 2018-03-12 Year/Month/Day for the commencement of the entry into force of the Temporary list； 09:09:20 Hour/Minute/Second for the commencement of the entry into force of the Temporary list；
EffectNumber	unsigned int (optional)	0 < EffectNumber < 2^32	Effective number of passes through Temporary List 3 or Temporary List 4 1~ 4294967295

Key	Type	Values	Description
isCheckSimilarity	int (optional)	0~1	Whether or not to perform image verification (if the similarity of the added image is greater than the black/white list threshold, it indicates that the list already exists) 0:not calibrated 1:calibrated
PersonalPassword	string (optional)		personal password(six-digit access password)
strategyInfo	json object (optional)		List Associated Access Strategy Information Keywords
strategyData	json array (optional)		List Associated Access Policy Information JSON Array Keywords **Required if strategyInfo is not empty**
strategyID	string (optional)		Access strategy ID **Required if strategyData is not empty**
picinfo	string **Use either with picURI**		Base64 encoded data for face images(no more than1M)
picURI	string **Use either with picinfo**	https://btngps.oss-cn-beijing.aliyuncs.com/image/xxx.jpg	Cloud URI address for face image data

4. Example of Request Message

```
URL of the request:http://192.168.2.10/action/EditPersonNew
Content of the request:
{
    "operator": "EditPersonNew",
    "info": {
        "DeviceID": 1743725,
        "IdType": 0,
        "CustomizeID": 123,
        "PersonType": 0,
```

```
        "Name": "test_example",
        "Gender": 0,
        "Nation": 1,
        "CardType": 0,
        "IdCard": "430923199011044411",
        "Birthday": "1999-11-04",
        "Telnum": "19999999999",
        "Native": "Earth",
        "Address": "ShenZhen",
        "Notes": "Note",
        "MjCardFrom": 2,
        "WiegandType": 1,
        "CardMode": 0,
        "MjCardNo": 45782213,
        "Tempvalid": 0,
        "ValidBegin": "2018-03-12T09:09:20",
        "ValidEnd": "2018-03-12T09:09:20",
        "isCheckSimilarity": 0,
        "strategyInfo": {
            "strategyData": [
                {"strategyID":3}
            ]
        }
    },
    "picURI": "https://tse3-mm.cn.bing.net/th/id/OIP-C.gxpRH969ov5C4qfT9F3IBAHaGX?pid=ImgDet&rs=1"
}
```


5. Explanation of the Paragraph of the Reply Message

   Parameter information(Note: `Optional` is optional):

Key	Type	Values	Description
operator	string	EditPersonNew	Individual list additions or modifications
info	JSON Object		concrete content
code	int		Command execution error code 200-successes，see also 7.19 Add or Modify for Individual Lists Error Codes
Result	String	“Ok”/“Fail”	Result
Detail	String (optional)		Error message when Result is “Fail”.

6. Example of Reply Message

```
{
    "operator": "EditPersonNew",
    "code": 200,
    "info": {
        "Result": "Ok"
    }
}
```

3.1.2 Batch List Additions or Modifications

1. Description

   Add or modify the list according to `CustomizeID` or `PersonUUID`, if `CustomizeID` or `PersonUUID` already exists, then modify the list, if not, then add the list corresponding to `CustomizeID` or `PersonUUID`; this interface can only work with either `CustomizeID` or `PersonUUID`. `CustomizeID` or `PersonUUID` can only be used for one or the other. This interface only supports to add person's picture by `URI`.**Maximum number of personnel to be issued at one time is 1000。**This interface is a combination of Add and Edit.，**So when editing a list of people, fields involving other list attributes don't need to be sent down**，Fields that are not sent down retain the value of the last edit. For example, if you need to change a person's name after adding them to the list for the first time, you only need to issue the person's `CustomizeID` or `PersonUUID` and the name field that needs to be changed.

2. API Description

Items	Description
Operator	`EditPersonsNew`
Request URL	`http://<server ipaddr>/action/EditPersonsNew` (Where `<server ipaddr>` is the device IP，for example:192.168.1.10)

3. Explanation of the Paragraph of the Requested Message

   Parameter information(Note: `Optional` is optional):

Key	Type	Values	Description
operator	string	EditPersonsNew	Batch list additions or modifications
DeviceID	int		ID number of the device to be operated See 6.8.1 Getting Device Information
IdType	int	0/2	Type of ID to be operated 0:CustomizeID, 2:PersonUUID
Total	int	maximum values is 1000	Number of people to be added or modified in bulk in this request
info	json object		Concrete content
CustomizeID	unsigned int **Must be filled in when IdType=0**		User-defined ID You can manage the corresponding personnel information such as modification and deletion by this ID number. Note: You can not use the same ID to add people repeatedly, otherwise it will lead to errors in modifying and deleting operations.
PersonUUID	string **Must be filled in when IdType=2**		User-defined UUID This UUID number can be used to manage personnel information such as modification and deletion. Note: You can not use the same ID to add people repeatedly, otherwise it will lead to errors in operations such as modification and deletion (not more than 48 bytes)
PersonType	int (optional)	0~1	Type of list 0: white list 1: black list **Default 0(white list): when adding people**
Name	string (optional)		Name Maximum length is 32 characters (including terminator) **Default “” (empty):when adding a new person**

Key	Type	Values	Description
Gender	int (optional)	0~1	Genders 0: male 1: female **Default 0(male): when adding people**
Nation	int (optional)	Reserve	Reserve
CardType	int (optional)		ID type 0:ID **Default 0(ID): when adding people**
IdCard	string (optional)		ID Number Maximum length is 32 characters (including terminator) **Default “” (empty):when adding a new person**
Birthday	string (optional)	YYYY-MM-DD	Birthday e.g., 2018-1-1 **Default 1970-00-00:when adding a new person**
Telnum	string (optional)		Telephone Number Maximum length is 32 characters (including terminator) **Default “” (empty):when adding a new person**
Native	string (optional)		Native Maximum length is 32 characters (including terminator) **Default “” (empty):when adding a new person**
Address	string (optional)		Address Maximum length is 72 characters (including terminator) **Default “” (empty):when adding a new person**
Notes	string (optional)		Notes Maximum length is 72 characters (including terminator) **Default “” (empty):when adding a new person**

Key	Type	Values	Description
MjCardFrom	int (optional)	0~3	Wiegand Card Number Generation Method 0: public number 1: automatic generation 2: manual input 3: do not use access card numbers **Default 0(public number):when adding a new person**
WiegandType	int (optional)	0~1 or 6~7	The Wiegand protocols that form the basis for the Wiegand card numbers **Must be filled in when MjCardFrom=2** ; 0: 26 bits 1: 34 bits 6:26 bits(8+16 facility code+userid(fill in separately)); 7:34 bits(8+24 facility code+userid(fill in separately)) **Default:1(34 bits)**
WGFacilityCode	int (optional)		Facility code **Must be filled in when WiegandType= 6 or 7.**Use with MjCardNo. **Not required WiegandType=0 or 1。** See 3.1.6 Wiegand Access Card Number and Built-in Swipe ID Card Number
MjCardNo	unsigned int (optional)		Wiegand Access Card Number(userid) **Must be filled in when WiegandType= 6 or 7** , Use with WGFacilityCode. Separate composition of Wiegand card numbers **when WiegandType=0 or 1** See 3.1.6 Wiegand Access Card Number and Built-in Swipe ID Card Number

Key	Type	Values	Description
CardMode	unsigned int (optional)	0~1	Component Access Card Number Model 0: Decimal Composition Card Number 1: Hexadecimal Composition Card Number **Default 0: Decimal Composition Card Number.** See 3.1.6 Wiegand Access Card Number and Built-in Swipe ID Card Number
RFCardMode	unsigned int (optional)	0~1	Component RF (ID) Card Number Model **For built-in card machine type** 0: Decimal Composition Card Number 1: Hexadecimal Composition Card Number **Default 1: Hexadecimal Composition Card Number**
RFIDCard	string (optional)	If RFCardMode=0，Fill in the decimal string (“1369406761”)；If RFCardMode=1，Fill in the hexadecimal string (“519F7D29”)	ID Card Number Maximum length is 18 characters,**For built-in card machine type**(including terminator) See 3.1.6 Wiegand Access Card Number and Built-in Swipe ID Card Number

Key	Type	Values	Description
Tempvalid	int (optional)	0~3	Is it a temporary list 0: Permanent list 1: Temporary list 1 (starting and ending time periods) 2: Temporary list 2 (daily time slots, supported by new database version) 3: Temporary list 3 (number of times valid, supported by new database version) 4: Temporary List 4 (a combination of Temporary List 2 and Temporary List 3, supported by the new database version and after neutral version 9.4) **Default 0:Permanent list**
ValidBegin	string (optional)	YYYY-MM-DDThh:mm:ss	In the case of Temporary List 1, the start time of the entry into force of the Temporary List，e.g., 2018-03-12T09:09:20 If Temporary list 2 or Temporary list 4，e.g., 2018-03-12T09:09:20 2018-03-12 Year/Month/Day for the commencement of the entry into force of the Temporary list； 09:09:20Hour/Minute/Second for the commencement of the entry into force of the Temporary list；
ValidEnd	string (optional)	YYYY-MM-DDThh:mm:ss	In the case of Temporary List 1, the end of the effective date of Temporary List 1，e.g., 2018-03-12T09:10:00 If Temporary list 2 or Temporary list 4，e.g., 2018-03-12T09:09:20 2018-03-12 Year/Month/Day for the commencement of the entry into force of the Temporary list； 09:09:20 Hour/Minute/Second for the commencement of the entry into force of the Temporary list；
EffectNumber	unsigned int (optional)	0 < EffectNumber < 2^32	Effective number of passes through Temporary List 3 or Temporary List 4,1~4294967295

Key	Type	Values	Description
PersonalPassword	string (optional)		personal password(six-digit access password)
isCheckSimilarity	int (optional)	0~1	Whether or not to perform image verification (if the similarity of the added image is greater than the black/white list threshold, it indicates that the list already exists) 0:not calibrated 1:calibrated
strategyInfo	json object (optional)		List Associated Access Strategy Information Keywords
strategyData	json array (optional)		List Associated Access Policy Information JSON Array Keywords **Required if strategyInfo is not empty**
strategyID	string (optional)		Access strategy ID **Required if strategyData is not empty**
picURI	string (optional)	https://btngps.oss-cn-beijing.aliyuncs.com/image/xxx.jpg	Cloud URI address for face image data

4. Example of Request Message

```
URL of the request:http://192.168.2.10/action/EditPersonsNew
Content of the request:
{
    "operator": "EditPersonsNew",
    "DeviceID": 1743725,
    "IdType": 0,
    "Total": 2,
    "info": [
        {
            "CustomizeID": 123,
            "PersonType": 0,
            "Name": "test1",
            "Gender": 0,
            "Nation": 1,
            "CardType": 0,
            "IdCard": "430923199011044411",
            "Birthday": "1999-11-04",
            "Telnum": "19999999999",
```

```
            "Native": "Earth",
            "Address": "Shenzhen",
            "Notes": "Notes",
            "MjCardFrom": 2,
            "WiegandType": 1,
            "CardMode": 0,
            "MjCardNo": 1046230,
            "Tempvalid": 0,
            "ValidBegin": "2018-03-12T09:09:20",
            "ValidEnd": "2018-03-12T09:09:20",
            "picURI": "https://tse4-mm.cn.bing.net/th/id/OIP-C._BHy_gVOd_Ie7VGTAPHoEgHaFj?pid=ImgDet&rs=1"
        },
        {
            "CustomizeID": 456,
            "PersonType": 0,
            "Name": "test2",
            "Gender": 0,
            "Nation": 1,
            "CardType": 0,
            "IdCard": "430923199011044412",
            "Birthday": "1999-11-05",
            "Telnum": "18888888888",
            "Native": "Fujian",
            "Address": "Ningde",
            "Notes": "Notes",
            "MjCardFrom": 2,
            "WiegandType": 1,
            "CardMode": 0,
            "MjCardNo": 1046240,
            "Tempvalid": 0,
            "ValidBegin": "2018-03-12T09:09:20",
            "ValidEnd": "2018-03-12T09:09:20",
            "picURI": "https://tse2-mm.cn.bing.net/th/id/OIP-C.-iptOFuk2qc7VLulCiQV4gAAAA?pid=ImgDet&rs=1"
        }
    ]
}
```


5. Explanation of the Paragraph of the Reply Message

   Parameter information(Note: `Optional` is optional):

Key	Type	Values	Description
operator		EditPersonsNew	Batch list additions or modifications
code	Int		Instruction Execution Error Codes 200-successes，see also 7.18 Add or Modify Personnel in Bulk Error Codes
info	Object		concrete content
DeviceID	String		ID number of the device to be operated
EditsErrNum	int	0~1000	Number of failed batch additions or modifications, return parameter
EditsErrInfo	Object		Batch add or modify personnel failure information(CustomizeID/PersonUUID + errcode),return parameter，errcode see 7.18 Add or Modify Personnel in Bulk Error Codes
EditsSucNum	Int	0~1000	Number of successful batch additions or modifications,return parameter
EditsSucInfo	Object		Batch add or modify personnel success information(CustomizeID/PersonUUID) ,return parameter
CustomizeID	String (optional)		User-defined ID You can use this ID number to modify, delete, etc. to manage the corresponding personnel information. Note: You cannot add personnel with the same ID repeatedly, otherwise it will cause errors in modification and deletion operations.
PersonUUID	string (optional)		User-defined UUID This UUID number can be used to manage personnel information such as modification and deletion. Note: You can not use the same ID to add people repeatedly, otherwise it will lead to errors in operations such as modification and deletion (not more than 48 bytes)
Result	String	"Ok"/"Fail"	Results of the implementation of the batch increase in personnel instruction
Detail	String (optional)		Error message when Result is “Fail”.

6. Example of Reply Message

Bulk addition or modification of persons by `CustomizeID`,reply message:

```
{
    "operator": "EditPersonsNew",
    "code": 200,
    "info": {
        "DeviceID": 1743725,
        "EditsErrNum": "1",
        "EditsSucNum": "1",
        "EditsErrInfo": [
            {
                "CustomizeID": "12389",
                "errcode": "475"
            }
        ],
        "EditsSucInfo": [
            {
                "CustomizeID": "456"
            }
        ],
        "Result": "OK"
    }
}
```

Bulk addition or modification of persons by `PersonUUID` ,reply message:

```
{
    "operator": "EditPersonsNew",
    "code": 200,
    "info": {
        "DeviceID": 1743725,
        "EditsErrNum": "0",
        "EditsSucNum": "2",
        "EditsErrInfo": [],
        "EditsSucInfo": [
            {
                "PersonUUID": "abc"
            },
            {
                "PersonUUID": "efg"
            }
        ],
        "Result": "OK"
    }
}
```

3.1.3 Single or Multiple List Deletions

1. Description

   Deletion of the personnel list interface also deletes **the control records (authentication records) of the corresponding personnel list and is unrecoverable**，please proceed with caution.Personnel bindings to the pass policy are also deleted, but not the pass policy.Deletion of a list provides a `LibID` or `customID` or `PersonUUID` deletion method, and `LibID` or `customID` or `PersonUUID` cannot be zero.

2. API Description

Items	Description
Operator	`DeletePerson`
Request URL	`http://<server ipaddr>/action/DeletePerson` (Where `<server ipaddr>` is the device IP，for example:192.168.1.10)

3. Explanation of the paragraph of the requested paper

   Parameter information(Note: `Optional` is optional):

Key	Type	Values	Description
operator	string	DeletePerson	Deletion of information on persons in the List database
info	json object		Concrete content
DeviceID	int		ID number of the device to be operated See 6.8.1 Getting Device Information
TotalNum	int (optional)	1~256	How many people need to be deleted for this operation
IdType	int	0~2	0: User-defined CustomizeID， 1: Equipment database LibID, 2: PersonUUID
LibID	json array **Must be filled in when IdType=1**		A collection of one or more `LibIDs`. `LibID` is unsigned int type, format is Json array of `LibID`.
CustomizeID	json array **Must be filled in when IdType=0**		A collection of one or more `CustomizeID` `CustomizeID` is int type，format is Json array of `CustomizeID`
PersonUUID	json array **Must be filled in whenIdType=2**		A collection of one or more `UUID` UUID is a string of no more than 48 bytes.，format is Json array of UUID

4. Example of Request Message

```
URL of the request:http://192.168.2.10/action/DeletePerson
Content of the request:
{
    "operator": "DeletePerson",
    "info": {
        "DeviceID": 1743726,
        "TotalNum": 2,
        "IdType": 0,
        "CustomizeID": [
            123,
            456
        ]
    }
}
```

5. Explanation of the Paragraph of the Reply Message

   Parameter information(Note: `Optional` is optional):

Key	Type	Values	Description
operator	string	DeletePerson	Single or multiple list deletions
info	JSON Object		Concrete content
code	int		Command execution error code 200-successes，see also7.20 Single or Multiple List Deletion Error Codes
Result	String	“Ok”/“Fail”	Operating result
Detail	String (optional)		Error message when Result is “Fail”.

6. Example of Reply Message

```
{
    "operator": "DeletePerson",
    "code": 200,
    "info": {
        "Result": "Ok"
    }
}
```

3.1.4 All Lists Deleted

1. Description

   All personnel list deletion interface at the same time will delete all personnel list **control records (authentication records) and can not be restored** , careful operation. Call to delete all personnel list success, face recognition all-in-one opportunity to **automatic restart!**

2. API Description

Items	Description
Operator	`DeleteAllPerson`
Request URL	`http://<server ipaddr>/action/DeleteAllPerson` (Where `<server ipaddr>` is the device IP, for example:192.168.1.10)

3. Explanation of the paragraph of the requested paper

   Parameter information(Note: `Optional` is optional):

Key	Type	Values	Description
operator	string	DeleteAllPerson	All lists deleted
info	json object		Concrete content
DeleteAllPersonCheck	json array	1	Required if operator is DeleteAllPerson.

4. Example of Request Message

```
URL of the request:http://192.168.2.10/action/DeleteAllPerson
Content of the request:
{
    "operator": "DeleteAllPerson",
    "info": {
        "DeleteAllPersonCheck": 1
    }
}
```

5. Explanation of the Paragraph of the Reply Message

   Parameter information(Note: `Optional` is optional):

Key	Type	Values	Description
operator	string	DeleteAllPerson	All lists deleted
info	JSON Object		Concrete content
code	int		Command execution error code 200-successes，see also 7.21 All Lists Delete Error Codes
Result	String	“Ok”/“Fail”	Operating result
Detail	String (optional)		Error message when Result is “Fail”.

6. Example of Reply Message

```
{
    "operator": "DeleteAllPerson",
    "code":200,
    "info": {
    "Result": "Ok"
    }
}
```

3.1.5 Getting the Database Version

1. Description

   The new database supports Temporary List 2 and Temporary List 3. The old database is factory restored in the latest version, the device is rebooted, and the device boots to the latest database.

2. API Description

Items	Description
Operator	`GetDatabaseVer`
Request URL	`http://<server ipaddr>/action/GetDatabaseVer` (Where `<server ipaddr>` is the device IP, for example:192.168.1.10)

3. Example of Request Message

```
URL of the request:http://192.168.2.10/action/GetDatabaseVer
Content of the request:（NULL）
```

4. Explanation of the Paragraph of the Reply Message

   Parameter information(Note: `Optional` is optional):

Key	Type	Values	Description
operator	string	GetDatabaseVer	Getting the database version
info	json object		Concrete content
DatabaseVersion	int	0~1	Database version 0:Old database 1:New database

5. Example of Reply Message

```
{
    "operator": "GetDatabaseVer",
    "info": {
        "DatabaseVersion": 1
    }
}
```

3.1.6 Wiegand Access Card Number and Built-in Swipe ID Card Number

1. Description

   Wiegand access control card number is mainly for the external card reader models, with the card mode under the comparison of the pass need to fill in this card number; external card reader Wiegand output are used in this Wiegand access control card number.In the mode where the Wiegand protocol is selected as **8+16** or **8+24**,the `WGFacilityCode` and `MjCardNo` fields together form the Wiegand access card number.The Wiegand protocol is either **26-bit** or **34-bit** mode, the `MjCardNo` field alone constitutes the Wiegand access card number.

   The built-in card swiping model of the face recognition all-in-one machine uses the `RFIDCard` (ID card number) field for the swipe card **comparison pass** , and if the built-in swipe card model wants to output the Wiegand card number, it is also necessary to fill in the corresponding Wiegand access card number in accordance with the 3.1 List Editing and Modification rules;

   That is, the `RFIDCard` field is used for comparison pass and Wiegand output **in swipe mode**, while the built-in swipe model outputs Wiegand access control card number when swiping face pass, the built-in swipe model needs to fill in the same field according to the corresponding rules in order to output the same Wiegand card number in face swipe and swipe mode.

   The Wiegand access card number fields returned in 4.4 Reporting of Certification Results,3.6 List Search and Query and 3.7 Control Record Inquiryare all returned according to the `Wiegand` and `CardMode` parameters of 6.1.2 Setting of Door Opening Conditions and Output Control Parameters .When the system parameter **Wiegand is 4 or 5 or 6 or 7**, the returned `WGFacilityCode` and `MjCardNo` fields together form the Wiegand access card number; when **Wiegand is 0 or 1**, the `MjCardNo` field alone forms the Wiegand access card number.A `CardMode` card mode of 0 (decimal) returns the Wegen access card number in decimal, and a `CardMode` card mode of 1 (hexadecimal) returns the Wiegand access card number in hexadecimal.The `RFIDCard` field is returned as a string, and when adding/modifying the list, the corresponding decimal/hexadecimal value will be deposited depending on the value of `RFCardMode`.

3.2 Photo Ads

The version of Face Machine software with (AD) in the version number supports polling time to play the advertisement picture, the picture format supports png/jpg/bmp format picture, and currently supports up to 10 advertisement pictures. Default advertisement effect: Stranger advertisement picture does not disappear, the UI of the face machine will show the main video picture when the personnel passes the state, continue to verify the reset time, and switch back to the advertisement picture.

5 inch machine advertisement picture pixel size 720*1280,7 inch machine advertisement picture pixel size 600*800,8 inch regular machine advertisement picture pixel size 800*1068,8 inch temperature measuring machine advertisement picture pixel size 800*1117,8 inch touch screen machine advertisement picture pixel size 800*1280.Other sizes of equipment does not support advertising function for the time being

3.2.1 Upload Ads

1. Description

   Add or modify advertisements.

2. API Description

Items	Description
Operator	`EditAD`
Request URL	`http://<server ipaddr>/action/EditAD` (Where `<server ipaddr>` is the device IP, for example:192.168.1.10)

3. Explanation of the Paragraph of the Requested Message

   Parameter information(Note: `Optional` is optional):

Key	Type	Values	Description
operator	String	EditAD	Add or modify advertisements.
info	Object		Concrete content
DeviceID	int (optional)		ID number of the device to be operated See 6.8.1 Getting Device Information
addid	String(optional)	Operation ID	reserve
adslot	int	0~9	Advertising Slot
path	String		Image download path
polltime	int (optional)	Default 10 seconds	Length of time each advertisement is polled

4. Example of Request Message

```
URL of the request:http://192.168.2.10/action/EditAD
Content of the request:
{
    "operator": "EditAD",
    "info": {
        "adid": "",
        "path":
"http://pics.sc.chinaz.com/files/pic/pic9/201903/zzpic16841.jpg",
        "adslot": 0,
        "polltime": 10
    }
}
```

5. Explanation of the Paragraph of the Reply Message

   Parameter information(Note: `Optional` is optional):

Key	Type	Values	Description
operator	string	EditAD	Upload Ads
info	JSON Object		Concrete content
code	int		Command execution error code 200-successes，see also 7.39 Upload Ads Error Codes
Result	String	“Ok”/ “Fail”	Operating result
Detail	String (optional)		Error message when Result is “Fail”.

6. Example of Reply Message

```
{
    "operator": "EditAD",
    "code":200,
    "info": {
        "Result": "Ok"
    }
}
```

3.2.2 Drop Ads

1. Description

   Drop Ads.

2. API Description

Items	Description
Operator	`DelAD`
Request URL	`http://<server ipaddr>/action/DelAD` (Where `<server ipaddr>` is the device IP, for example:192.168.1.10)

3. Explanation of the Paragraph of the Requested Message

   Parameter information(Note: `Optional` is optional):

Key	Type	Values	Description
operator	String	DelAD	Drop Ads
info	Object		Concrete content
DeviceID	int (optional)		ID number of the device to be operated See 6.8.1 Getting Device Information
adid	String(optional)	Operation ID	reserve
adslot	int	0~9	Advertising Slot

4. Example of Request Message

```
URL of the request:
Content of the request:
{
    "operator":"DelAD",
    "info": {
        "adid":"",
        "adslot":0
    }
}
```

5. Explanation of the Paragraph of the Reply Message

   Parameter information(Note: `Optional` is optional):

Key	Type	Values	Description
operator	string	DelAD	Drop Ads
info	JSON Object		Concrete content
code	int		Command execution error code 200-successes，see also 7.40 Drop Ads Error Codes
Result	String	“Ok”/ “Fail”	Operating result
Detail	String (optional)		Error message when Result is “Fail”.

6. Example of Reply Message

```
{
    "operator": "DeltAD",
    "code": 200,
    "info": {
        "Result": "Ok"
    }
}
```

3.3 QR Code Display

The QR code section can be divided into two parts:

One part is that the QR code image can be displayed on the whole UI interface of the All-in-One machine. The All-in-One PC supports two forms of displaying the QR code image: one is to directly send down the string data of the QR code image to be encoded, and the All-in-One Machine generates the QR code image; the other is to directly send down the base64 image data of the QR code, and the All-in-One PC directly displays the image.

The other part is that the 485 interface of the all-in-one machine supports the external QR code scanner to upload the QR code scanning result data directly through the transmission. For the face recognition MFP that supports external QR code scanner, under the setting of QR code scanning information uploading mode, the parameter setting is shown in 4.5 QR Code Reporting.

For machines with different screen sizes, you need to calculate the starting position and image size to avoid out-of-bounds. If you send the base64 data of the image, you need to specify the original real image format of the image, otherwise the image data can not be displayed normally.

3.3.1 Setting the QR Code Image

1. Description

   Setting the QR code image.

2. API Description

Items	Description
Operator	`ShowQRCode`
Request URL	`http://<server ipaddr>/action/ShowQRCode` (Where `<server ipaddr>` is the device IP, for example:192.168.1.10)

3. Explanation of the Paragraph of the Requested Message

   Parameter information(Note: `Optional` is optional):

Key	Type	Values	Description
operator	string	ShowQRCode	Setting the QR code image.
info	json object		Concrete content
ShowStatus	int (optional)		Whether to display QR code image 0:Cancel display 1:Display Default 1:Display
ImageType	Int (optional)	1~4	Display Image Type 1:String of QR code images to be displayed 2:The format of the image to be displayed is png 3:The format of the image to be displayed is jpg 4:The format of the image to be displayed is bmp Default 1:String of QR code images to be displayed.
AbsX	Int (optional)	10~Maximum value for different UI heights Default:10	Display image based on UI start position X-value
AbsY	Int (optional)	10~Maximum value for different UI widths Default:10	Display image based on UI start position Y-value
ImageW	Int (optional)	10~400(Default:200)	Display image width
ImageH	Int (optional)	10~400(Default:200)	Display image height
QRCodeData	string	**32K**Bytes(including terminator)	String data for the QR code image to be encoded when ImageType = 1 BASE64 data for the png image to be displayed when ImageType = 2 BASE64 data for the jpg image to be displayed when ImageType = 3 BASE64 data for the bmp image to be displayed when ImageType = 4

4. Example of Request Message

```
URL of the request:http://192.168.2.10/action/ShowQRCode
Content of the request:
{
    "operator": "ShowQRCode",
    "info": {
        "ImageType": 1,
        "AbsX": 10,
        "AbsY": 10,
        "ImageW": 200,
        "ImageH": 200,
        "QRCodeData": "A1B2C3D4E5r6t7y8u9Pn"
    }
}
```

5. Explanation of the Paragraph of the Reply Message

   Parameter information(Note: `Optional` is optional):

Key	Type	Values	Description
operator	string	ShowQRCode	Setting the QR code image.
info	JSON Object		Concrete content
code	int		Command execution error code 200-successes,see also 7.41 Setting QR Code Image Error Codes
Result	String	“Ok”/“Fail”	Operating result
Detail	String (optional)		Error message when Result is "Fail"

6. Example of Reply Message

```
{
    "operator": "ShowQRCode",
    "code": 200,
    "info": {
        "Result": "Ok"
    }
}
```

3.4 Photo Services

Image service functions include: image comparison similarity, searching local face database by image, detecting image faces, and obtaining panoramic image.

3.4.1 Image Similarity Comparison

1. Description

   This interface provides for comparing the similarity of two images.

2. API Description

Items	Description
Operator	`GetPictureSimilarity`
Request URL	`http://<server ipaddr>/action/GetPictureSimilarity` (Where `<server ipaddr>` is the device IP, for example:192.168.1.10)

3. Explanation of the Paragraph of the Requested Message

   Parameter information(Note: `Optional` is optional):

Key	Type	Values	Description
operator	string	GetPictureSimilarity	Getting the similarity of people's images
picinfo1	string		Base64 encoded data of face images (no more than 1M)
picinfo2	string		Base64 encoded data of face images (no more than 1M)

4. Example of Request Message

```
URL of the request:
Content of the request:
{
    "operator":"GetPictureSimilarity",
    "picinfo1":"......",
    "picinfo2":"......"
}
```

5. Explanation of the Paragraph of the Reply Message

   Parameter information(Note: `Optional` is optional):

Key	Type	Values	Description
operator	string	GetPictureSimilarity	Getting the similarity of people's images
info	json object		Concrete content
Similarity	float		Face Image Similarity
code	int		Command execution error code 200-successes,see also7.35 Image Comparison Similarity Error Codes
Result	String	“Ok”/“Fail”	Operating result
Detail	String (optional)		Error message when Result is "Fail"

6. Example of Reply Message

```
{
    "operator": "GetPictureSimilarity",
    "info": {
        "Similarity": 93.4
    }
}
```

3.4.2 Search Local Face Database by Image

1. Description

   The interface provides the comparison with the picture and the local face database of the face machine to get the corresponding results.

2. API Description

Items	Description
Operator	`GetPictureSearch`
Request URL	`http://<server ipaddr>/action/GetPictureSearch` (Where `<server ipaddr>` is the device IP, for example:192.168.1.10)

3. Explanation of the Paragraph of the Requested Message

   Parameter information(Note: `Optional` is optional):

Key	Type	Values	Description
operator	string	GetPictureSearch	Search local face database by image
info	json object		Concrete content
MaxSimilarity	float		Similarity threshold
MaxNum	int		Maximum number
picinfo	string		Base64 encoded data of face images (no more than 1M)

4. Example of Request Message

```
URL of the request:http://192.168.2.10/action/GetPictureSearch
Content of the request:
{
    "operator": "GetPictureSearch",
    "info": {
        "MaxSimilarity":80,
        "MaxNum":5
    },
    "picinfo":"......"
}
```

5. Explanation of the Paragraph of the Reply Message

Key	Type	Values	Description
operator	string	GetPictureSearch	Get the information and image of the face image that satisfies the condition
info	json object		Concrete content
TotalNum	int		Total number of lists returned from search results (up to 20)
SearchInfo	Array		List of returns
Uid	unsigned int		Database ID
Type	int	0~1	Type of list 0: white list 1: black list
Name	string		Name (not more than 64 bytes)
Cardtype	int		Card Number Type
MjCardNo	unsigned int		Access card number
Sex	int		Genders
Nation	int		Nation
Certype	int		Type of document
Cernumber	string		Identity card number (not more than 32 bytes)
Birth	string		Birth year
Phone	string		Telephone number
Place	string		Native (not more than 32 bytes)
Addr	string		Address (not more than 72 bytes)
Notes	string		Notes (not more than 64 bytes)
Filetype	int		Document type
Fileindex	unsigned int		Document serial number
Filepos	unsigned int		File Location

Key	Type	Values	Description
CustomizeID	unsigned int		User-defined ID This ID number can be used to modify, delete, and manage personnel information. Note: You can not use the same ID to add people repeatedly, otherwise it will lead to errors in modifying and deleting operations.
PersonUUID	string		User-defined UUID(not more than 48 bytes) This ID number can be used to modify, delete, and manage personnel information. Note: You can not use the same UUID to add people repeatedly, otherwise it will lead to errors in modifying and deleting operations.
Time	string		Creation time
code	int		Command execution error code 200-successes,see also7.36 Search for Local Face Database by Image Error Codes
Result	String	“Ok”/“Fail”	Operating result
Detail	String (optional)		Error message when Result is "Fail"

6. Example of Reply Message

```
{
    "operator": "GetPictureSearch",
    "info": {
        "TotalNum": 1,
        "SearchInfo": [
            {
                "Uid": 1,
                "Type": 0,
                "Name": "test_example",
                "Cardtype": 0,
                "MjCardNo": 1,
                "Sex": 0,
                "Nation": 1,
                "Certype": 0,
                "Cernumber": " ",
                "Birth": "2000-01-01",
                "Phone": " ",
                "Place": " ",
                "Addr": " ",
                "Notes": " ",
```

```
                "Filetype": 0,
                "Fileindex": 0,
                "Filepos": 0,
                "CustomizeID": 0,
                "PersonUUID": " ",
                "Time": "2022-12-20/18:02:24"
            }
        ]
    }
}
```

3.4.3 Detecting Faces in Image

1. Description

   This interface provides rough detection of whether a face can be extracted from an image.

2. API Description

Items	Description
Operator	`DetectFaceFromPic`
Request URL	`http://<server ipaddr>/action/DetectFaceFromPic` (Where `<server ipaddr>` is the device IP, for example:192.168.1.10)

3. Explanation of the Paragraph of the Requested Message

   Parameter information(Note: `Optional` is optional):

Key	Type	Values	Description
operator	string	DetectFaceFromPic	Detecting Faces in Images
picinfo	string		Base64 encoded data of face images (no more than 1M)

4. Example of Request Message

```
URL of the request:http://192.168.2.10/action/DetectFaceFromPic
Content of the request:
{
    "operator": "DetectFaceFromPic",
    "picinfo":"......"
}
```

5. Explanation of the Paragraph of the Reply Message

   Parameter information:Whether face is detected or not

Key	Type	Values	Description
operator	string	DetectFaceFromPic	Detecting Faces in Images
info	json object		Concrete content
DetectFace	int	0~1	Detecting Image Face Results: 0:No face detected 1:Face detected
code	int		Command execution error code 200-successes,see also7.37 Detecting Image Face Error Codes
Result	String	“Ok”/“Fail”	Operating result
Detail	String (optional)		Error message when Result is "Fail"

6. Example of Reply Message

```
{
    "operator": "DetectFaceFromPic",
    "info": {
        "DetectFace": 1
    }
}
```

3.4.4 Manual Snapshot

1. Description

   This interface is only supported in special versions (versions beginning with V14). This interface supports getting the panorama capture image or face image returned by the device.

2. API Description

Items	Description
Operator	`FrontalFaceSnap`
Request URL	`http://<server ipaddr>/action/FrontalFaceSnap` (Where `<server ipaddr>` is the device IP, for example:192.168.1.10)

3. Explanation of the Paragraph of the Requested Message

   Parameter information(Note: `Optional` is optional):

Key	Type	Values	Description
operator	string	FrontalFaceSnap	Get a snapshot panorama
info	json object		Concrete content
DeviceID	int		ID number of the device to be operated See 6.8.1 Getting Device Information
Overall	int	0~1	Image Type 0:Get face image(get failed as timeout return) 1:Get a snapshot panorama
TimeOut	int (optional)	0~30	Manual capture timeout, default value is 15, unit: s

4. Example of Request Message

```
URL of the request:
Content of the request:
{
    "operator": "FrontalFaceSnap",
    "info": {
        "DeviceID": 1743726,
        "Overall": 1
    }
}
```

5. Explanation of the Paragraph of the Reply Message

Key	Type	Values	Description
operator	string	FrontalFaceSnap	Manual snapshot
FacePic	string		Base64 encoded data of the image (no more than 1.5M), return data
code	int		Command execution error code 200-successes,see also7.38 Manual Snapshot Error Codes
Result	String	“Ok”/“Fail”	Operating result
Detail	String (optional)		Error message when Result is "Fail"

6. Example of Reply Message

```
{
    "operator": "FrontalFaceSnap",
    "code": 200,
    "info": {
        "Result": "Ok"
    },
    "FacePic": "....."
}
```

3.4.5 Get Face Image

1. Description

   Get face images in the device based on parameters such as file type, file index, and file location.

2. API Description

Items	Description
Operator	`GetImage`
Request URL	`http://<server ipaddr>/action/GetImage` (Where `<server ipaddr>` is the device IP, for example:192.168.1.10)

3. Explanation of the Paragraph of the Requested Message

   Parameter information(Note: `Optional` is optional):

Key	Type	Values	Description
operator	string	GetImage	Search local face database by image
info	json object		Concrete content
dwfiletype	unsigned int	0~2	Document type 0:Black and white list image files 1:ID card image files 2:Real-time captured image files (Only black and white list images are currently supported)
dwfileindex	unsigned int		Document index reference3.4.2 Search Local Face Database by Image
dwfilepos	unsigned int		Document Location Reference 3.4.2 Search Local Face Database by Image

4. Example of Request Message

```
URL of the request:http://192.168.2.10/action/GetImage
Content of the request:
{
    "operator": "GetImage",
    "info": {
        "dwfiletype": 0,
        "dwfileindex": 0,
        "dwfilepos": 0
    }
}
```

5. Explanation of the Paragraph of the Reply Message

   If getting the image is successful, the reply message body is the image data.

   If getting the image fail，Parameter information(Note:`Optional`is optional):

Key	Type	Values	Description
operator	string	HttpGetImage	Get face image
info	JSON Object		Concrete content
code	int		Command execution error code 200-successes,see also7.43 Get Face Image Error Codes
Result	String	“Ok”/“Fail”	Operating result
Detail	String (optional)		Error message when Result is "Fail"

6. Example of Reply Message

```
{
    "operator": "HttpGetImage",
    "code": 463,
    "info": {
        "Result": "Fail",
        "Detail": "Unkonw Parameter"
    }
}
```

3.5 Playing Voice Files

Gets the name of an audio file that can be played in the device, or you can play a specified voice file.

3.5.1 Get the Number of Files to Play and Their Names

1. Description

   Get a list of voice file names that can be played on the device

2. API Description

Items	Description
Operator	`GetAudio`
Request URL	`http://<server ipaddr>/action/GetAudio` (Where `<server ipaddr>` is the device IP, for example:192.168.1.10)

3. Example of Request Message

```
URL of the request:http://172.168.2.10/action/GetAudio
Content of the request:（空）
```

4. Explanation of the Paragraph of the Reply Message

   Parameter information(Note: `Optional` is optional):

Key	Type	Values	Description
operator	string	GetAudio	Get a list of voice file names that can be played on the device
info	JSON Object		Concrete content
AudioNum	int		Number of audio files
AudioName	JSON array		List of audio file names in the format of a JSON array of audio file names
AudioCustomerNum	int		Number of user-defined sound files
AudioCustomerName	JSON array		User-defined sound file name collection

5. Example of Reply Message

```
{
    "operator": "GetAudio",
```

```
    "info": {
        "AudioNum": 58,
        "AudioName": [
            "listExpired.wav",
            "noIDCardReader.wav",
            "welcome.wav",
            ...
            "SLYellowSorry.wav",
            "tempertureOk.wav",
            "unRecorded.wav",
            "greenCodeWelcome.wav"
        ],
        "AudioCustomerNum": "0",
        "AudioCustomerName": []
    }
}
```

3.5.2 Playing Audio Files

1. Description

   Play an audio file with the specified filename.

2. API Description

Items	Description
Operator	`SetAudio`
Request URL	`http://<server ipaddr>/action/SetAudio` (Where `<server ipaddr>` is the device IP, for example:192.168.1.10)

3. Explanation of the Paragraph of the Requested Message

   Parameter information(Note: `Optional` is optional):

Key	Type	Values	Description
operator	string	SetAudio	Play an audio file with the specified filename
info	JSON Object		Concrete content
DeviceID	int		ID number of the device to be operated See 6.8.1 Getting Device Information
AudioType	int	0~1	Types of Playback Files 0:Playing Local Files 1:Playing customized files
AudioName	string		Name of the audio file to be played (Must be a wav file, sample rate:16000Hz, sample bit:16 bit, mono. Maximum file name length is 64 bytes)

4. Example of Request Message

```
URL of the request:http://172.168.2.10/action/SetAudio
Content of the request:
{
    "operator": "SetAudio",
    "info": {
        "DeviceID":1300001,
        "AudioType":0,
        "AudioName":"welcome.wav"
    }
}
```

5. Explanation of the Paragraph of the Reply Message

   Parameter information(Note: `Optional` is optional):

Key	Type	Values	Description
operator	string	SetAudio	Play an audio file with the specified filename
info	JSON Object		Concrete content
code	int		Command execution error code 200-successes,see also 7.3 Playing Audio Files Error Codes
Result	String	“Ok”/ “Fail”	Operating result
Detail	String (optional)		Error message when Result is "Fail"

6. Example of Reply Message

```
{
    "operator": "SetAudio",
    "code": 200,
    "info": {
        "Result": "Ok"
    }
}
```

3.6 List Search and Query

Query the list that matches the search criteria based on the corresponding search criteria.

3.6.1 Individual List Search

1. Description

   Query the list based on the `CustomizeID`, `UUID`, or database ID you are looking for.

2. API Description

Items	Description
Operator	`SearchPerson`
Request URL	`http://<server ipaddr>/action/SearchPerson` (Where `<server ipaddr>` is the device IP, for example:192.168.1.10)

3. Explanation of the Paragraph of the Requested Message

   Parameter information(Note: `Optional` is optional):

Key	Type	Values	Description
operator	string	SearchPerson	Query the list based on the `CustomizeID`, `UUID`, or database ID you are looking for.
info	json object		Concrete content
DeviceID	int		ID number of the device to be operated See [6.8.1 Getting Device Information](
SearchType	int (optional)	0~2	0:CustomizeID 1:UUID 2:Database ID Default 0
SearchID	string		CustomizeID、UUID or DatabaseID
Picture	int	0~1	Whether to include image information 0:List without picture information 1:List with picture information

4. Example of Request Message

```
URL of the request:http://192.168.2.10/action/SearchPerson
Content of the request:
{
    "operator": "SearchPerson",
    "info": {
        "DeviceID": 1743726,
        "SearchType": 2,
        "SearchID": "1",
        "Picture": 1
    }
}
```

5. Explanation of the Paragraph of the Reply Message

   Parameter information(Note: `Optional` is optional):

Key	Type	Values	Description
operator	string	SearchPerson	Individual list search
info	json object		Concrete content
DeviceID	int		ID number of the device to be operated See 6.8.1 Getting Device Information
LibID	int		List database ID
PersonType	int	0~1	List Type 0: White list 1: Black list
Name	string		Name Maximum length 32 character length (including terminator)
Gender	int	0~1	Genders 0: male 1: female
Nation	int	Reserve	Reserve
CardType	int		ID Type 0:ID
IdCard	string		ID Number Maximum length is 32 characters (including terminator)
Birthday	string	YYYY-MM-DD	Birthday e.g., 2018-1-1
Telnum	string		Telephone Number Maximum length is 32 characters (including terminator)
Native	string		Native Maximum length is 32 characters (including terminator)
Address	string		Address Maximum length is 72 characters (including terminator)
Notes	string		Notes Maximum length is 64 characters (including terminator)

Key	Type	Values	Description
PersonalPassword	string (optional)		personal password(six-digit access password)
strategyNum	int		Number of access strategy tied to personnel
strategyID	json array	int array	Access strategy ID
MjCardFrom	int	0~3	Wiegand Card Number Generation Method 0: public number 1: automatic generation 2: manual input 3: do not use access card numbers
WGFacilityCode	int (optional)		Facility code When **Wiegand=4 or 5 or 6 or 7** in 6.1 Door Opening Conditions and Output Control Parameters and the **MjCardNo** field together form the Wiegand card number When **Wiegand=0 or 1**, this parameter has no practical meaning. See 3.1.6 Wiegand Access Card Number and Built-in Swipe ID Card Number
MjCardNo	unsigned int		Wiegand access card number (userid) When **Wiegand=4 or 5 or 6 or 7** in 6.1 Door Opening Conditions and Output Control Parameters and the **WGFacilityCode** field together form the Wiegand Card Number When **Wiegand=0 or 1**, **alone** constitutes a Wiegand card number. See 3.1.6 Wiegand Access Card Number and Built-in Swipe ID Card Number
RFIDCard	string (optional)		ID Card Number Maximum length is 18 characters,for built-in card machine type(including terminator) See 3.1.6 Wiegand Access Card Number and Built-in Swipe ID Card Number

Key	Type	Values	Description
Tempvalid	int	0~3	Is it a temporary list 0: Permanent list 1: Temporary list 1 (starting and ending time periods) 2: Temporary list 2 (daily time slots, supported by new database version) 3: Temporary list 3 (number of times valid, supported by new database version) 4: Temporary List 4 (a combination of Temporary List 2 and Temporary List 3, supported by the new database version and after neutral version 9.4)
CustomizeID	unsigned int		User-defined ID You can manage the corresponding personnel information such as modification and deletion by this ID number. Note: You can not use the same ID to add people repeatedly, otherwise it will lead to errors in modifying and deleting operations.
PersonUUID	string		User-defined UUID (not more than 48 bytes) This UUID number can be used to manage personnel information such as modification and deletion. Note: You can not use the same ID to add people repeatedly, otherwise it will lead to errors in operations such as modification and deletion
EffectNumber	int		Effective number of passes through Temporary List 3 or Temporary List 4
ValidBegin	string	YYYY-MM-DDThh:mm:ss	In the case of Temporary List 1 or Temporary List 4, the start time of the entry into force of the Temporary List e.g., 2018-03-12T09:09:20
ValidEnd	string	YYYY-MM-DDThh:mm:ss	In the case of Temporary List 1 or Temporary List 4, the end of the effective date of Temporary List e.g., 2018-03-12T09:10:00
picinfo	string (optional)		Base64 encoded data for face images

Parameter information of the reply message in case of error.

Key	Type	Values	Description
operator	string	SearchPerson	Individual list search
info	JSON Object		Concrete content
code	int		Command execution error code 200-successes,see also 7.27 Individual List Search Error Codes
Result	String	“Ok”/“Fail”	Operating result
Detail	String (optional)		Error message when Result is "Fail"

6. Example of Reply Message

```
{
    "operator": "SearchPerson",
    "info": {
        "DeviceID": 1743726,
        "LibID": 1,
        "PersonType": 0,
        "Name": "test_example",
        "Gender": 0,
        "Nation": 1,
        "CardType": 0,
        "IdCard": " ",
        "Birthday": "2000-01-01",
        "Telnum": " ",
        "Native": " ",
        "Address": " ",
        "Notes": " ",
        "strategyNum": 2,
        "strategyID": [
            1,
            2
        ],
        "MjCardFrom": 0,
        "MjCardNo": 1,
        "RFIDCard": "0",
        "Tempvalid": 0,
        "CustomizeID": 0,
        "PersonUUID": " ",
        "EffectNumber": "0",
        "ValidBegin": "0000-00-00T00:00:00",
        "ValidEnd": "0000-00-00T00:00:00"
```

```
    },
    "picinfo": "........."
}
```

3.6.2 Total Number of List Queries

1. Description

   Returns the total number of eligible lists based on the query criteria.

   Search Description:

   1、Search by time: set the start time and the end time of the search, the name is empty.

   2、Search by name or fuzzy search by name: set the start time and end time of the search to be empty, the name is not empty.

2. API Description

Items	Description
Operator	`SearchPersonNum`
Request URL	`http://<server ipaddr>/action/SearchPersonNum` (Where `<server ipaddr>` is the device IP, for example:192.168.1.10)

3. Explanation of the Paragraph of the Requested Message

   Parameter information(Note: `Optional` is optional):

Key	Type	Values	Description
operator	string	SearchPersonNum	Returns the total number of eligible lists based on the query criteria.
info	json object		Concrete content
DeviceID	int		ID number of the device to be operated See 6.8.1 Getting Device Information
PersonType	int (optional)	0~2	List Type 0:White list 1:Black list 2:All Default 0
BeginTime	string (optional)		Starting point of search Defaults to an empty string
EndTime	string (optional)		End point of search Defaults to an empty string
Name	string (optional)		Name (not more than 32 bytes) Defaults to an empty string
Gender	int (optional)		Genders 0:Male 1:Female 2:All Default 2
Age	string (optional)		Age range:e.g.1-100 Default 0-100
MjCardNo	unsigned int (optional)		Access card number 0:The condition is invalid Default 0

4. Example of Request Message

```
URL of the request:http://192.168.2.10/action/SearchPersonNum
Content of the request:
{
    "operator": "SearchPersonNum",
    "info": {
        "DeviceID": 1743726,
        "PersonType": 0,
        "BeginTime": "2018-06-01T00:00:00",
        "EndTime": "2022-12-21T23:59:59",
```

```
        "Gender": 2,
        "Age": "0-100",
        "MjCardNo": 0,
        "Name": ""
    }
}
```

5. Explanation of the Paragraph of the Reply Message

   Parameter information(Note: `Optional` is optional):

Key	Type	Values	Description
operator	string	SearchPersonNum	Total number of list queries
info	JSON Object		Concrete content
DeviceID	int		ID number of the device to be operated See 6.8.1 Getting Device Information
MaxListsNum	unsigned int		Maximum total number of lists supported by the device
PersonNum	unsigned int		Total number of lists matching the search criteria
code	int		Command execution error code 200-successes,see also 7.28 Total Number of List Queries Error Codes
Result	String	“Ok”/“Fail”	Operating result
Detail	String (optional)		Error message when Result is "Fail"

6. Example of Reply Message

```
{
    "operator": "SearchPersonNum",
    "info": {
        "DeviceID": 1743726,
        "MaxListsNum": 24576,
        "PersonNum": 1
    }
}
```

3.6.3 Multiple List Searches

1. Description

   Search for lists based on the search criteria as well as the starting position to be searched and the number of lists to be searched.

   Search Description:
   1、Search by time: set the start time and the end time of the search, the name is empty.

   2、Search by name or fuzzy search by name: set the start time and end time of the search to be empty, the name is not empty.

2. API Description

Items	Description
Operator	`SearchPersonList`
Request URL	`http://<server ipaddr>/action/SearchPersonList` (Where `<server ipaddr>` is the device IP, for example:192.168.1.10)

3. Explanation of the Paragraph of the Requested Message

   Parameter information(Note: `Optional` is optional):

Key	Type	Values	Description
operator	string	SearchPersonList	Return lists based on starting position and number of queries
info	json object		Concrete content
DeviceID	int		ID number of the device to be operated See 6.8.1 Getting Device Information
PersonType	int (optional)	0~2	List Type 0: White list 1: Black list 2: All Default 0
BeginTime	string (optional)		Starting point of search Defaults to an empty string
EndTime	string (optional)		End point of search Defaults to an empty string
Name	string (optional)		Name (not more than 32 bytes) Defaults to an empty string
Gender	int (optional)	0~2	Genders 0:Male 1:Female 2:All Default 2
Age	string (optional)		Age range:e.g.1-100 Default 0-100
MjCardNo	unsigned int (optional)		Access card number 0:The condition is invalid Default 0
BeginNO	int (optional)		Queries the starting position of the list, i.e., from which person Default 0
RequestCount	int (optional)		The total number of messages returned by the query list, the maximum number of messages returned is 100, and only 100 messages are returned even if the setting exceeds 100. Default 15

Key	Type	Values	Description
Picture	int	0~1	Whether to include image information 0:List without picture information 1:List with picture information Default 0

4. Example of Request Message

```
URL of the request:http://192.168.2.10/action/SearchPersonList
Content of the request:
{
    "operator": "SearchPersonList",
    "info": {
        "DeviceID":1743726,
        "PersonType":0,
        "BeginTime":"2018-06-01T00:00:00",
        "EndTime":"2022-12-21T23:59:59",
        "Gender":2,
        "Age":"0-100",
        "MjCardNo":0,
        "Name":"",
        "BeginNO":0,
        "RequestCount":100,
        "Picture":0
    }
}
```

5. Explanation of the Paragraph of the Reply Message

   Parameter information(Note: `Optional` is optional):

   Returns information about the parameters of the list.

Key	Type	Values	Description
operator	string	SearchPersonList	Multiple list searches
info	json object		Concrete content
DeviceID	int		ID number of the device to be operated See 6.8.1 Getting Device Information
ListNum	int		Number of lists in the reply message
TotalNum	int		Total number of matches in the device list database
List	json array		Specific list information in the reply message
LibID	int		List database ID
PersonType	int	0~1	List Type 0: White list 1: Black list
Name	string		Name Maximum length 32 character length (including terminator)
Gender	int	0~1	Genders 0: male 1: female
Nation	int	Reserve	Reserve
CardType	int		ID Type 0:ID
IdCard	string		ID Number Maximum length is 32 characters (including terminator)
Birthday	string	YYYY-MM-DD	Birthday e.g., 2018-1-1
Telnum	string		Telephone Number Maximum length is 32 characters (including terminator)
Native	string		Native Maximum length is 32 characters (including terminator)

Key	Type	Values	Description
Address	string		Address Maximum length is 72 characters (including terminator)
Notes	string		Notes Maximum length is 64 characters (including terminator)
MjCardFrom	int	0~3	Wiegand Card Number Generation Method 0: public number 1: automatic generation 2: manual input 3: do not use access card numbers
PersonalPassword	string (optional)		Personal password(six-digit access password)
strategyNum	int		Number of access strategy tied to personnel
strategyID	json array		Access strategy ID
WGFacilityCode	int (optional)		Facility code When **Wiegand=4 or 5 or 6 or 7** in 6.1 Door Opening Conditions and Output Control Parameters and the **MjCardNo** field together form the Wiegand card number When **Wiegand=0 or 1**, this parameter has no practical meaning. See 3.1.6 Wiegand Access Card Number and Built-in Swipe ID Card Number
MjCardNo	unsigned int		Wiegand access card number (userid) When **Wiegand=4 or 5 or 6 or 7** in 6.1 Door Opening Conditions and Output Control Parameters and the **WGFacilityCode** field together form the Wiegand Card Number When **Wiegand=0 or 1**, **alone** constitutes a Wiegand card number. See 3.1.6 Wiegand Access Card Number and Built-in Swipe ID Card Number

Key	Type	Values	Description
RFIDCard	string (optional)		ID Card Number Maximum length is 18 characters,for built-in card machine type(including terminator) See 3.1.6 Wiegand Access Card Number and Built-in Swipe ID Card Number
Tempvalid	int	0~3	Is it a temporary list 0: Permanent list 1: Temporary list 1 (starting and ending time periods) 2: Temporary list 2 (daily time slots, supported by new database version) 3: Temporary list 3 (number of times valid, supported by new database version) 4: Temporary List 4 (a combination of Temporary List 2 and Temporary List 3, supported by the new database version and after neutral version 9.4)
CustomizeID	unsigned int		User-defined ID You can manage the corresponding personnel information such as modification and deletion by this ID number. Note: You can not use the same ID to add people repeatedly, otherwise it will lead to errors in modifying and deleting operations.
PersonUUID	string		User-defined UUID (not more than 48 bytes) This UUID number can be used to manage personnel information such as modification and deletion. Note: You can not use the same ID to add people repeatedly, otherwise it will lead to errors in operations such as modification and deletion
EffectNumber	int		Effective number of passes through Temporary List 3 or Temporary List 4

Key	Type	Values	Description
ValidBegin	string	YYYY-MM-DDThh:mm:ss	In the case of Temporary List 1 or Temporary List 4, the start time of the entry into force of the Temporary List . e.g., 2018-03-12T09:09:20
ValidEnd	string	YYYY-MM-DDThh:mm:ss	In the case of Temporary List 1 or Temporary List 4, the end of the effective date of Temporary List . e.g., 2018-03-12T09:10:00
picinfo	string		Base64 encoded data for face images

Parameter information of the reply message in case of error.

Key	Type	Values	Description
operator	string	SearchPersonList	Multiple list searches
info	JSON Object		Concrete content
code	int		Command execution error code 200-successes,see also7.29 Multiple List Searches Error Codes
Result	String	“Ok”/“Fail”	Operating result
Detail	String (optional)		Error message when Result is "Fail"

6. Example of Reply Message

```
{
    "operator": "SearchPersonList",
    "info": {
        "DeviceID": 1743726,
        "Listnum": 3,
        "Totalnum": 64,
        "List": [
            {
                "LibID": 5,
                "PersonType": 0,
                "Name": "example60",
                "Gender": 0,
                "Nation": 1,
                "CardType": 0,
                "IdCard": " ",
                "Birthday": "1970-01-01",
```

```
                "Telnum": " ",
                "Native": " ",
                "Address": " ",
                "Notes": " ",
                "MjCardFrom": 0,
                "strategyNum": 0,
                "strategyID": [],
                "MjCardNo": 0,
                "RFIDCard": "0",
                "Tempvalid": 0,
                "CustomizeID": 0,
                "PersonUUID": " ",
                "ValidBegin": "0000-00-00T00:00:00",
                "ValidEnd": "0000-00-00T00:00:00"
            },
            {
                "LibID": 3,
                "PersonType": 0,
                "Name": "example62",
                "Gender": 1,
                "Nation": 1,
                "CardType": 0,
                "IdCard": " ",
                "Birthday": "1970-01-01",
                "Telnum": " ",
                "Native": " ",
                "Address": " ",
                "Notes": " ",
                "MjCardFrom": 0,
                "strategyNum": 0,
                "strategyID": [],
                "MjCardNo": 1,
                "RFIDCard": "20",
                "Tempvalid": 0,
                "CustomizeID": 0,
                "PersonUUID": " ",
                "ValidBegin": "0000-00-00T00:00:00",
                "ValidEnd": "0000-00-00T00:00:00"
            },
            {
                "LibID": 2,
                "PersonType": 0,
                "Name": "example63",
                "Gender": 1,
                "Nation": 1,
                "CardType": 0,
                "IdCard": " ",
                "Birthday": "1970-01-01",
                "Telnum": " ",
                "Native": " ",
```

```
                "Address": " ",
                "Notes": " ",
                "MjCardFrom": 0,
                "strategyNum": 0,
                "strategyID": [],
                "MjCardNo": 1,
                "RFIDCard": "0",
                "Tempvalid": 0,
                "CustomizeID": 0,
                "PersonUUID": " ",
                "ValidBegin": "0000-00-00T00:00:00",
                "ValidEnd": "0000-00-00T00:00:00"
            }
        ]
    }
}
```

3.7 Control Log Inquiry

Control record query mainly includes querying the total number of control records and querying the control record personnel information.

3.7.1 The Total Number of Control Log Queried

1. Description

   Total number of control log queried.

2. API Description

Items	Description
Operator	`SearchControlNum`
Request URL	`http://<server ipaddr>/action/SearchControlNum` (Where `<server ipaddr>` is the device IP, for example:192.168.1.10)

3. Explanation of the Paragraph of the Requested Message

   Parameter information(Note: `Optional` is optional):

Key	Type	Values	Description
operator	string	SearchControlNum	Total number of control log according to conditions
info	json object		Concrete content
DeviceID	int		ID number of the device to be operated See 6.8.1 Getting Device Information
SearchType	int (optional)	0~1	Search type 0: Search according to the conditions (name, time and other conditions, take the **concatenation** that meets each condition). 1: Search according to the ID + time Default to search according to conditions without setting this field
IdType	int	0~2	**This field is valid when SearchType = 1. Required if SearchType = 1, i.e., search by ID + time.** 0:CustomizeID 1:LibID 2:PersonUUID
CustomizeID	unsigned int **Required if IdType=0**		**This field is valid when SearchType = 1** User-defined ID
LibID	unsigned int **Required if IdType = 1**		**This field is valid when SearchType = 1** List database ID (not the control log database ID, if it is equal to 0 it means this data is a control log generated before this list was deleted)
PersonUUID	string **Required if IdType = 2**		**This field is valid when SearchType = 1** UUID of the list database (no more than 48 bytes)

Key	Type	Values	Description
Name	string (optional)		**This field is valid when SearchType = 0** Name (according to the name search, the default does not judge the conditions of the conditions of the search); if you are looking for remote door opening control records, please enter “Remote door open”
MjCardNo	unsigned int (optional)		**This field is valid when SearchType = 0** Access card number Default 0
Gender	int (optional)	0~2	**This field is valid when SearchType = 0** Genders 0:Male 1:Female 2:All Not setting this field defaults to 2
OpenDoor	int (optional)	0~2	0: All 1: Open door 2: Rejection , Not setting this field defaults to 0
BeginTime	string		Search starting time , **mandatory field**
EndTime	string		Search ending time , **mandatory field**
Age	string (optional)		Age range e.g.1-100 , Not setting this field defaults to 1-200

4. Example of Request Message

```
URL of the request:http://192.168.2.10/action/SearchControlNum
Content of the request:

Search according to the ID + time
{
    "operator": "SearchControlNum",
    "info": {
        "DeviceID": 1743726,
        "SearchType": 1,
        "IdType": 1,
        "LibID": 1,
        "BeginTime": "2020-01-01T00:00:00",
        "EndTime": "2022-12-21T23:09:00"
    }
}

Search according to the conditions (name, time and genders):
{
```

```
    "operator": "SearchControlNum",
    "info": {
        "DeviceID": 1743726,
        "SearchType": 0,
        "Name": "test",
        "Gender": 2,
        "BeginTime": "2020-01-01T00:00:00",
        "EndTime": "2023-12-21T23:09:00"
    }
}
```

5. Explanation of the Paragraph of the Reply Message

   Parameter information(Note: `Optional` is optional):

Key	Type	Values	Description
operator	string	SearchControlNum	Query the total number of control log according to the conditions
info	json object		Concrete content
DeviceID	int		ID number of the device to be operated See 6.8.1 Getting Device Information
ControlNum	int		Returns the total number of control log by time condition
code	int		Command execution error code 200-successes,see also7.30 The Total Number of Control Log Queried Error Codes
Result	String	“Ok”/“Fail”	Operating result
Detail	String (optional)		Error message when Result is "Fail"

6. Example of Reply Message

```
{
    "operator": "SearchControlNum",
    "info": {
        "DeviceID": 1743726,
        "ControlNum": 4
    }
}
```

3.7.2 Control Log Personnel Information Queries

1. Description

   Queries control log personnel information based on search criteria.

2. API Description

Items	Description
Operator	`SearchControl`
Request URL	`http://<server ipaddr>/action/SearchControl` (Where `<server ipaddr>` is the device IP, for example:192.168.1.10)

3. Explanation of the Paragraph of the Requested Message

   Parameter information(Note: `Optional` is optional):

Key	Type	Values	Description
operator	string	SearchControl	Queries control log personnel information based on search criteria.
info	json object		Concrete content
DeviceID	int		ID number of the device to be operated See 6.8.1 Getting Device Information
SearchType	int (optional)	0~1	Search type 0: Search according to the conditions (name, time and other conditions, take the **concatenation** that meets each condition). 1: Search according to the ID + time Default to search according to conditions without setting this field
IdType	int	0~2	**This field is valid when SearchType = 1. Required if SearchType = 1, i.e., search by ID + time.** 0:CustomizeID 1:LibID 2:PersonUUID
CustomizeID	unsigned int **Must be filled in when IdType=0**		**This field is valid when SearchType = 1** User-defined ID
LibID	unsigned int **Must be filled in when IdType=1**		**This field is valid when SearchType = 1** List database ID (not the control log database ID, if it is equal to 0 it means this data is a control log generated before this list was deleted)
PersonUUID	string **Must be filled in when IdType=2**		**This field is valid when SearchType = 1** UUID of the list database (no more than 48 bytes)

Key	Type	Values	Description
Name	string (optional)		**This field is valid when SearchType = 0** Name (according to the name search, the default does not judge the conditions of the conditions of the search); if you are looking for remote door opening control records, please enter “Remote door open”
MjCardNo	unsigned int (optional)		**This field is valid when SearchType = 0** Access card number Default 0
Gender	int (optional)	0~2	**This field is valid when SearchType = 0** Genders 0:Male 1:Female 2:All Not setting this field defaults to 2
OpenDoor	int (optional)	0~2	0: All 1: Open door 2: Rejection Not setting this field defaults to 0
BeginTime	string		Search starting time，**mandatory field**
EndTime	string		Search ending time，**mandatory field**
Ascending	int (optional)	0~1(Default 0)	Search results in ascending or descending order 0:ascending (results moved backward from search start time) 1:descending (results moved forward from search end time)
Age	string (optional)		Age range e.g.1-100，Not setting this field defaults to 1-200
BeginNO	int (optional)		Starting number of the return No setting defaults to start at 0
RequestCount	int (optional)	0~100	Number of log returned Maximum return of 100 items, no setting default maximum return of 20 items
Picture	int (optional)	0~3	Whether to include images 0:Not include images 1:Include registered images 2:Include captured images 3:Include registered+captured images Default is 0

4. Example of Request Message

```
URL of the request:http://192.168.2.10/action/SearchControl
Content of the request:

Search according to the conditions:
{
    "operator": "SearchControl",
    "info": {
        "DeviceID": 1743726,
        "SearchType": 0,
        "Name":"test_example",
        "MjCardNo":1,
        "Gender": 1,
        "Ascending": 1,
        "BeginTime": "2018-11-13T00:00:00",
        "EndTime": "2023-12-21T23:59:59",
        "Picture": 0,
        "RequestCount": 20
    }
}

Search according to the ID + time
{
    "operator": "SearchControl",
    "info": {
        "DeviceID": 1743726,
        "SearchType": 1,
        "IdType":1,
        "LibID": 67,
        "Ascending": 1,
        "BeginTime": "2018-11-13T00:00:00",
        "EndTime": "2023-12-21T23:59:59",
        "BeginNO": 1,
        "Picture": 0,
        "RequestCount": 20
    }
}
```

5. Explanation of the Paragraph of the Reply Message

Key	Type	Values	Description
operator	string	SearchControl	Queries control log personnel information based on search criteria.
info	json object		Concrete content
DeviceID	int		ID number of the device to be operated
ControlNum	int		Total number of eligible logs in the control log database
TotalNum	int	0~100	Return the number of control logs, the maximum return 100, do not set the default maximum return 20
SearchInfo	json array		Back to search results (list)
LibID	int		Database ID of the control log
VerfyType	int	1:Universal model 1~3 or 21~22 or 24~25 or 27 ; 2:Special model: **Support for mask detection** or (|) operation 0x100 indicates a pass with mask detection; **Supports body temperature detection** or (|) operation 0x200 indicates a pass with temperature detection ;	Authentication Type 1: Whitelist Verification 2: ID Verification 3: Whitelist + ID Verification 21:RF Card Verification(**Built-in card swiping model**) 22:RF Card Verification + Whitelist Verification(**Built-in card swiping model**) 24:Wergand Card Verification 25. Wigand Card + Whitelist Verification 27:HTTP Remote Door Opening

Key	Type	Values	Description
		Support mask + body temperature detection Or (|) operation 0x300 means with mask + temperature detection pass; e.g.Masks + whitelist verification （0x101） Body Temperature + Whitelist Verification （0x201） Mask + Body Temperature + Wiegand Swipe + Whitelist Verification （0x319）	
PersonType	int		List Type 0: White list 1: Black list
Name	string		Name Maximum length 32 character length (including terminator)
Gender	int	0~2	Genders 0: Male 1: Female 2: All
Nation	int		Nation
IdCard	string		ID Number
Birthday	string		Birthday

Key	Type	Values	Description
Telnum	string		Telephone Number
Native	string		Native
Address	string		Address
CustomizeID	unsigned int		User-defined ID
PersonUUID	string		User-defined Uuid
MjCardFrom	int	0~3	Wiegand Card Number Generation Method 0: public number 1: automatic generation 2: manual input 3: do not use access card numbers
MjCardNo	unsigned int		Wiegand access card number (userid) When **Wiegand=4 or 5 or 6 or 7** in 6.1 Door Opening Conditions and Output Control Parameters and the **WGFacilityCode** field together form the Wiegand Card Number When **Wiegand=0 or 1**, **alone** constitutes a Wiegand card number. See 3.1.6 Wiegand Access Card Number and Built-in Swipe ID Card Number
WGFacilityCode	int (optional)		Facility code When **Wiegand=4 or 5 or 6 or 7** in 6.1 Door Opening Conditions and Output Control Parameters and the **MjCardNo** field together form the Wiegand card number When **Wiegand=0 or 1**, this parameter has no practical meaning. See 3.1.6 Wiegand Access Card Number and Built-in Swipe ID Card Number
RFIDCard	string (optional)		ID Card Number Maximum length is 18 characters,for built-in card machine type(including terminator). See 3.1.6 Wiegand Access Card Number and Built-in Swipe ID Card Number

Key	Type	Values	Description
Temperature	double (optional)	37.20	Detecting face temperature in real time, returning to detect temperature only when FaceMaskTPTMode=1, 4, 5 mode in 6.15 Temperature Parameters (**Temperature detection machine version support**)
TemperatureAlarm	int (optional)	0~1	Real-time detection of whether the face temperature exceeds the threshold (**Temperature detection machine version support**) 0:not exceeded 1:exceeded
TemperatureMode	int (optional)	0~1	Temperature Display Mode 0: Celsius 1: Fahrenheit
Time	string		Certification Passing Time
VerifyStatus	string	0~2	Personnel access status 0:None 1:Allowed 2:Denied
Similarity	float		Similarity
RegPicinfo	string (optional)		Register the base64 encoding of the image
SnapPicinfo	string (optional)		Base64 encoding of captured images
RemoteOpenDoor	int (optional)	1	If it is an open door control record for a call to 5.3 Remote Open Door, return this identifier
code	int		Command execution error code:200-successes,see also 7.31 Control Log Personnel Information Queries Error Codes
Result	String	“Ok”/“Fail”	Operating result
Detail	String (optional)		Error message when Result is "Fail"

6. Example of Reply Message

```
Search according to the conditions:（SearchType = 0）
{
    "operator": "SearchControl",
```

```
    "info": {
        "DeviceID": 1743726,
        "ControlNum": 4,
        "TotalNum": 4,
        "SearchInfo": [
            {
                "LibID": 1,
                "VerfyType": 1,
                "PersonType": 0,
                "Name": "test_example",
                "Gender": 0,
                "Nation": 1,
                "IdCard": " ",
                "Birthday": "2000-01-01",
                "Telnum": " ",
                "Native": " ",
                "Address": " ",
                "CustomizeID": 0,
                "PersonUUID": " ",
                "MjCardFrom": 0,
                "MjCardNo": 1,
                "RFIDCard": "0",
                "VerifyStatus": 1,
                "Similarity": 95.2,
                "Time": "2022-12-20T09:51:00"
            },
            ...
            ...
            {
                "LibID": 1,
                "VerfyType": 1,
                "PersonType": 0,
                "Name": "test_example",
                "Gender": 0,
                "Nation": 1,
                "IdCard": " ",
                "Birthday": "2000-01-01",
                "Telnum": " ",
                "Native": " ",
                "Address": " ",
                "CustomizeID": 0,
                "PersonUUID": " ",
                "MjCardFrom": 0,
                "MjCardNo": 1,
                "RFIDCard": "0",
                "VerifyStatus": 1,
                "Similarity": 93.6,
                "Time": "2022-12-20T16:54:41"
            }
        ]
```

```
    }
}

Search according to the ID + time(SearchType = 1)
{
    "operator": "SearchControl",
    "info": {
        "DeviceID": 1743726,
        "LibID": 67,
        "VerfyType": 1,
        "PersonType": 0,
        "Name": "xtc",
        "Gender": 0,
        "Nation": 1,
        "IdCard": " ",
        "Birthday": "2000-01-01",
        "Telnum": " ",
        "Native": " ",
        "Address": " ",
        "CustomizeID": 0,
        "PersonUUID": " ",
        "MjCardFrom": 0,
        "MjCardNo": 1,
        "RFIDCard": "0",
        "TotalNum": 2,
        "SearchInfo": [
            {
                "VerifyStatus": 1,
                "Similarity": 94.8,
                "Time": "2022-12-30T18:05:00"
            },
            {
                "VerifyStatus": 1,
                "Similarity": 96.8,
                "Time": "2022-12-30T18:04:21"
            }
        ]
    }
}
```

3.8 Capture Log Inquiry

Capture log query is about strangers capture log query, if it is the pass comparison of the personnel information is not stored in the capture log; capture log query mainly includes the total number of capture log query and capture log information query.

3.8.1 Search for the Total Number of Captured Logs

1. Description

   Queries the total number of captured logs according to time conditions.

2. API Description

Items	Description
Operator	`SearchCaptureNum`
Request URL	`http://<server ipaddr>/action/SearchCaptureNum` (Where `<server ipaddr>` is the device IP, for example:192.168.1.10)

3. Explanation of the Paragraph of the Requested Message

   Parameter information(Note: `Optional` is optional):

Key	Type	Values	Description
operator	string	SearchCaptureNum	Queries the total number of captured logs according to time conditions.
info	json object		Concrete content
DeviceID	int		ID number of the device to be operated See 6.8.1 Getting Device Information
BeginTime	string		Search starting time，**mandatory field**
EndTime	string		Search ending time，**mandatory field**

4. Example of Request Message

```
URL of the request:http://192.168.2.10/action/SearchCaptureNum
Content of the request:
{
    "operator": "SearchCaptureNum",
    "info": {
        "DeviceID": 1743726,
        "BeginTime": "2020-01-01T00:00:00",
        "EndTime": "2022-12-21T23:09:00"
    }
}
```

5. Explanation of the Paragraph of the Reply Message

   Returned parameter information.

Key	Type	Values	Description
operator	string	SearchCaptureNum	Queries the total number of log captured according to time conditions.
info	json object		Concrete content
DeviceID	int		ID number of the device to be operated
CaptureNum	int		Returns the total number of captured logs by time condition
code	int		Command execution error code:200-successes,see also7.32 Search for the Total Number of Captured Logs Error Codes
Result	String	“Ok”/“Fail”	Operating result
Detail	String (optional)		Error message when Result is "Fail"

6. Example of Reply Message

```
{
    "operator": "SearchCaptureNum",
    "info": {
        "DeviceID": 1743726,
        "CaptureNum": 14
    }
}
```

3.8.2 Capture Log Personnel Information Query

1. Description

   Query the information of the person who caught the log according to the conditions

2. API Description

Items	Description
Operator	`SearchCapture`
Request URL	`http://<server ipaddr>/action/SearchCapture` (Where `<server ipaddr>` is the device IP, for example:192.168.1.10)

3. Explanation of the Paragraph of the Requested Message

   Parameter information(Note: `Optional` is optional):

Key	Type	Values	Description
operator	string	SearchCapture	Query the information of the person who caught the log according to the conditions
info	json object		Concrete content
DeviceID	int		ID number of the device to be operated See 6.8.1 Getting Device Information
BeginTime	string		Search starting time，**mandatory field**
EndTime	string		Search ending time，**mandatory field**
Ascending	int (optional)	0~1(Default 0)	Search results in ascending or descending order 0:ascending (results moved backward from search start time) 1:descending (results moved forward from search end time)
BeginNO	int (optional)		Starting number of the return No setting defaults to start at 0
RequestCount	int (optional)	0~100	Number of log returned Maximum return of 100 items, no setting default maximum return of 15 items

4. Example of Request Message

```
URL of the request:http://192.168.2.10/action/SearchCapture
Content of the request:

Search by time:
{
    "operator": "SearchCapture",
    "info": {
        "DeviceID": 1743726,
        "BeginTime": "2020-01-13T00:00:00",
        "EndTime": "2022-12-21T23:59:59"
    }
}

Limit the number of captured logs returned:
{
    "operator": "SearchCapture",
    "info": {
        "DeviceID": 1743726,
        "BeginTime": "2020-01-01T00:00:00",
```

```
        "EndTime": "2022-12-21T23:59:59",
        "RequestCount": 10
    }
}

Limit the number of captured logs returned and the starting position:
{
    "operator": "SearchCapture",
    "info": {
        "DeviceID": 1743726,
        "BeginTime": "2020-01-01T00:00:00",
        "EndTime": "2022-12-21T23:59:59",
        "BeginNO": 5,
        "RequestCount": 10
    }
}
```

5. Explanation of the Paragraph of the Reply Message

Key	Type	Values	Description
operator	string	SearchCapture	Query the information of the person who caught the log according to the conditions
info	json object		Concrete content
DeviceID	int		ID number of the device to be operated
CaptureNum	int		Returns the total number of captured logs by time conditio
Listnum	int (optional)		Number of captured logs returned by querying according to conditions
List	Array		Capture log information array
LibSnapID	int		Capture database ID of the returned capture logs information
CreateTime	string	YYYY-MM-DDThh:mm:ss	Recording time
Temperature	double (optional)	37.20	Detecting face temperature in real time, returning to detect temperature only when FaceMaskTPTMode=1, 4, 5 mode in 6.15 Temperature Parameters (**Temperature detection machine version support**)
TemperatureAlarm	int (optional)	0~1	Real-time detection of whether the face temperature exceeds the threshold (**Temperature detection machine version support**) 0:not exceeded 1:exceeded
TemperatureMode	int (optional)	0~1	Temperature Display Mode 0: Celsius 1: Fahrenheit
SnapPicinfo	string		Base64 encoding of captured images
code	int		Command execution error code 200-successes,see also7.33 Capture Log Personnel Information Query Error Codes
Result	String	“Ok”/“Fail”	Operating result
Detail	String (optional)		Error message when Result is "Fail"

6. Example of Reply Message

```
{
    "operator": "SearchCapture",
    "info": {
        "DeviceID": 1305433,
        "Listnum": 15,
        "List": [
            {
                "LibSnapID": 3303,
                "CreateTime": "2020-01-13T20:02:43",
                "SnapPicinfo": "......
        },
            {
                "LibSnapID": 3302,
                "CreateTime": "2020-01-13T19:36:06",
                "SnapPicinfo": "......
        },
            {
                "LibSnapID": 3301,
                "CreateTime": "2020-01-13T18:34:15",
                "SnapPicinfo": "......
        },
            ......
            {
                "LibSnapID": 3289,
                "CreateTime": "2020-01-13T16:31:26",
                "SnapPicinfo": "......
        }
        ]
    }
}
```

3.9 Access Strategy

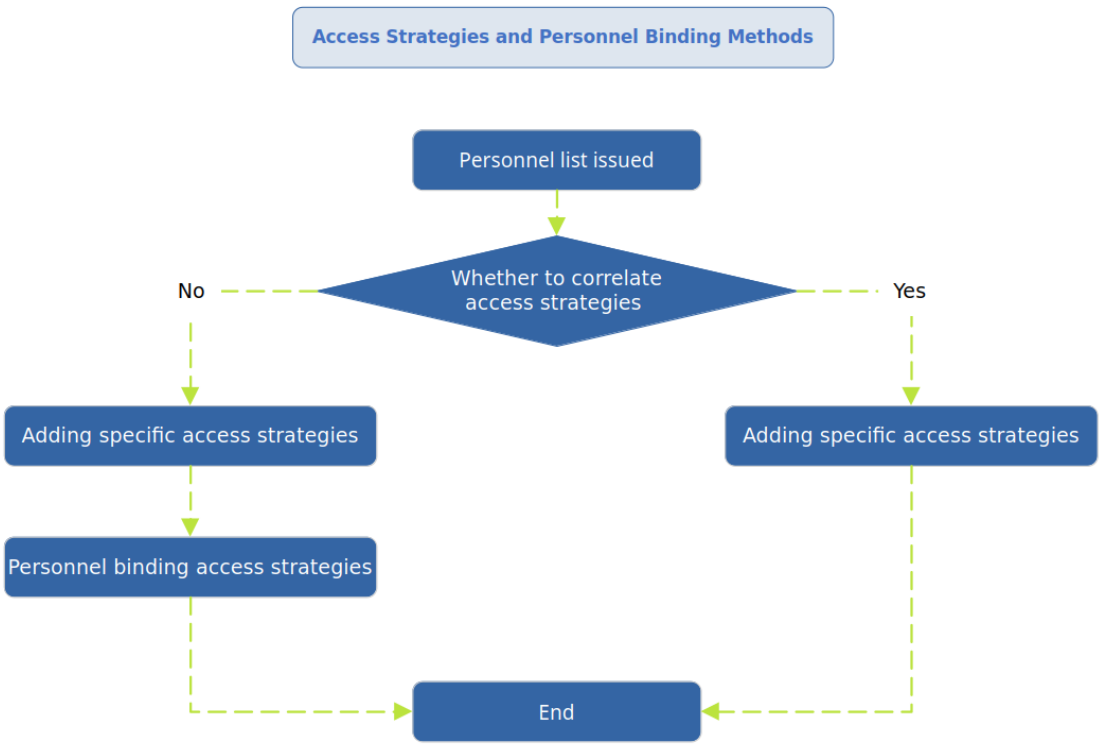
Since the access time rules of the temporary list in the personnel list type can't satisfy the multiple time periods to control the access of personnel, this interface is developed to satisfy the time periods to control the access of personnel through the access strategy.

3.9.Access Strategy Rules

1. Each strategy is identified by a strategy ID (hereinafter referred to as `strategyID`), and each strategy consists of two parts, the regular strategy time (Monday to Sunday) and the holiday strategy time. 2.
2. Regular strategy time can be set for more than one time slot per day, for example: Monday can be set for more than one time slot
3. Holiday strategy time only supports adding one corresponding time slot, if there are more than one time slots for the same holiday, then you need to add more holidays.

4. The access strategy for the passable time period settings, if it is prohibited to pass the time, you need to take the passable time period of the opposite time, assuming that the passable time period of 9:00-18:00, if you want to prohibit the passage of time is also 9:00-18:00, issued to the face of the machine can be set for the passable time period for the 00:00-08:59 and 18:00-23:59.
5. The passable time period for the passable time period can be set. The personnel outside of the access strategy time policy will be restored to the passable state by default, assuming the expiration date of passable 2020-12-29T23:59:59, then the personnel bound to this strategy after this time will be restored to the passable state.
6. Restoring the factory operation by checking the Restore Personnel List will delete all the information of the access strategy.
7. A personnel list can be bound to more than one access strategy, and the final passable The time period is the concatenation of the time periods of multiple access strategy.
8. Try to use permanent lists for the issued list types, and then bind the access strategy, do not issue temporary lists and use the access strategy at the same time.

3.9.2 Access Strategies and Personnel Binding Methods



Here are two ways to bind a access strategy to a person, as shown above.

Mode 1: When issuing the personnel list, directly bring the corresponding associated access strategy information, and then add the corresponding access strategy information.

Mode 2: The list of personnel is issued without the corresponding associated access strategy information, then add the corresponding access strategy information, and finally bind the list of personnel to the corresponding access strategy through the personnel binding access strategy interface.

3.9.3 Add/Modify Access Strategies

1. Description

   Each access strategy will be uniquely identified with a `strategyID`, and a non-existing `strategyID` will be considered as adding a corresponding access strategy, while an existing `strategyID` will be considered as modifying the corresponding access strategy. Assuming that Wednesday is not available all day, the `wednesday` keyword will not be passed, and All-in-One machines passing the corresponding keyword without a time will set the default pass time.

2. API Description

Items	Description
Operator	`AddAccessStrategy`
Request URL	`http://<server ipaddr>/action/AddAccessStrategy` (Where `<server ipaddr>` is the device IP, for example:192.168.1.10)

3. Explanation of the Paragraph of the Requested Message

   Parameter information(Note: `Optional` is optional):

Key	Type	Values	Description
operator		AddAccessStrategy	Add/modify access strategies
info			Specific Information Structures
strategyID	Int	1~N	Access strategy ID
strategyName	String (optional)		Access strategy name (64 bytes including terminator)
accessNumLimit	Int (optional)		Limitations on the number of passes 0:Unlimited number of passes (default) 1:Restrictions on the number of passes
allowCnt	Int (optional)		Number of passe Valid when accessNumLimit=1, required
startDate	String	YYYY-MM-DDThh:mm:ss (2020-01-01T00:00:00)	Effective time of access strategy
endDate	String	YYYY-MM-DDThh:mm:ss (2020-12-29T23:59:59)	Access strategy expiration time
monday	Array (optional)		Monday access strategy keywords
tuesday	Array (optional)		Tuesday access strategy keywords
wednesday	Array (optional)		Wednesday access strategy keywords
thursday	Array (optional)		Thursday access strategy keywords
friday	Array (optional)		Friday access strategy keywords
saturday	Array (optional)		Saturday access strategy keywords
sunday	Array (optional)		Sunday access strategy keywords
startTime	String	hh:mm:ss(08:00:00)	Multi-time period start time Default is (00:00:00)

Key	Type	Values	Description
endTime	String	hh:mm:ss(23:59:59)	Multi-time period end time Default is (23:59:59)
holidayInfo	String (optional)		Holiday information keywords
holidayStartDate	String (optional)	YYYY-MM-DDThh:mm:ss (2020-10-1T00:00:00)	Holiday start time
holidayEndDate	String (optional)	YYYY-MM-DDThh:mm:ss (2020-10-1T23:59:59)	Holiday end time
holidayPeriod	Array		Holiday multi-time period keywords (Currently one holiday corresponds to one time period)

4. Example of Request Message

```
URL of the request:http://172.168.5.106/action/AddAccessStrategy
Content of the request:
{
    "operator": "AddAccessStrategy",
    "info": {
        "strategyID": 1,
        "strategyName": "test1",
        "accessNumLimit": 0,
        "allowCnt": 0,
        "startDate": "2020-01-01T08:12:10",
        "endDate": "2023-12-09T08:08:12",
        "monday": [
            {
                "endTime": "01:09:00",
                "startTime": "00:00:00"
            },
            {
                "endTime": "03:41:00",
                "startTime": "02:02:00"
            }
        ],
        "tuesday": [
            {
                "endTime": "08:59:00",
                "startTime": "00:00:00"
```

```
            },
            {
                "endTime": "10:20:00",
                "startTime": "10:05:00"
            }
        ],
        "wednesday": [
            {
                "endTime": "10:59:00",
                "startTime": "00:00:00"
            }
        ],
        "thursday": [
            {
                "endTime": "21:59:00",
                "startTime": "00:00:00"
            }
        ],
        "friday": [
            {
                "endTime": "08:59:00",
                "startTime": "04:00:00"
            }
        ],
        "sunday": [
            {
                "endTime": "15:46:00",
                "startTime": "00:00:00"
            }
        ],
        "holidayInfo": [
            {
                "holidayStartDate": "2020-10-10T00:00:00",
                "holidayEndDate": "2020-10-10T23:00:00",
                "holidayPeriod": [
                    {
                        "endTime": "12:43:00",
                        "startTime": "01:00:00"
                    }
                ]
            },
            {
                "holidayStartDate": "2020-10-10T00:00:00",
                "holidayEndDate": "2020-10-10T23:59:59",
                "holidayPeriod": [
                    {
                        "endTime": "23:45:07",
                        "startTime": "17:00:00"
                    }
                ]
```

```
            }
        ]
    }
}
```

5. Explanation of the Paragraph of the Reply Message

   Parameter information(Note: `Optional` is optional):

Key	Type	Values	Description
operator	string	AddAccessStrategy	Add/modify access strategies
info	JSON Object		Concrete content
code	int		Command execution error code: 200-successes,see also7.46 Add/Modify Access Strategies Error Codes
Result	String	“Ok”/“Fail”	Operating result
Detail	String (optional)		Error message when Result is "Fail"

6. Example of Reply Message

```
{
    "operator": "AddAccessStrategy",
    "code": 200,
    "info": {
        "Result": "Ok"
    }
}
```

3.9.4 Delete Access Strategies

1. Description

   Deletes the access strategies corresponding to the specified `strategyID` and unbinds the deleted access strategies from the corresponding personnel.

2. API Description

Items	Description
Operator	`DelAccessStrategy`
Request URL	`http://<server ipaddr>/action/DelAccessStrategy` (Where `<server ipaddr>` is the device IP, for example:192.168.1.10)

3. Explanation of the Paragraph of the Requested Message

Key	Type	Values	Description
operator		DelAccessStrategy	Delete access strategies
info			Concrete content
strategyID	json array	1~N	Access strategy ID

4. Example of Request Message

```
URL of the request:http://172.168.5.106/action/DelAccessStrategy
Content of the request:
{
    "operator":"DelAccessStrategy",
    "info":
    {
        "strategyID":[1,2]
    }
}
```

5. Explanation of the Paragraph of the Reply Message

   Parameter information(Note: `Optional` is optional):

Key	Type	Values	Description
operator	string	DelAccessStrategy	Delete access strategies
info	JSON Object		Concrete content
code	int		Command execution error code 200-successes,see also7.47 Delete Access Strategies Error Codes
delAccessStrategyErrNum	string	0~N	Number of failed deletions of access strategies
delAccessStrategyErrInfo	Array		Delete access strategies failure message keyword
delAccessStrategySucNum	string	0~N	Number of successful access strategies deletions
delAccessStrategySucInfo	Array		Delete access strategies success message keyword
JsonNum	string	0-N	strategyID index in Json array
strategyID	string	0-N	Access strategy ID
errcode	string		Error code
Result	String	“Ok”/“Fail”	Operating result
Detail	String (optional)		Error message when Result is "Fail"

6. Example of Reply Message

```
{
    "operator": "DelAccessStrategy",
    "code": 200,
    "info": {
        "DeviceID": 1787156,
        "delAccessStrategyErrNum": "0",
        "delAccessStrategySucNum": "2",
        "delAccessStrategyErrInfo": [],
        "delAccessStrategySucInfo": [
            {
                "JsonNum": "0",
                "strategyID": "1"
            },
```

```
            {
                "JsonNum": "1",
                "strategyID": "2"
            }
        ],
        "Result": "OK"
    }
}
```

3.9.5 Query All Access Strategies IDs and Names

1. Description

   Queries all access strategies IDs in the device along with their names.

2. API Description

Items	Description
Operator	`QueryAllStrategyID`
Request URL	`http://<server ipaddr>/action/QueryAllStrategyID` (Where `<server ipaddr>` is the device IP, for example:192.168.1.10)

3. Explanation of the Paragraph of the Requested Message

   Parameter information(Note: `Optional` is optional)

Key	Type	Values	Description
operator		QueryAllStrategyID	Query all access strategies IDs and names

4. Example of Request Message

```
URL of the request:http://172.168.5.106/action/QueryAllStrategyID
Content of the request:
{
    "operator":"QueryAllStrategyID"
}
```

5. Explanation of the Paragraph of the Reply Message

   Parameter information(Note: `Optional` is optional):

Key	Type	Values	Description
operator	string	QueryAllStrategyID	Query all access strategies IDs and names
code	int		Command execution error code 200-successes,see also7.48 Query All Access Strategy IDs and Names Error Codes
info	JSON Object		Concrete content
strategyNum	int	0~N	Number of access strategies
strategyInfo	Array		Access strategy information keyword (valid when StrategyNum>0)
strategyID	String	1~N	Access strategy ID
strategyName	String		Access strategy name
Result	String	“Ok”/“Fail”	Operating result
Detail	String (optional)		Error message when Result is "Fail"

6. Example of Reply Message

```
{
    "operator": "QueryAllStrategyID",
    "code": 200,
    "info": {
        "DeviceID": 1787156,
        "strategyNum": 2,
        "strategyInfo": [
            {
                "strategyID": "2",
                "strategyName": "test2"
            },
            {
                "strategyID": "1",
                "strategyName": "test1"
            }
        ],
        "Result": "OK"
    }
}
```

3.9.6 Query Access Strategy Details by Access Strategy ID

1. Description

   Query access strategy details by access strategy ID.

2. API Description

Items	Description
Operator	`AccessStrategyIDQueryStrategy`
Request URL	`http://<server ipaddr>/action/AccessStrategyIDQueryStrategy` (Where `<server ipaddr>` is the device IP, for example:192.168.1.10)

3. Explanation of the Paragraph of the Requested Message

Key	Type	Values	Description
operator		AccessStrategyIDQueryStrategy	Query access strategy details by access strategy ID
info			Specific Information Structures
strategyID	json array	1~N	Access strategy ID

4. Example of Request Message

```
URL of the request:http://172.168.5.106/action/AccessStrategyIDQueryStrategy
Content of the request:
{
    "operator":"AccessStrategyIDQueryStrategy",
    "info":{
        "strategyID":[1,2]
    }
}
```

5. Explanation of the Paragraph of the Reply Message

   Parameter information(Note: `Optional` is optional):

Key	Type	Values	Description
operator	string	AccessStrategyIDQueryStrategy	Query access strategy details by access strategy ID
info	JSON Object		Concrete content
code	int		Command execution error code 200-successes,see also 7.49 Query Access Strategy Details by Access Strategy ID Error Codes
Result	String	“Ok”/“Fail”	Operating result
Detail	String (optional)		Error message when Result is "Fail"
strategyNum	int	0~N	Number of access strategies
strategyInfo	Array		Access strategy information keyword (valid when strategyNum > 0)
strategyID	Int	1~N	Access strategy ID
strategyName	String (optional)		Access strategy name (64 bytes including terminator)
accessNumLimit	Int (optional)		Limitations on the number of passes 0:Unlimited number of passes (default) 1:Restrictions on the number of passes
allowCnt	Int (optional)		Number of passe Valid when accessNumLimit=1, required
startDate	String	YYYY-MM-DDThh:mm:ss (2020-01-01T00:00:00)	Effective time of access strategy
endDate	String	YYYY-MM-DDThh:mm:ss (2020-12-29T23:59:59)	Access strategy expiration time

Key	Type	Values	Description
monday	Array (optional)		Monday access strategy keywords
tuesday	Array (optional)		Tuesday access strategy keywords
wednesday	Array (optional)		Wednesday access strategy keywords
thursday	Array (optional)		Thursday access strategy keywords
friday	Array (optional)		Friday access strategy keywords
saturday	Array (optional)		Saturday access strategy keywords
sunday	Array (optional)		Sunday access strategy keywords
startTime	String	hh:mm:ss(08:00:00)	Multi-time period start time Default is (00:00:00)
endTime	String	hh:mm:ss(23:59:59)	Multi-time period end time Default is (23:59:59)
holidayInfo	String (optional)		Holiday information keywords
holidayStartDate	String (optional)	YYYY-MM-DDThh:mm:ss (2020-10-1T00:00:00)	Holiday start time
holidayEndDate	String (optional)	YYYY-MM-DDThh:mm:ss (2020-10-1T23:59:59)	Holiday end time
holidayPeriod	Array		Holiday multi-time period keywords (Currently one holiday corresponds to one time period)

Other fields describing the details of the access strategy can be found in 3.9.3 Add/Modify Access Strategies .

6. Example of Reply Message

```
{
    "operator": "AccessStrategyIDQueryStrategy",
```

```
    "code": 200,
    "info": {
        "DeviceID": 1787156,
        "strategyNum": 6,
        "strategyInfo": [
            {
                "strategyID": 1,
                "strategyName": "test1",
                "accessNumLimit": 0,
                "allowCnt": 0,
                "startDate": "2020-01-01T08:12:10",
                "endDate": "2023-12-09T08:08:12",
                "monday": [
                    {
                        "startTime": "00:00:00",
                        "endTime": "01:09:00"
                    },
                    {
                        "startTime": "02:02:00",
                        "endTime": "03:41:00"
                    }
                ],
                "wednesday": [
                    {
                        "startTime": "00:00:00",
                        "endTime": "10:59:00"
                    }
                ],
                "holidayInfo": [
                    {
                        "holidayStartDate": "2020-10-10T00:00:00",
                        "holidayEndDate": "2020-10-10T23:00:00",
                        "holidayPeriod": [
                            {
                                "startTime": "01:00:00",
                                "endTime": "12:43:00"
                            }
                        ]
                    },
                    {
                        "holidayStartDate": "2020-10-10T00:00:00",
                        "holidayEndDate": "2020-10-10T23:59:59",
                        "holidayPeriod": [
                            {
                                "startTime": "17:00:00",
                                "endTime": "23:45:07"
                            }
                        ]
                    }
                ]
```

```
        },
        {
            "strategyID": 2,
            "strategyName": "test2",
            "accessNumLimit": 0,
            "allowCnt": 0,
            "startDate": "2020-01-01T08:12:10",
            "endDate": "2023-12-09T08:08:12",
            "monday": [
                {
                    "startTime": "00:00:00",
                    "endTime": "01:09:00"
                },
                {
                    "startTime": "02:02:00",
                    "endTime": "03:41:00"
                }
            ],
            "tuesday": [
                {
                    "startTime": "00:00:00",
                    "endTime": "08:59:00"
                },
                {
                    "startTime": "10:05:00",
                    "endTime": "10:20:00"
                }
            ],
            "holidayInfo": [
                {
                    "holidayStartDate": "2020-10-10T00:00:00",
                    "holidayEndDate": "2020-10-10T23:00:00",
                    "holidayPeriod": [
                        {
                            "startTime": "01:00:00",
                            "endTime": "12:43:00"
                        }
                    ]
                },
                {
                    "holidayStartDate": "2020-10-10T00:00:00",
                    "holidayEndDate": "2020-10-10T23:00:00",
                    "holidayPeriod": [
                        {
                            "startTime": "00:00:00",
                            "endTime": "18:43:00"
                        }
                    ]
                },
                {
```

```
                        "holidayStartDate": "2020-10-10T00:00:00",
                        "holidayEndDate": "2020-10-10T23:59:59",
                        "holidayPeriod": [
                            {
                                "startTime": "17:00:00",
                                "endTime": "23:45:07"
                            }
                        ]
                    }
                ]
            }
        ],
        "Result": "OK"
    }
}
```

3.9.7 Query All Associated Users by Access Strategy ID

1. Description

   Query all associated users by access strategy ID.

2. API Description

Items	Description
Operator	`AccessStrategyIDQueryPersons`
Request URL	`http://<server ipaddr>/action/AccessStrategyIDQueryPersons` (Where `<server ipaddr>` is the device IP, for example:192.168.1.10)

3. Explanation of the Paragraph of the Requested Message

   Parameter information(Note: `Optional` is optional)

Key	Type	Values	Description
operator		AccessStrategyIDQueryPersons	Query all associated users by access strategy ID.
info			Specific Information Structures
strategyID	int	1~N	Access strategy ID

4. Example of Request Message

```
{
    "operator":"AccessStrategyIDQueryPersons",
    "info":
    {
        "strategyID":1
    }
}
```

5. Explanation of the Paragraph of the Reply Message

   Parameter information(Note: `Optional` is optional):

Key	Type	Values	Description
operator	string	AccessStrategyIDQueryPersons	Query all associated users by access strategy ID.
info	JSON Object		Concrete content
code	int		Command execution error code 200-successes,see also7.50 Query All Associated Users by Access Strategy ID Error Codes
personNum	int	0~N	Number of people matching the access strategy
personInfo	json array		The json array of the queried person information
PersonUUID	String		PersonUUID is a unique identifier for the list of persons
CustomizeID	Int		CustomizeID is a unique identifier for the list of persons
name	string		Reserve
Result	String	“Ok”/“Fail”	Operating result
Detail	String (optional)		Error message when Result is "Fail"

6. Example of Reply Message

```
{
    "operator": "AccessStrategyIDQueryPersons",
    "code": 200,
    "info": {
        "DeviceID": 1787156,
```

```
        "personNum": 2,
        "personInfo": [
            {
                "PersonUUID": "abcdfdfhFDH",
                "CustomizeID": 1239562,
                "name": "Reserved"
            },
            {
                "PersonUUID": "edfhsfh",
                "CustomizeID": 123459,
                "name": "Reserved"
            }
        ],
        "Result": "OK"
    }
}
```

3.9.8 Personnel Binding Access Strategy

1. Description

   This interface is mainly to realize that when issuing personnel, you can not bind the personnel first, but only issue the personnel list information, and after the list is issued, you can bind the corresponding access strategy. However, it should be noted that the binding relationship between the access strategy information in the list distribution and the personnel list in this interface is **a full override operation**. For example, the first time the personnel is bound to the access strategy group 1 and group 2, the second time the personnel is bound to the access strategy group 1, then the final personnel is bound to the access strategy group 1 only. **Since there can be two fields in the HTTP protocol to identify a unique list, the PersonUUID and CustomizeID for all lists in the all-in-one can only be one or the other, not both.**

2. API Description

Items	Description
Operator	`PersonsBindStrategyID`
Request URL	`http://<server ipaddr>/action/PersonsBindStrategyID` (Where `<server ipaddr>` is the device IP, for example:192.168.1.10)

3. Explanation of the Paragraph of the Requested Message

Key	Type	Values	Description
operator		PersonsBindStrategyID	Personnel binding access strategy
info			Specific Information Structures
personsInfo	json array		People and strategy information keywords
PersonUUID	String		PersonUUID is a unique identifier for the list of persons (48 bytes or less, use either PersonUUID or CustomizeID, not both)
CustomizeID	Int		CustomizeID is a unique identifier for the list of persons （ Use either PersonUUID or CustomizeID, not both ）
strategyID	json array	1-N	Access strategy ID

4. Example of Request Message

```
URL of the request:http://172.168.5.106/action/PersonsBindStrategyID
Content of the request:
{
    "operator": "PersonsBindStrategyID",
    "info": {
        "personsInfo": [
            {
                "CustomizeID":123456,
                "strategyID": [1,2]
            },
            {
                "CustomizeID":1239562,
                "strategyID": [1]
            }
        ]
    }
}
```

5. Explanation of the Paragraph of the Reply Message

   Parameter information(Note:`Optional`is optional):

Key	Type	Values	Description
operator	string	PersonsBindStrategyID	Personnel binding access strategy
info	JSON Object		Concrete content
code	int		Command execution error code 200-successes,see also 7.51 Personnel Binding Access Strategy Error Codes
bindErrNum	string	0~N	Number of failed personnel bind access strategy
bindErrInfo	Array		Personnel binding access strategy failure message keyword
bindSucNum	string	0~N	Number of successful personnel binding access strategy
bindSucInfo	Array		Personnel binding access strategy success message keyword
PersonUUID	String		PersonUUID is a unique identifier for the list of persons
CustomizeID	Int		CustomizeID is a unique identifier for the list of persons
strategyID	Array		Access strategy ID
errcode	String		Error code
Result	String	“Ok”/“Fail”	Operating result
Detail	String (optional)		Error message when Result is "Fail"

6. Example of Reply Message

```
{
    "operator": "PersonsBindStrategyID",
    "code": 200,
    "info": {
        "DeviceID": 1787156,
        "bindErrNum": "0",
        "bindSucNum": "2",
        "bindErrInfo": [],
        "bindSucInfo": [
            {
                "CustomizeID":123456,
```

```
                "strategyID": [
                    1,
                    2
                ]
            },
            {
                "CustomizeID": 1239562,
                "strategyID": [
                    1
                ]
            }
        ],
        "Result": "OK"
    }
}
```

3.9.9 Personnel Unbinding Access Strategy

1. Description

   The Person Unbinding Access Strategy mainly implements the unbinding of a person from a access strategy, if this person is itself on a permanent list, then this person will become **passable**.

2. API Description

Items	Description
Operator	`PersonsUnbindStrategyID`
Request URL	`http://<server ipaddr>/action/PersonsUnbindStrategyID` (Where `<server ipaddr>` is the device IP, for example:192.168.1.10)

3. Explanation of the Paragraph of the Requested Message

   Parameter information(Note: `Optional` is optional)

Key	Type	Values	Description
operator		PersonsUnbindStrategyID	Personnel unbinding access strategy
info			Specific Information Structures
personsInfo	json array		People and strategy information keywords
PersonUUID	String		PersonUUID is a unique identifier for the list of persons (48 bytes or less, use either PersonUUID or CustomizeID, not both)
CustomizeID	Int		CustomizeID is a unique identifier for the list of persons (Use either PersonUUID or CustomizeID, not both)
strategyID	json array		Access strategy ID

4. Example of Request Message

```
URL of the request:http://172.168.5.106/action/PersonsUnbindStrategyID
Content of the request:
{
    "operator": "PersonsUnbindStrategyID",
    "info": {
        "personsInfo": [
            {
                "PersonUUID": "edfhsfh",
                "strategyID": [
                    1
                ]
            },
            {
                "PersonUUID": "1239562",
                "strategyID": [
                    1,
                    2
                ]
            }
        ]
    }
}
```

5. Explanation of the Paragraph of the Reply Message

   Parameter information(Note: `Optional` is optional):

Key	Type	Values	Description
operator	string	PersonsUnbindStrategyID	Personnel unbinding access strategy
info	JSON Object		Concrete content
code	int		Command execution error code 200-successes,see also 7.52 Personnel Unbinding Access Strategy Error Codes
unBindErrNum	string	0~N	Number of failed personnel unbind access strategy
unBindErrInfo	Array		Personnel unbinding access strategy failure message keyword
unBindSucNum	string	0~N	Number of successful personnel unbinding access strategy
unBindSucInfo	Array		Personnel unbinding access strategy success message keyword
PersonUUID	String		PersonUUID is a unique identifier for the list of persons
CustomizeID	Int		CustomizeID is a unique identifier for the list of persons
strategyID	Array		Access strategy ID
errcode	String		Error code
Result	String	“Ok”/“Fail”	Operating result
Detail	String (optional)		Error message when Result is "Fail"

6. Example of Reply Message

```
{
    "operator": "PersonsUnbindStrategyID",
    "code": 200,
    "info": {
        "DeviceID": 1787156,
        "unBindErrNum": "0",
        "unBindSucNum": "2",
        "unBindErrInfo": [],
        "unBindSucInfo": [
```

```
            {
                "PersonUUID": "edfhsfh",
                "strategyID": [
                    1
                ]
            },
            {
                "PersonUUID": "1239562",
                "strategyID": [
                    1,
                    2
                ]
            }
        ],
        "Result": "OK"
    }
}
```

3.9.10 Batch Downstream Access Strategy

1.Description

Enable batch add or modify access strategy time.（**Suggested bulk distribution of 100 items at a time**）

API Description

Items	Description
Operator	`AddAccessStrategy`
Request URL	`http://<server ipaddr>/action/AddAccessStrategys` (Where `<server ipaddr>` is the device IP, for example:192.168.1.10)Parameter information(Note: `Optional` is optional)

Key	Type	Values	Description
operator	String	AddAccessStrategys	Batch downstream access strategy
meessageId	String		Message id, used by the platform to distinguish each message.
facesluiceId	String (optional)		The clientID for connecting to the server, by default, the device ID/SN is used to ensure uniqueness.
bIsInt	int	0,1	Returns the answer data, whether the access strategy ID is a string or an int; 0-string; 1-int (default string, if this value is not passed)
info	Object		Concrete content
strategyID	int	1~N	Access strategy ID
strategyName	String (optional)		Access strategy name (64 bytes including terminator)
accessNumLimit	int		Limitations on the number of passes 0:Unlimited number of passes (default) 1:Restrictions on the number of passes
allowCnt	int (optional)		Number of passe(Reserved, not in force for the time being) Valid when accessNumLimit=1, required
startDate	String	YYYY-MM-DDTh h:mm:ss(2020-01-0 1T00:00:00)	Effective time of access strategy
endDate	String	YYYY-MM-DDTh h:mm:ss(2020-12-2 9T23:59:59)	Access strategy expiration time
monday	Array (optional)		Monday access strategy keywords
tuesday	Array (optional)		Tuesday access strategy keywords
wednesday	Array (optional)		Wednesday access strategy keywords
thursday	Array (optional)		Thursday access strategy keywords

Key	Type	Values	Description
friday	Array (optional)		Friday access strategy keywords
saturday	Array (optional)		Saturday access strategy keywords
sunday	Array (optional)		Sunday access strategy keywords
startTime	String	hh:mm:ss(08:00:00)	Multi-time period start time Default is (00:00:00)
endTime	String	hh:mm:ss(23:59:59)	Multi-time period end time Default is (23:59:59)
holidayInfo	Array		Holiday information keywords
holidayStartDate	String (optional)	YYYY-MM-DDTh h:mm:ss(2020-01-0 1T00:00:00)	Holiday start time
holidayEndDate	String (optional)	YYYY-MM-DDTh h:mm:ss(2020-12-2 9T23:59:59)	Holiday end time
holidayPeriod	Array		Holiday multi-time period keywords (Currently one holiday corresponds to one time period)

4.Example of adding/modifying access strategies from the platform

```
{
    "operator":"AddAccessStrategys",
    "info":[
            {
                "monday":[{"startTime":"15:00:00","endTime":"21:00:00"},{"startTime":"15:00:00","endTime":"20:00:00"}],
                "tuesday":[{"startTime":"05:00:00","endTime":"07:00:00"},{"startTime":"9:00:00","endTime":"10:00:00"}],
                "wednesday":[{"startTime":"12:00:00","endTime":"16:00:00"},{"startTime":"16:01:00","endTime":"23:00:00"}],
                "thursday":[{"startTime":"8:00:00","endTime":"20:00:00"},{"startTime":"08:30:00","endTime":"20:00:00"}],
                "friday":[{"startTime":"06:00:00","endTime":"23:00:00"},{"startTime":"13:00:00","endTime":"14:00:00"}],
                "saturday":[{"startTime":"01:00:00","endTime":"23:00:00"},{"startTime":"02:00:00","endTime":"22:00:00"}],
                "sunday":[{"startTime":"01:00:00","endTime":"23:00:00"}],
```

```
                "holidayInfo":[
                    {"holidayStartDate":"2024-10-
10T00:00:00","holidayEndDate":"2024-10-11T23:00:00","holidayPeriod":
[{"startTime":"01:00:00","endTime":"12:43:00"}]},
                    {"holidayStartDate":"2023-10-
10T00:00:00","holidayEndDate":"2023-10-10T23:59:59","holidayPeriod":
[{"startTime":"09:00:00","endTime":"23:45:00"}]}
                ],
                "bDisablePass":0,
                "nLevel":1,
                "strategyID":75,
                "strategyName":"XXX request leave of absence",
                "startDate":"2023-01-01T00:00:00",
                "endDate":"2099-01-01T23:59:59"
            },
            {
                "monday":[{"startTime":"05:00:00","endTime":"23:59:00"},
{"startTime":"03:00:00","endTime":"23:00:00"}],
                "tuesday":[],
                "wednesday":[{"startTime":"01:00","endTime":"23:00"}],
                "thursday":[],"friday":[],"saturday":[],"sunday":[],
                "strategyName":"2",
                "startDate":"2023-01-01T00:00:500","endDate":"2099-01-01T23:59:59"
            }
        ]
}
```


5.Explanation of the Paragraph of the Reply Message

Parameter information(Note: `Optional` is optional)

Key	Type	Values	Description
operator		AddAccessStrategys-Ack	Batch downstream access strategy reply
messageId	String		Message id, used by the platform to distinguish each message
code	String		Instruction execution error cod 200-successes,see also 3.9.3 Add/Modify Access Strategies
info	Object		Concrete content
facesluiceId	String		The clientID for connecting to the server, by default, the device ID/SN is used to ensure uniqueness.
strategyID	String	1~N	Access strategy ID
AddErrNum	int		Number of failed add access strategies
AddSucNum	int		Number of successful add access strategies
AddErrInfo	Array		Add access strategies failure id data
AddSucInfo	Array		Add access strategies success id data
Result	String	"OK"/"fail"	Operating result
detail	String (optional)		Error message when Result is "Fail"

6.Example of Reply Message

```
{
    "operator": "AddAccessStrategys",
    "code": 200,
    "info": {
        "DeviceID": 1933458,
        "AddErrNum": 0,
        "AddSucNum": 2,
        "AddErrInfo": [],
        "AddSucInfo": [
            75,
            75
        ],
        "Result": "OK"
    }
}
```

4、HTTP Subscription

The types of data subscription include: subscription of authentication result information, subscription of stranger capture, and so on. It is necessary to pay attention to whether the device is enabled to `Continue Transmitting After Disconnection`, and when it is enabled to `Continue Transmitting After Disconnection`, the platform **needs to reply to the device in accordance with the format after** receiving the reported data.

> `Continue Transmitting After Disconnection`:
>
> **The authentication (control) logs** and **stranger capture logs** generated by the device have the difference of enabling and disabling the function of `Continue Transmitting After Disconnection` in the reporting mode. The platform only needs to receive and process the data when the data is reported without enabling the function of `Continue Transmitting After Disconnection`, but **there is a risk of data loss in case of network abnormality**; when the data is reported with the function of `Continue Transmitting After Disconnection` enabled, the platform needs to reply to the device according to the following format within **10 seconds** after receiving the data, and if there is no reply to the device or the reply to the device is incorrect, the Face Recognition All-in-One Machine will push **the same record data** all the time.
>
> ```
> 1 {"code": 200,"desc": "OK"}
> ```

4.1 Reported information does not require authentication

1. Description

   Set the HTTP subscription parameters, the types of subscription messages, the subscription address corresponding to each type of message, the heartbeat interval, and whether to enable the `Continue Transmitting After Disconnection` function. When the device reports data to the subscribed platform without authentication, the value of the `“Auth”` field in the request message is set to `“none”`.

2. API Description

Items	Description
Operator	`Subscribe`
Request URL	`http://<server ipaddr>/action/Subscribe` (Where `<server ipaddr>` is the device IP, for example:192.168.1.10)

3. Explanation of the Paragraph of the Requested Message

   Parameter information(Note: `Optional` is optional):

Key	Type	Values	Description

Key	Type	Values	Description
operator	string	Subscribe	News Subscription
info	json object		Concrete content
DeviceID	int		ID number of the device to be operated See 6.8.1 Getting Device Information
Num	int	1~N	How many types of information t subscribe to
Topics	json array	"Snap", "Verify", "VerifyWithSnap", "VerifyWithReg", "VerifyNoPic", "QRCode", "WGCard", "Alarm", "IDCard"	A collection of single or multiple subscription message types (not exceeding 128 bytes), such as: ["Snap"] : Subscribe to real-time captured logs ["Verify"] : Subscription authentication result upload (with capture + registration image) ["VerifyWithSnap"] : Subscribe t authentication logs reporting (wit snapshots only) ["VerifyWithReg"] : Subscription certification logs upload (with registration image only) ["VerifyNoPic"] : Subscription certification logs reporting (witho images) ["QRCode"] : Subscribe to the Q code to report ["Snap", "Verify"]Subscribe to capture and authentication logs reporting ["IDCard"]Subscription to ID card information reporting ["WGCard"] : Wiegand swipe car information reporting ["Alarm"] : Alarm information reporting
SubscribeAddr	string	http://< Subscribe server ip>:port	Subscription address, including i and port，e.g.,http://192.168.2.1 8080

Key	Type	Values	Description
SubscribeUrl	json object		A collection of single or multiple reported URLs, including capture authentication result, heartbeat, QR code, IC card number authentication result (access control machine), password authentication result (access control machine), and ID card information. e.g., {"Snap":"/Subscribe/Snap", "Verify":"/Subscribe/Verify", "HeartBeat":"/Subscribe/heartbe "Card":"/Subscribe/Card", "QRCode":"/Subscribe/QRCode "PassWord":"/Subscribe/ PassWord ", "IDCard":"/Subscribe/IDCard"}
BeatInterval	int(optional)		Heartbeat interval, in seconds, if not set, default 30 seconds
bTimedPush	int(optional)	0~1	Capture, authentication, QR cod etc. are pushed to the server usi timed push, which is not turned o by default. 0:Don't turn on timed push 1:Turn on timed push
PushInterval	int **Required when bTimedPush=1**		Timed push interval
Auth	string (optional)	“none” or “Basic”	Post Whether to authenticate wh submitting reported information None : Authentication processing not required Basic : post Basic Authentication
User	string **Required when Auth=Basic**		User name, maximum length 64 character length (including terminator)

Key	Type	Values	Description
Pwd	string **Required when Auth=Basic**		Password, maximum length 64 character length (including terminator)

ResumefromBreakpoint	int	0~1	Whether to enable `Continue Transmitting After Disconnection` (with Stranger Capture subscription or Authentication subscription mode enabled)
			0:Not enabled
			1:Enabled.
			Don't enble the `Continue Transmitting After Disconnection` function, only ensure the push data, no need th server return.
			If the feature is enabled, the following json packets are requir to be returned by the server for pushing the stranger capture and authentication information.
			{
			"code": 200,
			"desc": "OK",
			}
			If the All-in-One machine does not receive the correct packet from the server after 10 seconds, it continues to push this message.

Key	Type	Values	Description
BeginTime	string (optional)	YYYY-MM-DDThh:mm:ss	The default start time of the `Continue Transmitting Afte Disconnection`, you can not fill Default is the time to set up the stranger capture subscription or authentication subscription and enble `Continue Transmitting After Disconnection` function the stranger capture and authentication capture informatic before the breakpoint is pushed, this time will be changed to the time of completing the push of th `Continue Transmitting Afte Disconnection`.e.g., 2018-03-12T09:10:00

4. Example of Request Message

```
URL of the request:http://172.168.5.227/action/Subscribe
Content of the request:
{
    "operator": "Subscribe",
    "info": {
        "DeviceID": 1743726,
        "Num": 2,
        "Topics": [
            "Snap",
            "Verify"
        ],
        "SubscribeAddr": "http://172.168.5.99:8080",
        "SubscribeUrl": {
            "Snap": "/Subscribe/Snap",
            "Verify": "/Subscribe/Verify",
            "HeartBeat": "/Subscribe/heartbeat"
        },
        "BeatInterval": 30,
        "ResumefromBreakpoint": 0,
        "Auth": "none"
    }
}
```

5. Explanation of the Paragraph of the Reply Message

   Parameter information(Note: `Optional` is optional):

Key	Type	Values	Description
operator	string	Subscribe	Setting HTTP Subscription Parameters
info	JSON Object		Concrete content
code	int		Command execution error code:200-successes,see also7.23 Setting HTTP Subscription Parameters Error Codes
Result	String	“Ok”/“Fail”	Operating result
Detail	String (optional)		Error message when Result is "Fail"

6. Example of Reply Message

```
{
    "operator": "Subscribe",
    "code": 200,
    "info": {
        "Result": "Ok"
    }
}
```

4.2 Provide Username and Password, the Information Reported Needs to be Authenticated

1. Description

   Set the HTTP subscription parameters, when the device reports data to the subscribed platform for authentication, the `Auth“` field should be `Basic”` `(only` Basic `authentication method is supported at present), and you need to add` User“ `and ``Pwd”` fields in the request message.

2. API Description

Items	Description
Operator	`Subscribe`
Request URL	`http://<server ipaddr>/action/Subscribe` (Where `<server ipaddr>` is the device IP, for example:192.168.1.10)

3. Explanation of the Paragraph of the Requested Message

   Parameter information(Note: `Optional` is optional):

Key	Type	Values	Description

Key	Type	Values	Description
operator	string	Subscribe	News Subscription
info	json object		Concrete content
DeviceID	int		ID number of the device to be operated See 6.8.1 Getting Device Information
Num	int	1~4	How many types of information t subscribe to
Topics	json array	"Snap", "Verify", "VerifyWithSnap", "VerifyWithReg", "VerifyNoPic", "QRCode", "WGCard", "Alarm", "IDCard"	A collection of single or multiple subscription message types (not exceeding 128 bytes), such as ["Snap"] : Subscribe to real-time captured logs ["Verify"] :Subscription authentication result upload (with capture + registration image) ["VerifyWithSnap"] : Subscribe t authentication logs reporting (wit snapshots only) ["VerifyWithReg"] : Subscription certification logs upload (with registration image only) ["VerifyNoPic"] : Subscription certification logs reporting (witho images) ["QRCode"] : Subscribe to the Q code to report ["Snap", "Verify"]: Subscribe to capture and authentication logs reporting ["IDCard"]: Subscription to ID car information reporting ["WGCard"] : Wiegand swipe car information reporting ["Alarm"] : Alarm information reporting
SubscribeAddr	string	http://< Subscribe server ip>:port	Subscription address, including i and port，e.g.,http://192.168.2.1 8080

Key	Type	Values	Description
SubscribeUrl	json object		A collection of single or multiple reported URLs, including capture authentication result, heartbeat, QR code, IC card number authentication result (access control machine), password authentication result (access control machine), and ID card information. e.g., {"Snap":"/Subscribe/Snap", "Verify":"/Subscribe/Verify", "HeartBeat":"/Subscribe/heartbe "Card":"/Subscribe/Card", "QRCode":"/Subscribe/QRCode "PassWord":"/Subscribe/ PassWord ", "IDCard":"/Subscribe/IDCard"}
BeatInterval	int(optional)		Heartbeat interval, in seconds, if not set, default 30 seconds.
bTimedPush	int(optional)	0~1	Capture, authentication, QR cod etc. are pushed to the server usi timed push, which is not turned o by default. 0:Don't turn on timed push 1:Turn on timed push
PushInterval	int **Required when bTimedPush=1**		Timed push interval
Auth	string (optional)	“none” or“Basic”	Post Whether to authenticate wh submitting reported information None : Authentication processing not required Basic : post Basic Authentication
User	string **Required when Auth=Basic**		User name, maximum length 64 character length (including terminator)

Key	Type	Values	Description
Pwd	string **Required when Auth=Basic**		Password, maximum length 64 character length (including terminator)

ResumefromBreakpoint	int	0~1	Whether to enable `Continue Transmitting After Disconnection` (with Stranger Capture subscription or Authentication subscription mode enabled)
			0:Not enabled
			1:Enabled.
			Don't enable the `Continue Transmitting After Disconnection` function, only ensure the push data, no need th server return.
			If the feature is enabled, the following json packets are required to be returned by the server for pushing the strange capture and authentication information.
			{
			"code": 200,
			"desc": "OK",
			}
			If the All-in-One machine does not receive the correct packet from the server after 10 seconds, it continues to push this message.

Key	Type	Values	Description
BeginTime	string (optional)	YYYY-MM-DDThh:mm:ss	The default start time of the `Continue Transmitting Afte Disconnection`, you can not fill Default is the time to set up the stranger capture subscription or authentication subscription and enable the `Continue Transmitting After Disconnection` function. If the stranger capture and authentication capture informatic before the breakpoint is pushed, this time will be changed to the time of completing the push of th `Continue Transmitting Afte Disconnection`. e.g., 2018-03-12T09:10:00

4. Example of Request Message

```
URL of the request:http://192.168.2.10/action/Subscribe
Content of the request:
{
    "operator": "Subscribe",
    "info": {
        "DeviceID": 1743726,
        "Num": 2,
        "Topics": [
            "Snap",
            "Verify"
        ],
        "SubscribeAddr": "http://172.168.5.99:8080",
        "SubscribeUrl": {
            "Snap": "/Subscribe/Snap",
            "Verify": "/Subscribe/Verify",
            "HeartBeat": "/Subscribe/heartbeat"
        },
        "BeatInterval": 30,
        "ResumefromBreakpoint": 0,
        "Auth": "Basic",
        "User": "admin",
        "Pwd": "admin"
    }
}
```

5. Explanation of the Paragraph of the Reply Message

   Parameter information(Note:`Optional`is optional):

Key	Type	Values	Description
operator	string	Subscribe	Setting HTTP Subscription Parameters
info	JSON Object		Concrete content
code	int		Command execution error code:200-successes,see also 7.23 Setting HTTP Subscription Parameters Error Codes
Result	String	“Ok”/“Fail”	Operating result
Detail	String (optional)		Error message when Result is "Fail"

6. Example of Reply Message

```
{
    "operator": "Subscribe",
    "code": 200,
    "info": {
        "Result": "Ok"
    }
}
```

4.3 Strangers Captured and Reported

1. Description

   The capture logs of unregistered strangers in the device list or people whose comparison similarity does not reach the threshold value and are saved in the stranger capture logs of the device, turn on the HTTP stranger capture subscription function of the device, and the device will send the corresponding stranger capture log message to the subscribed URL.

   Note:If the device has enabled the `Continue Transmitting After Disconnection` function, the subscription platform needs to reply to the data reported by the device, otherwise the device will send the same stranger capture log; refer to the description in the4、HTTP Subscription `Continue Transmitting After Disconnection` Function section for the introduction of the specific reply, and note that after the `Continue Transmitting After Disconnection` function is enabled, it is necessary to reply to all the stranger capture logs reported by the device.

2. Explanation of the Paragraph of the Uploaded Message

   Parameter information:

Key	Type	Values	Description
operator	string	SnapPush	Stranger Snap Push
info	json object		Concrete content
DeviceID	int		ID number of the device to be operated See 6.8.1 Getting Device Information
CreateTime	string	YYYY-MM-DDThh:mm:ss	Stranger capture recording time. e.g., 2018-03-25T21:50:05
PictureType	unsigned int	0~4	Capture face image type 0: Failed 1: Successful blacklisting 2: Successful whitelisting 3:Failed ID+Face Matching 4:Successful ID+Face Matching
isNoMask	int (optional)	0~2	Whether to wear a mask **(Note: Mask model support)** 0:Passage with mask 1:Passage without mask 2:Passage without mask (open door condition checkbox allows passage without mask)
Temperature	double (optional)	37.20	Real-time detection of face temperature, return to detect temperature only when the door opening mode with temperature detection mode **(Temperature measurement model support)**
TemperatureAlarm	int (optional)	0~1	Real-time detection of whether the face temperature exceeds the threshold value **(Temperature measurement model support)** 0:Not exceeded; 1:Exceeded
TemperatureMode	int (optional)	0~1	Temperature display mode **(Temperature measurement model support)** 0: Celsius temperature 1: Fahrenheit temperature

Key	Type	Values	Description
Sendintime	int	0~1	Whether information is pushed out in a timely manner 0:Failure to push in a timely manner (push time greater than 10 seconds from the time of capture) 1:Timely push
Direction	int	0~2	Entrance/Exit Direction 0:Unidirectional 1:Entrance 2:Exit
SanpPic	string		Base64 encoded data for face images captured by strangers

3. Example of Uploaded Message

```
{
    "operator": "SnapPush",
    "info": {
        "DeviceID": 1743726,
        "CreateTime": "2022-12-20T09:29:53",
        "PictureType": 0,
        "Sendintime": 1 ,
        "Direction": 1
    },
    "SanpPic": "........."
}
```

4.4 Reporting of Certification Results

1. Description

   When the device turns on authentication logs subscription, all personnel identification logs (control logs) generated are reported to the subscribed URL, control records include (but are not limited to)

   1)Regular personnel identification logs

   2)Control logs generated by remote door opening commands

   3)Unauthorized access logs generated under a special version (access strategy)(“VerifyStatus”:24)

   4)Logs of blacklisted denials of passage(“VerifyStatus”:2)

Note:If the device has enabled the `Continue Transmitting After Disconnection` function, the subscription platform needs to reply to the data reported by the device, otherwise the device will send the same stranger capture log; refer to the description in the 4、HTTP Subscription `Continue Transmitting After Disconnection` Function section for the introduction of the specific reply, and note that after the `Continue Transmitting After Disconnection` function is enabled, it is necessary to reply to all the stranger capture logs reported by the device.

2. Explanation of the Paragraph of the Uploaded Message

   Parameter information:

Key	Type	Values	Description
operator	string	VerifyPush	Reporting of certification results
info	json object		Concrete content
DeviceID	int		ID number of the device to be operated See 6.8.1 Getting Device Information
PersonID	int		The device stores the ID number of the person, in the case of CustomizeID and PersonUUID is invalid, you can use this ID to modify the information of the list of people.
CreateTime	string	YYYY-MM-DDThh:mm:ss	Creation time of the control log. e.g., 2018-03-25T21:50:05
Similarity1	float	0.0~100.0	Black and white list comparison similarity
Similarity2	float	0.0~100.0	ID image comparison similarity
VerifyStatus	int	0~3 or 22、24	Authentication results 0: None 1: Allowed 2: Denied 3: Not yet registered 22:To be verified (control record for door opening method 3: face verification + remote door opening method) 24:No permission (control record for special version of non-pass time period)

Key	Type	Values	Description
VerfyType	int	1:Universal model 1~3 or 21~22 or 24~25 or 27 ; 2:Special model: **Support for mask detection** or (|) operation 0x100 indicates a pass with mask detection; **Supports body temperature detection** or (|) operation 0x200 indicates a pass with temperature detection ; **Support mask + body temperature detection** Or (|) operation 0x300 means with mask + temperature detection pass; e.g.Masks + whitelist verification (0x101) Body Temperature + Whitelist Verification (0x201) **Fingerprint support:** Face+Fingerprint 60 Swipe Card + Fingerprint 61 Face+Swipe+Fingerprint 62 Fingerprint 65	Authentication Type 1: Whitelist Verification 2: ID Verification 3: Whitelist + ID Verification 21:RF Card Verification(**Built-in card swiping model**) 22:RF Card Verification + Whitelist Verification(**Built-in card swiping model**) 24:Wergand Card Verification 25. Wigan Card + Whitelist Verification 27:HTTP Remote Door Opening
PersonType	int	0~1	List Type 0: White list 1: Black list
Name	string		Name Maximum length 32 character length (including terminator)

Key	Type	Values	Description
Gender	int	0~2	Genders 0: Male 1: Female 2: All
Nation	int	1~57	Native
CardType	int		ID type 0:ID
IdCard	string		ID number, maximum length 32 character length (including terminator)
Birthday	string	YYYY-MM-DD	Birthday，e.g., 2018-1-1
Telnum	string		Phone number, maximum length 32 characters long (including terminator)
Native	string		Native, maximum length 32 character length (including terminator)
Address	string		Address, maximum length 72 character length (including terminator)
Notes	string		Notes, maximum length 64 character length (including terminator)
MjCardFrom	int	0~3	Wiegand Card Number Generation Method 0: public number 1: automatic generation 2: manual input 3: do not use access card numbers

Key	Type	Values	Description
WGFacilityCode	int (optional)		Facility code When **Wiegand=4 or 5 or 6 or 7** in 6.1 Door Opening Conditions and Output Control Parameters and the **MjCardNo** field together form the Wiegand card number When **Wiegand=0 or 1**, this parameter has no practical meaning. See 3.1.6 Wiegand Access Card Number and Built-in Swipe ID Card Number
MjCardNo	unsigned int		Wiegand access card number （userid ） When **Wiegand=4 or 5 or 6 or 7** in 6.1 Door Opening Conditions and Output Control Parameters and the **WGFacilityCode** field together form the Wiegand Card Number When **Wiegand=0 or 1**, **alone** constitutes a Wiegand card number. See 3.1.6 Wiegand Access Card Number and Built-in Swipe ID Card Number
RFIDCard	string (optional)		ID Card Number Maximum length is 18 characters,for built-in card machine type(including terminator). See [3.1.6 Wiegand Access Card Number and Built-in Swipe ID Card Number](#316-Wiegand Access Card Number and Built-in Swipe ID Card Number

Key	Type	Values	Description
Tempvalid	int	0~1	Is it a temporary list 0: Permanent list 1: Temporary list 1 (starting and ending time periods) 2: Temporary list 2 (daily time slots, supported by new database version) 3: Temporary list 3 (number of times valid, supported by new database version) 4: Temporary List 4 (a combination of Temporary List 2 and Temporary List 3, supported by the new database version and after neutral version 9.4)
CustomizeID	unsigned int		User-defined ID You can manage the corresponding personnel information such as modification and deletion by this ID number. Note: You can not use the same ID to add people repeatedly, otherwise it will lead to errors in modifying and deleting operations.
PersonUUID	string		User-defined UUID (not more than 48 bytes) This UUID number can be used to manage personnel information such as modification and deletion. Note: You can not use the same ID to add people repeatedly, otherwise it will lead to errors in operations such as modification and deletion
ValidBegin	string	YYYY-MM-DDThh:mm:ss	In the case of Temporary List 1 or Temporary List 4, the start time of the entry into force of the Temporary List . e.g., 2018-03-12T09:09:20

Key	Type	Values	Description
ValidEnd	string	YYYY-MM-DDThh:mm:ss	In the case of Temporary List 1 or Temporary List 4, the end of the effective date of Temporary List . e.g., 2018-03-12T09:10:00
isNoMask	int	0~2	Whether to wear a mask **(Note: Mask model support)** 0:Passage with mask 1:Passage without mask 2:Passage without mask (open door condition checkbox allows passage without mask)
Temperature	double	37.20	Real-time detection of face temperature, return to detect temperature only when the door opening mode with temperature detection mode **(Temperature measurement model support)**
TemperatureAlarm	int	0~1	Real-time detection of whether the face temperature exceeds the threshold value **(Temperature measurement model support)** 0:Not exceeded; 1:Exceeded
Sendintime	int	0~1	Whether information is pushed in a timely manner 0:Not pushed in a timely manner (push time is greater than the authentication time by 10 seconds) 1:Pushed in a timely manner
Direction	int (optional)	0~2	Entrance/Exit Direction 0:Unidirectional 1:Entrance 2:Exit
SanpPic	string		Base64 encoded data for face images captured
RegisteredPic	string		Base64 encoded data of face images registered in the list library

Key	Type	Values	Description
PushType	int	0~2	Push type 0:Reserved 1:Reserved 2:Call 4.9 Manual Push Control Logging data return
OpendoorWay	int	0~3	Open the door 0:Face open the door 1:Remote open the door 2:Remote open the door or face 3:Local verification remote open the door (comparison verification by the local completion of the report, open the door by the remote decision)
szQrCodeData	string		QR code content
dwFileIndex	string	0-N	File index of of image captured in control log
dwFilePos	string	0-N	Location of image captured in control log
SanpPic	string	base64	Snapshots image
RegisteredPic	string	base64	Registered image

3. Example of Uploaded Message

```
{
    "operator": "VerifyPush",
    "info": {
        "PersonID": 1,
        "CreateTime": "2022-12-20T09:51:41",
        "Similarity1": 96.065056,
        "Similarity2": 0.000000,
        "VerifyStatus": 1,
        "VerfyType": 1,
        "PersonType": 0,
        "Name": "boy",
        "Gender": 0,
        "Nation": 1,
        "CardType": 0,
        "IdCard": " ",
        "Birthday": "2000-01-01",
        "Telnum": " ",
        "Native": " ",
```

```
        "Address": " ",
        "Notes": " ",
        "MjCardFrom": 0,
        "DeviceID": 1743726,
        "PushType": 1,
        "OpendoorWay": 0,
        "szQrCodeData": "",
        "MjCardNo": "1",
        "RFIDCard": "0",
        "Tempvalid": 0,
        "CustomizeID": 0,
        "PersonUUID": " ",
        "ValidBegin": "0000-00-00T00:00:00",
        "ValidEnd": "0000-00-00T00:00:00",
        "Sendintime": 1,
        "Direction": 1
    },
    "dwFileIndex": "0",
    "dwFilePos": "1114112",
    "SanpPic":
".......TC6OTC7ujD8OvF7+rE8OrH7+nF7ej///",
    "RegisteredPic":
".........L2tKQ2M+M2M6OOM19GM"
}
```


4.5 QR Code Reporting

1. Description

   For face machines with external QR code readers or face machines with built-in QR code readers **(except for health code models)**, after the device turns on the QR code subscription function, the device reports the QR code information to the subscribed URL.

2. Explanation of the Paragraph of the Uploaded Message

   Parameter information:

Key	Type	Values	Description
operator	string	QRCodePush	QR code reporting
info	json object		Concrete content
DeviceID	int		ID number of the device to be operated See 6.8.1 Getting Device Information
CreateTime	string	YYYY-MM-DDThh:mm:ss	QR code scanning time。e.g., 2018-03-25T21:50:05
QRcodeInfo	string		QR code content

3. Example of Uploaded Message

```
{
    "operator": "QRCodePush",
    "info": {
        "DeviceID":1300001,
        "CreateTime":"2024-11-01T13:33:00",
        "QRcodeInfo":"a1b2c.....lAS23dfW39"
    }
}
```

4.6 Reporting of Identity Card Information

1. Description

   Face recognition machine in the ID card verification of the door opening mode, at the same time open the ID card information subscription function, each time to brush the ID card information will be reported to the subscription of the URL.

2. Explanation of the Paragraph of the Uploaded Message

Key	Type	Values	Description
operator	string	IDCardInfoPush	ID Information Push
info	json object		Concrete content
DeviceID	int		ID number of the device to be operated See 6.8.1 Getting Device Information
IDCard_Idno	string		I.D. number
IDCard_Name	string		Name
IDCard_Gender	int		Genders
IDCard_Nation	int		nation
IDCard_Birthday	string		Birthday
IDCard_Address	string		Address
IDCard_Idissue	string		Issuing authority
IDCard_Idperiod	string		Validity period: Month/Year - Month/Year
IDCard_photo	string		Photo ID, base64 encoding

3. Example of Uploaded Message

```
{
```

```
    "operator": "IDCardInfoPush",
    "info": {
        "DeviceID": 1300001,
        "IDCard_Idno": "1234567890",
        "IDCard_Name": "AA",
        "IDCard_Gender": 1,
        "IDCard_Nation": 1,
        "IDCard_Birthday":"2016-05-06",
        "IDCard_Address": "xxxxx",
        "IDCard_Idissue": "xxxxxx",
        "IDCard_Idperiod": "20190203-20190304"
    },
    "IDCard_photo": "....."
}
```

4.7 IC Card/RFID Card Reporting

1. Description

   Facial recognition all-in-one machine in the door opening mode with card authentication, at the same time open the RF card subscription function, each time the card number information will be swiped by the device will be reported to the subscription of the URL。

2. Explanation of the Paragraph of the Uploaded Message

   Parameter information:

   Built-in card model

Key	Type	Values	Description
operator	string	ICCardInfoPush	IC Card Number Information Push
info	json object		Concrete content
DeviceID	int		ID number of the device to be operated See 6.8.1 Getting Device Information
ICCardNum	string		IC card number (not more than 24 bytes)

   Non-built-in card model

Key	Type	Values	Description
operator	string	WGCardInfoPush	Wiegand Card Number Information Push
info	json object		Concrete content
DeviceID	int		ID number of the device to be operated See 6.8.1 Getting Device Information
WGCard_Idno	string		Wiegand card number (not exceeding 24 bytes)

3. Example of Uploaded Message

Built-in card model

```
{
    "operator": " ICCardInfoPush",
    "info": {
        "DeviceID":1300001,
        "ICCardNum ": "……"
    }
}
```

Non-built-in card model

```
{
    "operator": " WGCardInfoPush",
    "info": {
        "DeviceID":1300001,
        "WGCard_Idno ": "……"
    }
}
```

4.8 Alarm Reporting

1. Description

   When alarm subscription is enabled, the device reports alarm information to the corresponding URL.

2. Explanation of the Paragraph of the Uploaded Message

   Parameter information:

Key	Type	Values	Description
operator	string	AlarmInfoPush	Alarm information reporting
info	json object		Concrete content
DeviceID	int		ID number of the device to be operated See 6.8.1 Getting Device Information
code	int		Alarm Event Codes: 1: Door Magnetic Door Opening Timeout Alarm 2: Forced Door Opening Alarm 3: Takedown Alarm 4: Live Attack Alarm 5: Fire Alarm 6: Door Magnetic Status 7: Open Door Button 8: Open Door Chime
description	string(optional)		Alarm Event Description
event_time	long long		Event timestamp
status	int		Event status: Fire and tamper events:1-alarm status, 2-alarm canceled status, 3-alarm manually canceled status Open door events:1-open door timeout, 2-forced open door Door magnetic events:1-door magnetic alarm status, 2-door magnetic alarm canceled status
reserve	string(optional)		reserve

3. Example of Uploaded Message

```
Example of door magnetic status reporting.
{
    "operator": "AlarmInfoPush",
    "info": {
        "code":6,
        "description":"Magnetic door state",
        "event_time":1620982161074,
        "status":1,
        "DeviceID":1478428,
        "reserve":"reserve"
    }
}
```

```
{
    "operator": "AlarmInfoPush",
    "info": {
        "code":6,
        "description":"Magnetic door state",
        "event_time":1620982164375,
        "status":2,
        "DeviceID":1478428,
        "reserve":"reserve"
    }
}

Example of a dismantling alarm report.
{
    "operator": "AlarmInfoPush",
    "info": {
        "code":3,
        "description":"Tamper alarm",
        "event_time":1620985255475,
        "status":1,
        "DeviceID":1478428,
        "reserve":"reserve"
    }
}

{
    "operator": "AlarmInfoPush",
    "info": {
        "code":3,
        "description":"Tamper alarm",
        "event_time":1620985259137,
        "status":2,
        "DeviceID":1478428,
        "reserve":"reserve"
    }
}
```

4.9 Heartbeat Reporting

1. Description

   The device gets the address, port and URL of the subscribed heartbeat from the subscription configuration information, the device sends the heartbeat information to the subscription address throughout the subscription validity period, the subscription server **needs to reply to the heartbeat** after receiving the heartbeat from the device to keep the two sides validly connected, once the heartbeat handshake fails, the device side **will try to reconnect to the subscription address after shutting**

down the connection, and the interval of the heartbeat is set at 30 seconds. The heartbeat **report needs to subscribe to any HTTP message (authentication subscription or stranger capture subscription, etc.), can not subscribe to a separate heartbeat**.

2. Explanation of the Paragraph of the Uploaded Message

   Parameter information:

Key	Type	Values	Description
operator	string	HeartBeat	Heartbeat Information Reporting
info	json object		Concrete content
DeviceID	int		ID number of the device to be operated See 6.8.1 Getting Device Information
Ip	string		IP address of the device
MacAddr	string		Mac address of the device
WifiIp	string (optional)		IP address of the wifi (wifi needs to be configured)
Time	string	format:YYYY-MM-DDThh:mm:ss	The current time of the device. e.g.,2018-03-25T21:50:05

3. Example of Uploaded Message

```
{
    "operator": "HeartBeat",
    "info": {
        "DeviceID": 1743726,
        "Ip": "172.168.5.227",
        "MacAddr": "5c:f2:86:89:3a:b1",
        "Time": "2022-12-20T15:32:34"
    }
}
```

4.10 Query Subscribed Information

1. Description

   Queries the device's current HTTP subscription-related configuration information, including the type of subscription message, the IP address of the subscribing platform, the URL of each topic subscription, whether or not the `Continue Transmitting After Disconnection` function is enabled, whether or not authentication is enabled, and so on.

2. API Description

Items	Description
Operator	`GetSubscribe`
Request URL	`http://<server ipaddr>/action/GetSubscribe` (Where `<server ipaddr>` is the device IP, for example:192.168.1.10)

3. Example of Request Message

```
URL of the request:http://192.168.2.10/action/GetSubscribe
Content of the request:（空）
```

4. Explanation of the Paragraph of the Reply Message

   Parameter information(Note: `Optional` is optional):

Key	Type	Values	Description

Key	Type	Values	Description
operator	string	GetSubscribe	Getting configuration informa for a subscription
info	json object		Concrete content
DeviceID	int		ID number of the device to b operated See 6.8.1 Getting Device Information
ProtocolType	int	0~1	HTTP protocol type: 0: LAN 1: WAN
Num	int	1~4	How many types of informat subscribe to

Key	Type	Values	Description
Topics	json array	"Snap", "Verify", "VerifyWithSnap", "VerifyWithReg", "VerifyNoPic", "QRCode", "WGCard", "Alarm", "Card", "PassWord" , "IDCard"	A collection of single or mult subscription message types exceeding 128 bytes), e.g.: ["Snap"] : subscribe to real-ti snap upload ["Verify"] : subscribe to authentication result upload snap + registration image) ["VerifyWithSnap"] : subscrib authentication result upload snap image only) ["VerifyWithReg"] : subscribe authentication result report (registration image only) ["VerifyNoPic"] : subscribe authentication result report (image) ["QRCode"] : subscribe QR c report ["Snap", "Verify"] subscribe s and authentication result rep ["IDCard"] subscribe ID card information report ["WGCard"] : Wigan swipe c information report [" Alarm"] : Alarm informatior reporting ["Card"] subscription access machine IC card authenticat results reporting **(12-inch ac control machine models ar valid)** ["PassWord"] subscription ac control machine password authentication results reporti **(12-inch access control ma models are valid)** ["PassWc subscription access control machine password authentic results reporting **(12-inch ac control machine models ar valid)**

Key	Type	Values	Description
SubscribeAddr	string	http://< Subscribe server ip>:port	Subscription address, includ and port , e.g.,http://192.168 8080
SubscribeUrl	json object		A collection of single or mult reported URLs, including cap authentication result, heartb QR code, IC card number authentication result , passw authentication result , and ID information. e.g., {"Snap":"/Subscribe/Snap", "Verify":"/Subscribe/Verify", "HeartBeat":"/Subscribe/hea "Card":"/Subscribe/Card", "QRCode":"/Subscribe/QRC "PassWord":"/Subscribe/ PassWord ", "IDCard":"/Subscribe/IDCard
BeatInterval	int(optional)		Heartbeat interval, in second not set, default 30 seconds
bTimedPush	int(optional)	0~1	Capture, authentication, QR etc. are pushed to the serve timed push, which is not turn by default. 0:Don't turn on timed push 1:Turn on timed push
PushInterval	int(optional)		Timed push interval
Auth	string (optional)	“none” or“Basic”	Post Whether to authenticate submitting reported informat None : Authentication proces not required Basic : post Basic Authentica
User	string(optional) **Required when Auth=Basic**		User name, maximum length character length (including terminator)

Key	Type	Values	Description
Pwd	string(optional) **Required when Auth=Basic**		Password, maximum length character length (including terminator)

ResumefromBreakpoint	int	0~1	Whether to enable `Continu Transmitting After Disconnection` (with Stran Capture subscription or Authentication subscription r enabled)
			0:Not enabled
			1:Enabled.
			Don't enable `Continue Transmitting After Disconnection` function, or ensure the push data, no ne server return.
			If the feature is enabled, the following json packets are re to be returned by the server pushing the stranger capture authentication information.
			{
			"code": 200,
			"desc": "OK",
			}
			If the All-in-One machine d not receive the correct pac from the server after 10 seconds, it continues to pu this message.

Key	Type	Values	Description
BeginTime	string	YYYY-MM-DDThh:mm:ss	The default start time of the `Continue Transmitting A Disconnection`, you can nc Default is the time to set up t stranger capture subscriptio authentication subscription a enbale the `Continue Transmitting After Disconnection`. If the stran capture and authentication c information before the break pushed, this time will be cha to the time of completing the of the `Continue Transmitt After Disconnection`.e.g. 03-12T09:10:00

5. Example of Reply Message

```
{
    "operator": "GetSubscribe",
    "info": {
        "ProtocolType": 0,
        "Num": 2,
        "Topics": [
            "Snap",
            "Verify"
        ],
        "SubscribeAddr": "http://172.168.5.99:8080",
        "SubscribeUrl": {
            "Snap": "/Subscribe/Snap",
            "Verify": "/Subscribe/Verify",
            "HeartBeat": "/Subscribe/heartbeat"
        },
        "BeatInterval": 30,
        "ResumefromBreakpoint": 0,
        "BeginTime": "2022-12-19T17:40:50",
        "Auth": "none"
    }
}
```

4.11 Unsubscribe

1. Description

   Unsubscribe from topics, types include:Subscription for real-time capture information and subscription for authentication result information.

2. API Description

Items	Description
Operator	`Unsubscribe`
Request URL	`http://<server ipaddr>/action/Unsubscribe` (Where `<server ipaddr>` is the device IP, for example:192.168.1.10)

3. Explanation of the Paragraph of the Requested Message

   Parameter information(Note: `Optional` is optional):

Key	Type	Values	Description
operator	string	Unsubscribe	Unsubscribe
info	json object		Concrete content
DeviceID	int		ID number of the device to be operated See 6.8.1 Getting Device Information
Num	int	1~2	How many types of information to unsubscribe to
Topics	json array	"Snap", "Verify", "VerifyWithSnap", "VerifyWithReg", "VerifyNoPic", "QRCode", "WGCard", "Alarm", "IDCard"	A collection of single or multiple subscription message types (not exceeding 128 bytes), e.g.: ["Snap"] : subscribe to real-time snap upload ["Verify"] : subscribe to authentication result upload (with snap + registration image) ["VerifyWithSnap"] : subscribe to authentication result upload (with snap image only) ["VerifyWithReg"] : subscribe to authentication result report (with registration image only) ["VerifyNoPic"] : subscribe authentication result report (without image) ["QRCode"] : subscribe QR code report ["Snap", "Verify"] subscribe snap and authentication result report ["IDCard"] subscribe ID card information report ["WGCard"] : Wigan swipe card information report [" Alarm"] : Alarm information reporting

4. Example of Request Message

```
URL of the request:http://192.168.2.10/action/Unsubscribe
Content of the request:
{
    "operator": "Unsubscribe",
    "info": {
        "DeviceID": 1300001,
        "Num": 2,
        "Topics":["Snap", "Verify"]
    }
}
```

5. Explanation of the Paragraph of the Reply Message

   Parameter information(Note: `Optional` is optional):

Key	Type	Values	Description
operator	string	Unsubscribe	Unsubscribe
info	JSON Object		Concrete content
code	int		Command execution error code:200-successes,see also7.24 Unsubscribe Error Codes
Result	String	“Ok”/“Fail”	Operating result
Detail	String (optional)		Error message when Result is "Fail"

6. Example of Reply Message

```
{
    "operator": "Unsubscribe",
    "code": 200,
    "info": {
        "Result": "Ok"
    }
}
```

4.12 Manual Push Control Logging

1. Description

This interface is mainly to solve the problem of losing control records reported due to HTTP server-side reasons 4.4 Reporting of Certification Results, and manually through the interface settings, the All-in-One machine again actively uploads the The All-in-One again actively uploads the certification results of the time period to the server through the interface settings.

This interface requires that the subscription address and port settings of 4、HTTP Subscription are correct and **that the subscription for authentication result reporting is enabled**. This interface is mandatory to push the searched control logs, whether or not to use the `Continue Transmitting After Disconnection` function has no effect on this interface, this interface upload control logs **do not need** the server to reply, at the same time need to pay attention to the data reported by this interface may be due to the same record has been reported to the server before the server to save the same data more than once.

2. API Description

Items	Description
Operator	`ManualPushRecords`
Request URL	`http://<server ipaddr>/action/ManualPushRecords` (Where `<server ipaddr>` is the device IP, for example:192.168.1.10)

3. Explanation of the Paragraph of the Requested Message

   Parameter information(Note: `Optional` is optional):

Key	Type	Values	Description
operator		ManualPushRecords	Manual push control logging
info			Concrete content
DeviceID	int		ID number of the device to be operated See 6.8.1 Getting Device Information
TimeS	string	YYYY-MM-DDThh:mm:ss	Control log start time that needs to be pushed.e.g., 2020-07-29T21:50:05
TimeE	string	YYYY-MM-DDThh:mm:ss	Control log end time that needs to be pushed.e.g., 2020-07-29T21:50:05

4. Example of Request Message

```
URL of the request:http://172.168.5.227/action/ManualPushRecords
Content of the request:
{
    "operator": "ManualPushRecords",
    "info": {
        "DeviceID": 1743726,
        "TimeS": "2020-07-28T9:00:00",
        "TimeE": "2020-08-07T9:00:00"
    }
}
```

5. Explanation of the Paragraph of the Reply Message

   Parameter information(Note: `Optional` is optional):

Key	Type	Values	Description
operator	string	ManualPushRecords	Manual push control logging
info	JSON Object		Concrete content
code	int		Command execution error code 200-successes,see also 7.25 Manual Push Control Logging Error Codes
Result	String	“Ok”/“Fail”	Operating result
Detail	String (optional)		Error message when Result is "Fail"
RecordNum	string (optional)		Total number of matching control logs
RecordTip	string (optional)		Alerts

6. Example of Reply Message

```
{
    "operator": "ManualPushRecords",
    "code": 200,
    "info": {
        "Result": "Ok",
        "RecordNum": "4",
        "RecordTip": "Confirm connection to HTTP server is Ok ? Find 4
records, try to push now"
    }
}
```

4.13 Subscribe to Reporting Responses

1. Description

   The reply only takes effect when the door opening method is set to "**Platform verification, remote door opening**", and applies to the subscription reports of chapters 4.3~4.8, i.e. stranger capture report, authentication report, QR code report, ID card information report, IC card/RFID card report, alarm report, and the reply can perform the corresponding door opening or display operation. Reporting, alarm reporting, the reply can perform the corresponding door opening or display operation.

2. Explanation of the Paragraph of the Reply Message

   Parameter information(Note: `Optional` is optional):

Key	Type	Values	Description
code	int		Command execution error code 200-successes
openDoor	int (optional)		Whether to open the door or not, 0:not open the door, 1:open the door
showInfo	String (optional)		Set the current content of the device to be displayed, such as: please pass, prohibited, etc., the content of the utf-8 format. This parameter can be used without displaying the content, and can be used with status.
msgAttribute	JSON Object (optional)		Set the style of the text to be displayed, including "color", 'pleace', "font" fields, if not set, the default text style of the device will be used.
color	String (optional)		Text color, using hexadecimal color code, e.g., :ffffff; default is ffffff
pleace	int (optional)		Position of text display 0: centered 1: to the left 2: to the right Default 0
font	int (optional)		Font size for text Default 24pt

3. Example of Reply Message

```
{
    "code": 200,
    "openDoor": 1,
    "showInfo":"welcome",
    "msgAttribute":{
        "color":"ff00ff",
        "pleace":1,
        "font": 10
    }
}
```

5、Device Management

5.1 Software Version Upgrade

1. Description

   In 6.8.1 Getting Device Information, you can get the software version of the current Face Recognition All-in-One Machine, check the version that needs to be upgraded and the current software version of the Face Recognition All-in-One Machine to meet the upgrade premise of the corresponding version, and after the upgrade version is successfully upgraded, the Face Recognition All-in-One Machine will be restarted automatically.

2. API Description

Items	Description
Operator	`Upgrade`
Request URL	`http://<server ipaddr>/action/Upgrade` (Where `<server ipaddr>` is the device IP, for example:192.168.1.10)

3. Explanation of the Paragraph of the Requested Message

   Parameter information(Note: `Optional` is optional):

Key	Type	Values	Description
operator	string	Upgrade	Software version upgrade
info	json object		Concrete content
DeviceID	int		ID number of the device to be operated See 6.8.1 Getting Device Information
Name	string		Software version name
upgradeType	int (optional)	1-4	Upgrade type, default 1; 1: Indicates upgrading software firmware package, *.swx file; 2: Upgrading voice file, *.wav file; 3: Upgrading png image file, *.png file; 4: Upgrading jpg image file, *.jpg file
Path	string	e.g.:https://hq.oss-cn-shenzhen.aliyuncs.com/face/xxx.swx	The download path of the software version upgrade file (http or https), without upgradeType field or upgradeType=1, it means the download path of the upgrade file, **the suffix must be ".swx"**.

4. Example of Request Message

```
URL of the request:http://192.168.2.10/action/Upgrade
Content of the request:
{
    "operator": "Upgrade",
    "info": {
        "DeviceID": 1743726,
        "Name": "New Version",
        "upgradeType": 1,
        "Path": "http://172.168.5.99:8080/FaceGate_vall.52.11.0-L-M-[Ix-4G-QR-W-2.5W]-20221219.swx"
    }
}
```

5. Explanation of the Paragraph of the Reply Message

   Parameter information(Note: `Optional` is optional):

Key	Type	Values	Description
operator	string	Upgrade	Software version upgrade
info	JSON Object		Concrete content
code	int		Command execution error code 200-successes,see also7.22 Software Version Upgrade Error Codes
Result	String	“Ok”/ “Fail”	Operating result
Detail	String (optional)		Error message when Result is "Fail"

6. Example of Reply Message

```
{
    "operator": "Upgrade",
    "code": 200,
    "info": {
        "Result": "Ok"
    }
}
```

5.2 Device Reboot

1. Description

   Device reboot.

2. API Description

Items	Description
Operator	`RebootDevice`
Request URL	`http://<server ipaddr>/action/RebootDevice` (Where `<server ipaddr>` is the device IP, for example:192.168.1.10)

3. Explanation of the Paragraph of the Requested Message

   Parameter information(Note: `Optional` is optional):

Key	Type	Values	Description
operator	string	RebootDevice	Device reboot
info	json object		Concrete content
DeviceID	int		ID number of the device to be operated See 6.8.1 Getting Device Information
IsRebootDevice	int	0~1	0:Do not reboot the device 1:Reboot the device

4. Example of Request Message

```
URL of the request:http://172.168.2.10/action/RebootDevice
Content of the request:
{
    "operator": "RebootDevice",
    "info": {
        "DeviceID": 1743726,
        "IsRebootDevice": 1
    }
}
```

5. Explanation of the Paragraph of the Reply Message

   Parameter information(Note: `Optional` is optional):

Key	Type	Values	Description
operator	string	RebootDevice	Device reboot
info	JSON Object		Concrete content
code	int		Command execution error code 200-successes,see also7.26 Device Reboot Error Codes
Result	String	“Ok”/“Fail”	Operating result
Detail	String (optional)		Error message when Result is "Fail"

6. Example of Reply Message

```
{
    "operator": "RebootDevice",
    "code":200,
    "info": {
        "Result": "Ok"
    }
}
```

5.3 Remote Open Door

1. Description

   Remote control of door opening.

2. API Description

Items	Description
Operator	`OpenDoor`
Request URL	`http://<server ipaddr>/action/OpenDoor` (Where `<server ipaddr>` is the device IP, for example:192.168.1.10)

3. Explanation of the Paragraph of the Requested Message

   Parameter information(Note: `Optional` is optional):

Key	Type	Values	Description
operator	string	OpenDoor	Directly open the door, you can check whether the specific control mode is switching or Wiegand.6.1 Door Opening Conditions and Output Control Parameters, Wiegand is using a public card number.
info	json object		Concrete content
DeviceID	int		ID number of the device to be operated See 6.8.1 Getting Device Information
msg	string (optional)		Set the current content of the device to be displayed, such as: please pass, prohibited, etc., the content of the utf-8 format. This parameter can be used without displaying the content, and can be used with status.
status	int (Required with “msg” field)		**This value is greater than 0 to open the door**, don't need this parameter if you don't want to show the content, use it with msg.
msgAttribute	json object (optional)		Set the style of the text to be displayed, including “color”, ‘pleace’, “font” fields, if not set, the default text style of the device will be used.
color	string (optional)		Text color, using hexadecimal color code, e.g., :ffffff; default is ffffff
pleace	string (optional)		Position of text display 0: centered 1: to the left 2: to the right Default 0
font	string (optional)		Font size for text Default 24pt

4. Example of Request Message

```
URL of the request:http://172.168.5.227/action/OpenDoor
Content of the request:
{
    "operator": "OpenDoor",
    "info": {
        "DeviceID": "1770137",
        "status": 1,
```

```
        "msg": "Welcome!",
        "msgAttribute":{
            "color":"ff00ff",
            "pleace":"1",
            "font": "10"
        }
    }
}
```

5. Explanation of the Paragraph of the Reply Message

   Parameter information(Note:`Optional`is optional):

Key	Type	Values	Description
operator	string	OpenDoor	Remote control of door opening.
info	JSON Object		Concrete content
code	int		Command execution error code 200-successes,see also 7.34 Remote Open Door Error Codes
Result	String	“Ok”/“Fail”	Operating result
Detail	String (optional)		Error message when Result is "Fail"

6. Example of Reply Message

```
{
    "operator": "OpenDoor",
    "code": 200,
    "info": {
        "Result": "Ok"
    }
}
```

5.4 Restore Factory Settings

This interface mainly involves the setting of factory mode and all the settings such as list and control logs. **The operation of restoring capture logs, personnel list database and control logs is irreversible, so please be careful.**Parts that do not need to be recovered can be left out and partial recovery is supported.

5.4.1 Restore Factory Settings

1. Description

   The specified part will restore the factory settings， **“Name and ip address” or “Capture record” or “List registration record” or “Control record” or “RTSP” will be restored to the factory! “RTSP” to restore the factory application software will reboot!**

2. API Description

Items	Description
Operator	`SetFactoryDefault`
Request URL	`http://<server ipaddr>/action/SetFactoryDefault` (Where `<server ipaddr>` is the device IP, for example:192.168.1.10)

3. Explanation of the Paragraph of the Requested Message

   Parameter information(Note: `Optional` is optional):

Key	Type	Values	Description
operator	string	SetFactoryDefault	Setting up the device to restore the factory
info	json object		Concrete content
DefaltAll	int (optional)	0~1	Restore all factory settings 0: No factory restoration 1: Factory restoration
DefaltDoorSet	int (optional)	0~1	Door open condition whether to restore the factory 0: No factory restoration 1: Factory restoration
DefaltSoundSet	int (optional)	0~1	Whether the sound settings are restored to the factory 0: No factory restoration 1: Factory restoration
DefaltNetPar	int (optional)	0~1	Whether the network parameters are restored to the factory 0: No factory restoration 1: Factory restoration
DefaltCenterPar	int (optional)	0~1	Whether the center connection parameters are restored to the factory 0: No factory restoration 1: Factory restoration
DefaltCapture	int (optional)	0~1	Capture logs + control logs whether to restore the factory 0: No factory restoration 1: Factory restoration
DefaltLog	int (optional)	0~1	System Logs whether to restore the factory 0: No factory restoration 1: Factory restoration
DefaltPerson	int (optional)	0~1	Personnel list library + control logs whether to restore the factory 0: No factory restoration 1: Factory restoration

Key	Type	Values	Description
DefaltRecord	int (optional)	0~1	Control logs whether to restore the factory 0: No factory restoration 1: Factory restoration
DefaltMaintainTime	int (optional)	0~1	System maintenance time whether to restore the factory 0: No factory restoration 1: Factory restoration
DefaltSystemSettings	int (optional)	0~1	System parameters: ID card reader type; whether to record the capture, ID card, control logs and other parameters whether to restore the factory 0: No factory restoration 1: Factory restoration
DefaltEncParam	int (optional)	0~1	Video Parameter whether to restore the factory 0: No factory restoration 1: Factory restoration
DefaltRtsp	int (optional)	0~1	RTSP parameter whether to restore the factory 0: No factory restoration 1: Factory restoration
DefaltHttp	int (optional)	0~1	HTTP parameter whether to restore the factory 0: No factory restoration 1: Factory restoration
DefaltMqtt	int (optional)	0~1	MQTT parameter whether to restore the factory 0: No factory restoration 1: Factory restoration
DefaltWifi	int (optional)	0~1	Wifi parameter whether to restore the factory 0: No factory restoration 1: Factory restoration

Key	Type	Values	Description
DefaltMobile	int (optional)	0~1	4G parameter whether to restore the factory 0: No factory restoration 1: Factory restoration

4. Example of Request Message

```
URL of the request:http://192.168.2.10/action/SetFactoryDefault
Content of the request:
{
    "operator": "SetFactoryDefault",
    "info": {
        "DefaltDoorSet": 1,
        "DefaltSoundSet": 1,
        "DefaltNetPar": 0,
        "DefaltCenterPar": 0,
        "DefaltCapture": 0,
        "DefaltLog": 0,
        "DefaltPerson": 0,
        "DefaltRecord": 0,
        "DefaltMaintainTime": 0,
        "DefaltSystemSettings": 0,
        "DefaltEnterIPC": 0,
        "DefaltServerBasicPara": 0,
        "DefaltWorktype": 0
    }
}
```

5. Explanation of the Paragraph of the Reply Message

   Parameter information(Note: `Optional` is optional):

Key	Type	Values	Description
operator	string	SetFactoryDefault	Restore factory settings
info	JSON Object		Concrete content
code	int		Command execution error code 200-successes,see also 7.9 Restore Factory Settings Error Codes
Result	String	“Ok”/“Fail”	Operating result
Detail	String (optional)		Error message when Result is "Fail"

6. Example of Reply Message

```
{
    "operator": "SetFactoryDefault",
    "code": 200,
    "info": {
        "Result": "Ok"
    }
}
```

5.5 Device Keep Open Setting

1. Description

   Keep the device always open state, at this time, swipe the card, identification will not close the door, can be used for fire escape and other scenarios.

2. API Description

Items	Description
Operator	`SetKeepOpen`
Request URL	`http://<server ipaddr>/action/SetKeepOpen` (Where `<server ipaddr>` is the device IP, for example:192.168.1.10)

3. Explanation of the Paragraph of the Requested Message

   Parameter information(Note: `Optional` is optional):

Key	Type	Values	Description
operator	string	SetKeepOpen	Setting the device normally open
DeviceID	int		ID number of the device to be operated See 6.8.1 Getting Device Information
info	json object		Concrete content
KeepOpen	Int (optional)	0~2	Whether to set normally open 0:Not enabled 1:Door normally open 2:Door normally closed

4. Example of Request Message

```
URL of the request:http://192.168.2.10/action/SetKeepOpen
Content of the request:
{
    "operator": "SetKeepOpen",
    "DeviceID":1743725,
    "info": {
        "KeepOpen": 1
    }
}
```

5. Explanation of the Paragraph of the Reply Message

   Parameter information(Note: `Optional` is optional):

Key	Type	Values	Description
operator	string	SetKeepOpen	Setting the device normally open
info	JSON Object		Concrete content
code	int		Command execution error code 200-successes,see also 7.45 Device Keep Open Setting Error Codes
Result	String	“Ok”/“Fail”	Operating result
Detail	String (optional)		Error message when Result is "Fail"

6. Example of Reply Message

```
{
    "operator": "SetKeepOpen",
    "code": 463,
    "info": {
        "Result": "Fail",
        "Detail": "Key['KeepOpen'] value error"
    }
}
```

6、Device Parameters

6.1 Door Opening Conditions and Output Control Parameters

Door opening condition and output control parameter part mainly involves controlling the main parameter settings such as door opening condition and output control of the face machine, which can be adjusted according to the actual situation to realize different door opening condition to open the door.

6.1.1 Obtaining Door Opening Conditions and Outputting Control Parameters

1. Description

   The main purpose of this interface is to obtain the main parameters involving the opening conditions and output control of the face machine.

2. API Description

Items	Description
Operator	`GetDoorCondition`
Request URL	`http://<server ipaddr>/action/GetDoorCondition` (Where `<server ipaddr>` is the device IP, for example:192.168.1.10)

3. Example of Request Message

```
URL of the request:http://172.168.2.10/action/GetDoorCondition
Content of the request:（空）
```

4. Explanation of the Paragraph of the Reply Message

   Parameter information(Note: `Optional` is optional):

Key	Type	Values	Description
operator	string	GetDoorCondition	Obtaining door opening conditions and outputting control parameters
info	JSON Object		Concrete content
FaceThreshold	int	50~100	Black and White List comparison threshold
IDCardThreshold	int	50~100	ID card comparison threshold
OpendoorWay	int	0~3	0:Face door opening method; 1:Remote door opening method; 2:Remote door opening or face method; 3:Face verification + remote door opening method
VerifyMode	int	1.1:Universal model 1~4 or 21~26； 2.Masks or thermometry model: Whether mask testing or temperature testing is required Support for mask detection or (|) operation 0x100 indicates a pass with mask detection; Supports body temperature detection or (|) operation 0x200 indicates a pass with temperature detection Support mask + body temperature detection or (|) operation 0x300 means with mask + temperature detection pass Example: 1:Universal model 1)Whitelist validation:1	Authentication Type 1: Whitelist Verification 2: ID Verification 3: Whitelist + ID Verification 4: Whitelist or ID Verification 5:Separate mask or separate body temperature or separate mask + body temperature (to be used in conjunction with whether or not to include mask detection or body temperature detection, mask or temperature measurement models) 21:RF Card Verification(**Built-in card swiping model**) 22:RF Card Verification + Whitelist Verification(**Built-in card swiping model**) 23:RF card authentication or white list authentication

Key	Type	Values	Description
		2)Wergand card + whitelist validation:25 2:Mask or temperature measuring machine 1)Separate mask validation pass (0x105=0x100&5=261) 2)Mask + whitelist validation (0x101=0x100&1=257) 3)Separate body temperature pass (0x205=0x200&5=517) 4)Body temperature + whitelist verification (0x201=0x200&1=513) 5)Mouthpiece + body temperature alone (0x305=0x300&5=773) 6)Mouthpiece + body temperature + Wigan swipe + whitelist verification (0x319=0x300&25=793) Only health code version support:Support QR code check 0x400 1)QR code + temperature detection(1536=0x400+0x300)	(built-in card model) 24:Wiegand Card Verification 25:Wiegand Card + Whitelist Verification 26:Wiegand Card or Whitelist Authentication The following door opening methods are only supported by the health code version: 47:QR code or whitelist or ID card 48:whitelist + QR code or whitelist + ID card 55 QR code or ID card or whitelist (to check the health code) 56 QR code or ID card to skip the swipe 57 whitelist + health code information
VerifyResetTime	int	1~10	Verify reset time (sec) The UI displays a reminder message as well as a time for the same person to validate the reset time without being recognized again

Key	Type	Values	Description
Wiegand	int (optional)	0~1	Wiegand protocol type 0: 26 bits 1: 34 bits 4: 26 bits(8+16): facility code+userid 5: 34 bits(8+24); 6: 26 bits(8+16 fill in separately); 7: 34 bits(8+24 fill in separately);
PublicMjCardNo	unsigned int (optional)		Public Access Card Number
AutoMjCardBgnNo	unsigned int (optional)		Starting card number for automatic generation of access card numbers
AutoMjCardEndNo	unsigned int (optional)		Automatic generation of end card numbers for access control card numbers
ControlType	int	0~2	Controls the way the door opener interfaces. 0:Wiegand interface 1:switching mode 2:Wiegand interface+switching mode
IOType	int	0~1	Door opening action when controlling the door opening interface method 0:close 1:turn off
IOStayTime	int		Open Door Hold Time （ms）
DoorContactTime	int (optional)		Door magnetic trigger hold time（s）
Endian	int	0~1	Card number size end read mode 0:Big Endian 1:Little Endian

Key	Type	Values	Description
CardMode	int	0~1	Card Number Method 0:decimal 1:hexadecimal
IsOutFF	int	0~1	Stranger Wiegand output FFFFFFFF (Wiegand opens door) 0:non-output 1:output
SnapResetTime	int	0~15（default 0）	Recognition interval (seconds):Waiting time to recognize the next person after the comparison is passed
IsMaskOK	int (optional)	0~1	Used when the mask function is supported 1:Check to allow passage without a mask 0:Uncheck the box to allow passage without a mask
KeepCardZero	int	0~1	Whether or not to report and display the 0 in front of the reserved card number
PasswordEnable	int (optional)	0~1	Whether to enable password detection (device must have password function)
PasswordType	int (optional)	0~1	Password Detection Method: 0:common password 1:personal password （The device needs to have a password function）
keyPassword	String (option)		When PasswordType=0, the common password is a 6-digit number (the device must be equipped with a password function).

5. Example of Reply Message

1. The door opening interface is switching.

```
{
    "operator": "GetDoorCondition",
    "info": {
        "FaceThreshold": 80,
        "IDCardThreshold": 50,
        "VerifyResetTime": 2,
        "OpendoorWay": 0,
        "VerifyMode": 1,
        "ControlType": 1,
        "IOType": 0,
        "IOStayTime": 6,
        "Endian": 0,
        "CardMode": 0,
        "SnapResetTime": 0
    }
}
```

2. The door opening interface is a Wiegand interface.

```
{
    "operator": "GetDoorCondition",
    "info": {
        "FaceThreshold": 80,
        "IDCardThreshold": 50,
        "VerifyResetTime": 2,
        "OpendoorWay": 0,
        "VerifyMode": 1,
        "Wiegand": 0,
        "ControlType": 0,
        "PublicMjCardNo": 2,
        "AutoMjCardBgnNo": 2,
        "AutoMjCardEndNo": 4,
        "IOStayTime": 6,
        "Endian": 0,
        "CardMode": 0
    }
}
```

3. The door opening interface is a Wergen + switching interface.

```
{
    "operator": "GetDoorCondition",
    "info": {
        "FaceThreshold": 80,
        "IDCardThreshold": 50,
        "OpendoorWay": 0,
        "VerifyMode": 1,
```

```
        "Wiegand": 0,
        "ControlType": 0,
        "PublicMjCardNo": 2,
        "AutoMjCardBgnNo": 2,
        "AutoMjCardEndNo": 4,
        "IOStayTime": 6,
        "Endian": 0,
        "CardMode": 0
    }
}
```

6.1.2 Setting of Door Opening Conditions and Output Control Parameters

1. Description

   This interface is used to set the main parameters involving the door opening conditions and output control of the face machine.

2. API Description

Items	Description
Operator	`SetDoorCondition`
Request URL	`http://<server ipaddr>/action/SetDoorCondition` (Where `<server ipaddr>` is the device IP, for example:192.168.1.10)

3. Explanation of the Paragraph of the Requested Message

   Parameter information(Note: `Optional` is optional):

Key	Type	Values	Description
operator	string	SetDoorCondition	Setting of door opening conditions and output control parameters
info	JSON Object		Concrete content
FaceThreshold	int (optional)	50~100	Black and White List comparison threshold
IDCardThreshold	int (optional)	50~100	ID card comparison threshold
OpendoorWay	int (optional)	0~3	0:Face door opening method; 1:Remote door opening method; 2:Remote door opening or face method; 3:Face verification + remote door opening method
VerifyMode	int (optional)	1.1.1:Universal model 1~4 or 21~26 ; 2.Masks or thermometry model: Whether mask testing or temperature testing is required Support for mask detection or (|) operation 0x100 indicates a pass with mask detection; Supports body temperature detection or (|) operation 0x200 indicates a pass with temperature detection Support mask + body temperature detection or (|) operation 0x300 means with mask + temperature detection pass Example: 1:Universal model	Authentication Type 1: Whitelist Verification 2: ID Verification 3: Whitelist + ID Verification 4: Whitelist or ID Verification 5:Separate mask or separate body temperature or separate mask + body temperature (to be used in conjunction with whether or not to include mask detection or body temperature detection, mask or temperature measurement models) 21:RF Card Verification(**Built-in card swiping model**) 22:RF Card Verification +

Key	Type	Values	Description
		1)Whitelist validation:1 2)Wergand card + whitelist validation:25 2:Mask or temperature measuring machine 1)Separate mask validation pass (0x105=0x100&5=261) 2)Mask + whitelist validation (0x101=0x100&1=257) 3)Separate body temperature pass (0x205=0x200&5=517) 4)Body temperature + whitelist verification (0x201=0x200&1=513) 5)Mouthpiece + body temperature alone (0x305=0x300&5=773) 6)Mouthpiece + body temperature + Wigan swipe + whitelist verification (0x319=0x300&25=793) Only health code version support:Support QR code check 0x400 1)QR code + temperature detection(1536=0x400+0x300)	Whitelist Verification(**Built-in card swiping model**) 23:RF card authentication or white list authentication (built-in card model) 24:Wiegand Card Verification 25:Wiegand Card + Whitelist Verification 26:Wiegand Card or Whitelist Authentication The following door opening methods are only supported by the health code version: 47:QR code or whitelist or ID card 48:whitelist + QR code or whitelist + ID card 55 QR code or ID card or whitelist (to check the health code) 56 QR code or ID card to skip the swipe 57 whitelist + health code information
VerifyResetTime	int (optional)	1~10	Verify reset time (sec) The UI displays a reminder message as well as a time for the same person to validate the reset time without being recognized again

Key	Type	Values	Description
Wiegand	int (optional)	0~1	Wiegand protocol type 0: 26 bits 1: 34 bits 4: 26 bits(8+16): facility code+userid 5: 34 bits(8+24); 6: 26 bits(8+16 fill in separately); 7: 34 bits(8+24 fill in separately) ;
PublicMjCardNo	unsigned int (optional)		Public Access Card Number
AutoMjCardBgnNo	unsigned int (optional)		Starting card number for automatic generation of access card numbers
AutoMjCardEndNo	unsigned int (optional)		Starting card number for automatic generation of access card numbers
ControlType	int (optional)	0~2	Controls the way the door opener interfaces. 0:Wiegand interface 1:switching mode 2:Wiegand interface+switching mode
IOType	int (optional)	0~1	Door opening action when controlling the door opening interface method 0:close 1:turn off
IOStayTime	int (optional)		Open Door Hold Time （ms）
DoorContactTime	int (optional)		Door Magnetic Trigger Hold Time（ms）
Endian	int (optional)	0~1	Card number size end read mode 0:Big Endian 1:Little Endian

Key	Type	Values	Description
CardMode	int (optional)	0~1	Card Number Method 0:decimal 1:hexadecimal
IsOutFF	int (optional)	0~1	Stranger Wegen output FFFFFFFF (Wegen opens door) 0:non-output 1:output
SnapResetTime	int (optional)	0~15（Default 0）	Recognition interval (seconds):Waiting time to recognize the next person after the comparison is passed
IsMaskOK	int (optional)	0~1	Used when the mask function is supported 1:Check to allow passage without a mask 0:Uncheck the box to allow passage without a mask
IsCardOK	int (optional)	0~1	Whether unregistered card numbers are allowed 0: Not allowed 1: Allowed
KeepCardZero	int(optional)	0~1	Whether or not to report and display the 0 in front of the reserved card number
PasswordEnable	int (optional)	0~1	Whether to enable password detection (device must have password function)
PasswordType	int (optional)	0~1	Password Detection Method: 0:common password 1:personal password （The device needs to have a password function）

Key	Type	Values	Description
keyPassword	String (option)		When PasswordType=0, the common password is a 6-digit number (the device must be equipped with a password function).

4. Example of Request Message

   The door opening interface is switching.

```
URL of the request:http://172.168.2.68/action/SetDoorCondition
Content of the request:
{
    "operator": "SetDoorCondition",
    "info": {
        "FaceThreshold": 90,
        "IDCardThreshold": 50,
        "OpendoorWay": 0,
        "VerifyMode": 1,
        "ControlType": 1,
        "IOType": 0,
        "IOStayTime": 200,
        "Endian": 0,
        "CardMode": 1,
        "SnapResetTime": 0
    }
}
```

   The door opening interface is a Wiegand interface.

```
URL of the request:http://172.168.2.68/action/SetDoorCondition
Content of the request:
{
    "operator": "SetDoorCondition",
    "info": {
        "FaceThreshold": 90,
        "IDCardThreshold": 50,
        "OpendoorWay": 0,
        "VerifyMode": 1,
        "ControlType": 0,
        "Wiegand": 0,
        "PublicMjCardNo": 2,
        "AutoMjCardBgnNo": 2,
        "AutoMjCardEndNo": 4,
        "IOStayTime": 200,
        "Endian": 0,
        "CardMode": 1
```

```
    }
}
```

The door opening interface is a Wergen + switching interface.

```
URL of the request:http://172.168.2.68/action/SetDoorCondition
Content of the request:
{
    "operator": "SetDoorCondition",
    "info": {
        "FaceThreshold": 90,
        "IDCardThreshold": 50,
        "OpendoorWay": 0,
        "VerifyMode": 1,
        "ControlType": 0,
        "Wiegand": 0,
        "PublicMjCardNo": 2,
        "AutoMjCardBgnNo": 2,
        "AutoMjCardEndNo": 4,
        "IOStayTime": 200,
        "Endian": 0,
        "CardMode": 1
    }
}
```

5. Explanation of the Paragraph of the Reply Message

   Parameter information(Note: `Optional` is optional):

Key	Type	Values	Description
operator	string	SetDoorCondition	Setting of door opening conditions and output control parameters
code	int		Command execution error code 200-successes,see also7.1 Setting of Door Opening Conditions and Output Control Parameters Error Codes
info	JSON Object		Concrete content
Result	string		Operating result
Detail	string (optional)		Error message when Result is "Fail"

6. Example of Reply Message

   Command executed successfully:

```
{
    "operator": "SetDoorCondition",
    "code": 200,
    "info": {
        "Result": "Ok"
    }
}
```

Command execution failure:

```
{
    "operator": "SetDoorCondition",
    "code": 462,
    "info": {
        "Result": "Fail",
        "Detail": "Parameter error"
    }
}
```

6.1.3 Introduction to Door Opening Methods

At present, the face recognition machine supports the following door opening methods: face door opening method, remote door opening method, remote door opening or face method, face verification + remote door opening method.

0:Face opening mode:Face recognition all-in-one machine will push the stranger capture information and authentication comparison information, after passing the comparison, it will directly execute the door opening and UI prompting information, in this mode, calling the remote door opening instruction will be invalid.

1:Remote door opening mode: face recognition all-in-one machine only push the stranger capture information, do not push the authentication comparison information, **does not produce a control log**, the comparison through the door will not be directly executed by opening the door and the UI information prompts, this mode, you can call the remote door opening instructions, this mode is mainly used for the front-end capture, the back-end comparison, and the back-end implementation of the remote door opening.

2:Remote door opening or face mode:face recognition all-in-one machine will push the stranger capture information and authentication comparison information, and directly execute the door opening and UI prompt information after the comparison passes, in this mode, you can call the remote door opening instruction.

3:Face verification + remote door opening mode: face recognition all-in-one machine will push the stranger capture information and authentication comparison information, the comparison through the door will not be directly implemented to open the door and UI information prompts, open the door information and UI prompts need to be in the reply to the authentication push packet received in response to open the door information and UI prompts. This mode requires HTTP settings to be used **in the** `Continue Transmitting After Disconnection` **mode** and **the authentication subscription is turned on**. At this time, the face recognition all-in-one machine in the case of matching passes, the authentication matching information will be pushed to

the HTTP subscription address (platform), the platform needs to receive the authentication subscription of the push information, and then reply to the `Continue Transmitting After Disconnection` json data to confirm the packet directly with the open door and prompt information, the all-in-one machine to directly implement the open door and the UI information prompts, do not need to call the remote open the door command. In this mode, you need to pay attention to exclude whether the push authentication data is `Continue Transmitting After Disconnection` authentication data, if it is `Continue Transmitting After Disconnection` data and remote door opening authentication data, then the reply packet does not need to perform the door opening action again.

Example of an information confirmation packet received from an authentication subscription using Face Verification + Remote Door Opening when `Continue Transmitting After Disconnection` is enabled:

Parameter information(Note:`Optional` is optional):

Key	Type	Values	Description
code	int	200	Mandatory fields in `Continue Transmitting After Disconnection` mode
desc	string	OK	Mandatory fields in `Continue Transmitting After Disconnection` mode
openDoor	int(optional)	0/1	Door opening action 0:Do not open door 1:Open door
showInfo	string(optional)	64 bytes (including terminator)	UI Tips

Reply need to open door + UI tips:

```
{
    "code": 200,
    "desc": "OK",
    "openDoor": 1,
    "showInfo": "XX, please"
}
```

Reply no need to open door + UI tips:

```
{
    "code": 200,
    "desc": "OK",
    "openDoor": 0,
    "showInfo": "XX is not authorized to pass"
}
```

6.2 Prompt Sound and Interface Display Parameters

This interface is used for the sound playback of the face machine, display of UI elements and other major parameter settings.

6.2.1 Getting the Sound of the Prompt and the Parameters of the Interface Display

1. Description

   Get the main parameters for controlling the sound playback control of the face machine, display of UI elements, and so on.

2. API Description

Items	Description
Operator	GetSound
Request URL	`http://<server ipaddr>/action/GetSound` (Where `<server ipaddr>` is the device IP, for example:192.168.1.10)

3. Example of Request Message

```
URL of the request:http://172.168.2.10/action/GetSound
Content of the request:（NULL）
```

4. Explanation of the Paragraph of the Reply Message

   Parameter information(Note: `Optional` is optional):

Key	Type	Values	Description
operator	string	GetSound	Getting the sound of the prompt and the parameters of the interface display
info	JSON Object		Concrete content
VerifySuccAudio	int	0~1	Whether or not to broadcast sound after successful authentication 0: No broadcast 1: Broadcast
VerifyFailAudio	int	0~1	Whether to broadcast sound when authentication fails 0: No broadcast 1: Broadcast
RemoteCtrlAudio	int	0~1	Remote control of voice announcements 0:No broadcast 1: Broadcast
Volume	int	5~100 (Increase every 5 values)	Volume value
VerifySuccGuiTip	int	0~1	Whether or not the authentication success screen prompts 0: no prompt 1: prompt
VerifyFailGuiTip	int	0~1	Whether or not the authentication failure screen prompts 0: No prompt 1: Prompt
UnregisteredGuiTip	int	0~1	Is the list not registered screen prompted 0: Not prompted 1: Prompted
IPHide	int	0~1	Whether the interface IP is hidden or not 0: Not hidden 1: Hidden

Key	Type	Values	Description
IsShowName	int	0~2	Successful match, name prompts whether to display 0: not displayed 1: displayed 2: partially displayed (hide the latter part)
DisplayPicture	int	0~1	Images prompted by successful comparison 0: Captured image 1: Registered image
IsShowDeviceID	int	0~1	Whether to display the local ID 0: Do not show 1: Show
IsShowPersonNum	int	0~1	Whether to display the number of registered lists 0: Not displayed 1: Displayed
VerifyTipContent	string	User-defined	Successful authentication UI tips (length of 32 bytes or less, not too long, or lead to incomplete UI display) Default: “Access granted!”
UnregisterTipContent	string	User-defined	List Unregistered UI tips (32 bytes or less in length, otherwise it causes the UI to display incomplete) Default: “Stranger!”
BlacklistTipContent	String	User-defined	Authentication Failure UI Text Alert (within 32 bytes in length, otherwise it will cause the UI display to be incomplete) Default: “Access denied!”
CompanyName	String	User-defined	Company name (length of 64 bytes or less, the actual length can be displayed with reference to the UI display parameters)**Supported by some models**
IDCardNumHide	int (optional)	0~1	Whether to hide the card number in the interface**(Built-in card models)** 0: Not hidden 1: Hidden

Key	Type	Values	Description
ICCardNumHide	int (optional)	0~1	Whether to hide the IC card number in the interface**(Supported except for built-in card models)** 0: Not hidden 1: Hidden
TempLowAudio	int (optional)	0~1	Whether to turn on low-temperature broadcasting
TempDistance	int (optional)	0~1	Whether to turn on the temperature measurement distance prompt
bIsShowTamperAlarm	int (optional)	0~1	Whether tamper-evident is configured (requires hardware support)
CompanyLogoShow	int (optional)	0~1	Whether to display the company logo
CompanyLogo_X	int (optional)		When CompanyLogoShow=1, the horizontal coordinate of the logo.
CompanyLogo_Y	int (optional)		When CompanyLogoShow=1, the vertical coordinate of the logo.
CompanyLogo_H	int (optional)		When CompanyLogoShow=1, the height of the logo.
CompanyLogo_W	int (optional)		When CompanyLogoShow=1, the width of the logo.

5. Example of Reply Message

```
{
    "operator": "GetSound",
    "info": {
        "VerifySuccAudio": 1,
        "VerifyFailAudio": 1,
        "RemoteCtrlAudio": 0,
        "Volume": 80,
        "VerifySuccGuiTip": 1,
        "VerifyFailGuiTip": 1,
        "UnregisteredGuiTip": 1,
        "IPHide": 0,
        "IsShowName": 1,
        "DisplayPicture": 0,
        "ICCardNumHide": 0,
        "VerifyTipContent":"Access granted!",
        "UnregisterTipContent":"Stranger!",
```

```
        "BlacklistTipContent":"Access denied!",
        "CompanyLogoShow": 0
    }
}
```

6.2.2 Setting the Prompt Sound and Interface Display Parameters

1. Description

   This interface is used for sound playback control of the face machine, display of UI elements and other parameter settings.

2. API Description

Items	Description
Operator	`SetSound`
Request URL	`http://<server ipaddr>/action/SetSound` (Where `<server ipaddr>` is the device IP, for example:192.168.1.10)

3. Explanation of the Paragraph of the Requested Message

   Parameter information(Note: `Optional` is optional):

Key	Type	Values	Description
operator	string	SetSound	Setting the prompt sound and interface display parameters
info	JSON Object		Concrete content
VerifySuccAudio	int (optional)	0~1	Whether or not to broadcast sound after successful authentication 0: No broadcast 1: Broadcast
VerifyFailAudio	int (optional)	0~1	Whether to broadcast sound when authentication fails 0: No broadcast 1: Broadcast
RemoteCtrlAudio	int (optional)	0~1	Remote control of voice announcements 0: No broadcast 1: Broadcast
Volume	int (optional)	5~100 (Increase every 5 values)	Sound volume value
VerifySuccGuiTip	int (optional)	0~1	Whether or not the authentication success screen prompts 0: no prompt 1: prompt
VerifyFailGuiTip	int (optional)	0~1	Whether or not the authentication failure screen prompts 0: No prompt 1: Prompt
UnregisteredGuiTip	int (optional)	0~1	Is the list not registered screen prompted 0: Not prompted 1: Prompted
IPHide	int (optional)	0~1	Whether the interface IP is hidden or not 0: Not hidden 1: Hidden

Key	Type	Values	Description
IsShowName	int (optional)	0~2	Successful match, name prompts whether to display 0: not displayed 1: displayed 2: partially displayed (hide the latter part)
DisplayPicture	int (optional)	0~1	Images prompted by successful comparison 0: Captured image 1: Registered image
IDCardNumHide	int (optional)	0~1	Whether to hide the card number in the interface**(Built-in card models)** 0: Not hidden 1: Hidden
ICCardNumHide	int (optional)	0~1	Whether to hide the IC card number in the interface**(Supported except for built-in card models)** 0: Not hidden 1: Hidden
IsShowDeviceID	int (optional)	0~1	Whether to display the local ID 0: Do not show 1: Show
IsShowPersonNum	int (optional)	0~1	Whether to display the number of registered lists 0: Not displayed 1: Displayed
VerifyTipContent	string (optional)	User-defined	Successful authentication UI tips (length of 32 bytes or less, not too long, or lead to incomplete UI display) Default: “Access granted!”
UnregisterTipContent	string (optional)	User-defined	List Unregistered UI tips (32 bytes or less in length, otherwise it causes the UI to display incomplete) Default: “Stranger!”
BlacklistTipContent	String (optional)	User-defined	Authentication Failure UI Text Alert (within 32 bytes in length, otherwise it will cause the UI display to be incomplete) Default: “Access denied!”

Key	Type	Values	Description
CompanyName	String (optional)	User-defined	Company name (length of 64 bytes or less, the actual length can be displayed with reference to the UI display parameters)**Supported by some models**
TempLowAudio	int (optional)	0~1	Whether to turn on low-temperature broadcasting
TempDistance	int (optional)	0~1	Whether to turn on the temperature measurement distance prompt
bIsShowTamperAlarm	int (optional)	0~1	Whether tamper-evident is configured (requires hardware support)
CompanyLogoShow	int (optional)	0~1	Whether to display the company logo 0:Do not show 1:Show
CompanyLogo_X	int (optional)	User-defined	When CompanyLogoShow=1, the horizontal coordinate of the logo.
CompanyLogo_Y	int (optional)	User-defined	When CompanyLogoShow=1, the vertical coordinate of the logo.
CompanyLogo_H	int (optional)	User-defined	When CompanyLogoShow=1, the height of the logo.
CompanyLogo_W	int (optional)	User-defined	When CompanyLogoShow=1, the width of the logo.

4. Example of Request Message

```
URL of the request:http://172.168.2.10/action/SetSound
Content of the request:
{
    "operator": "SetSound",
    "info":{
        "VerifySuccAudio":1,
        "VerifyFailAudio":1,
        "RemoteCtrlAudio":1,
        "Volume":50,
        "VerifySuccGuiTip":1,
        "VerifyFailGuiTip":1,
        "UnregisteredGuiTip":1,
        "IPHide": 0,
        "IsShowName": 1,
        "DisplayPicture": 0,
        "ICCardNumHide": 0,
```

```
        "VerifyTipContent":"Access granted!",
        "UnregisterTipContent":"Stranger!",
        "BlacklistTipContent":"Access denied!"
    }
}
```

5. Explanation of the Paragraph of the Reply Message

   Parameter information(Note: `Optional` is optional):

Key	Type	Values	Description
operator	string	SetSound	Setting the prompt sound and interface display parameters
code	int		Command execution error code 200-successes,see also7.2 Setting the Prompt Sound and Interface Display Parameters Error Codes
info	JSON Object		Concrete content
Result	string		Operating result
Detail	string (optional)		Error message when Result is "Fail"

6. Example of Reply Message

```
{
    "operator": "SetSound",
    "code": 200,
    "info": {
        "Result": "Ok"
    }
}
```

6.3 Username and Password

Set the username and password for the Face Recognition All-in-One Machine.

6.3.1 Set Username and Password

1. Description

   Set the username and password for the Face Recognition All-in-One Machine.

2. API Description

Items	Description
Operator	`SetUserPwd`
Request URL	`http://<server ipaddr>/action/SetUserPwd` (Where `<server ipaddr>` is the device IP, for example:192.168.1.10)

3. Explanation of the Paragraph of the Requested Message

   Parameter information(Note: `Optional` is optional):

Key	Type	Values	Description
operator	string	SetUserPwd	Set the username and password for the Face Recognition All-in-One Machine.
info	JSON Object		Concrete content
User	string		Username, maximum length 64 character length (including terminator)
Pwd	string		Password, maximum length 64 character length (including terminator)

4. Example of Request Message

```
URL of the request:http://172.168.5.134/action/SetUserPwd
Content of the request:
{
    "operator":"SetUserPwd",
    "info":{
        "User":"admin",
        "Pwd":"admin"
    }
}
```

5. Explanation of the Paragraph of the Reply Message

   Parameter information(Note: `Optional` is optional):

Key	Type	Values	Description
operator	string	SetUserPwd	Set user name and password
info	JSON Object		Concrete content
code	int		Command execution error code 200-successes,see also7.4 Set Username and Password Error Codes
Result	String	“Ok”/“Fail”	Operating result
Detail	String (optional)		Error message when Result is "Fail"

6. Example of Reply Message

```
{
    "operator": "SetUserPwd",
    "code": 200,
    "info": {
        "Result": "Ok"
    }
}
```

6.4 Network Parameters

The main parameters are IP address of the device, IP subnet mask, IP gateway, web service port number, whether DHCP is enabled or not, default route and so on. Changing IP and other parameters will reboot the device. Currently do not reply to the result after setting.

6.4.1 Getting Network Parameters

1. Description

   Get the network parameters of the device.

2. API Description

Items	Description
Operator	`GetNetParam`
Request URL	`http://<server ipaddr>/action/GetNetParam` (Where `<server ipaddr>` is the device IP, for example:192.168.1.10)

3. Example of Request Message

```
URL of the request:http://192.168.2.10/action/GetNetParam
Content of the request:（NULL）
```

4. Explanation of the Paragraph of the Reply Message

   Parameter information(Note:`Optional`is optional):

Key	Type	Values	Description
operator	string	GetNetParam	Getting network parameters
info	JSON Object		Concrete content
IPAddr	string		Device IP address (up to 64 bytes)
Submask	string		Device IP subnet mask (up to 32 bytes)
Gateway	string		Device IP gateway (up to 32 bytes)
ListenPort	int		Data service port
WebPort	int		Web service port
DHCP	int	0~1	DHCP 0: disable 1: enable (**When enabled, note that it is recommended that the UI interface does not hide the IP**) Default 0
DNS1	string		DNS1
DNS2	string		DNS2
NetArp	int	0~1	Whether to detect ip conflicts regularly
DefRoute	int (optional)	0~1	Default Route (**WiFi/4G version specific**) 0: Wired Route 1: WiFi Route 2: 4G

5. Example of Reply Message

```
{
    "operator": "GetNetParam",
    "info": {
        "IPAddr": "192.168.2.10",
        "Submask": "255.255.255.0",
        "Gateway": "192.168.2.1",
        "ListenPort": 5000,
        "WebPort": 80
    }
}
```

6.4.2 Setting Network Parameters

1. Description

   Setting network parameters.

2. API Description

Items	Description
Operator	`SetNetParam`
Request URL	`http://<server ipaddr>/action/SetNetParam` (Where `<server ipaddr>` is the device IP, for example:192.168.1.10)

3. Explanation of the Paragraph of the Requested Message

   Parameter information(Note: `Optional` is optional):

Key	Type	Values	Description
operator	string	SetNetParam	Setting network parameters.
info	JSON Object		Concrete content
IPAddr	string (optional)		Device IP address (up to 64 bytes)
Submask	string (optional)		Device IP subnet mask (up to 32 bytes)
Gateway	string (optional)		Device IP gateway (up to 32 bytes)
ListenPort	int (optional)		Data service port
WebPort	int (optional)		Web service port
DHCP	int (optional)	0~1	DHCP 0: disable 1: enable (**When enabled, note that it is recommended that the UI interface does not hide the IP**) Default 0
NetArp	int (optional)	0~1	Whether to detect ip conflicts regularly
DefRoute	int (optional)	0~2	Default Route (**WiFi/4G version specific**) 0: Wired Route 1: WiFi Route 2: 4G

4. Example of Request Message

```
URL of the request:http://192.168.2.10/action/SetNetParam
Content of the request:
{
    "operator": "SetNetParam",
    "info": {
        "IPAddr": "172.168.5.250",
        "Submask": "255.255.255.0",
        "Gateway": "172.168.5.1",
        "ListenPort": "5000",
        "WebPort": "80"
    }
}
```

5. Explanation of the Paragraph of the Reply Message

   Parameter information(Note: `Optional` is optional):

Key	Type	Values	Description
operator	string	SetNetParam	Setting network parameters.
info	JSON Object		Concrete content
code	int		Command execution error code 200-successes,see also7.5 Setting Network Parameters Error Codes
Result	String	“Ok”/“Fail”	Operating result
Detail	String (optional)		Error message when Result is "Fail"

6. Example of Reply Message

```
{
    "operator": "SetNetParam",
    "code": 200,
    "info": {
        "Result": "Ok"
    }
}
```

6.5 Time Parameters

Get or set the device's time parameters.

6.5.1 Getting the Time Parameters

1. Description

   Get the current time of the device.

2. API Description

Items	Description
Operator	`GetSysTime`
Request URL	`http://<server ipaddr>/action/GetSysTime` (Where `<server ipaddr>` is the device IP, for example:192.168.1.10)

3. Example of Request Message

```
URL of the request:http://192.168.2.10/action/GetSysTime
Content of the request:（NULL）
```

4. Explanation of the Paragraph of the Reply Message

   Parameter information(Note: `Optional` is optional):

Key	Type	Values	Description
operator	string	GetSysTime	Get the time parameter
info	JSON object		Concrete content
Year	int	1900~	Year
Month	int	1~12	Month
Day	int	1~31	Day
Hour	int	0~23	Hour
Minute	int	0~59	Minute
Second	int	0~59	Second

5. Example of Reply Message

```
{
    "operator": "GetSysTime",
    "info": {
        "Year": 2022,
        "Month": 12,
        "Day": 14,
        "Hour": 10,
        "Minute": 10,
        "Second": 18
    }
}
```

6.5.2 Setting the Time Parameters

1. Description

   Set the system to the specified time.

2. API Description

Items	Description
Operator	`SetSysTime`
Request URL	`http://<server ipaddr>/action/SetSysTime` (Where `<server ipaddr>` is the device IP, for example:192.168.1.10)

3. Explanation of the Paragraph of the Requested Message

   Parameter information(Note: `Optional` is optional):

Key	Type	Values	Description
operator	string	SetSysTime	Setting time parameters
info	Json object		Concrete content
Year	int	1900~	Year
Month	int	1~12	Month
Day	int	1~31	Day
Hour	int	0~23	Hour
Minute	int	0~59	Minute
Second	int	0~59	Second

4. Example of Request Message

```
URL of the request:http://192.168.2.10/action/SetSysTime
Content of the request:
{
    "operator":"SetSysTime",
    "info":{
        "Year":"2017",
        "Month":"1",
        "Day":"1",
        "Hour":"0",
        "Minute":"0",
        "Second":"0"
    }
}
```

5. Explanation of the Paragraph of the Reply Message

   Parameter information(Note: `Optional` is optional):

Key	Type	Values	Description
operator	string	SetSysTime	Setting time parameters
info	JSON Object		Concrete content
code	int		Command execution error code 200-successes,see also7.6 Setting the Time Parameters Error Codes
Result	String	“Ok”/“Fail”	Operating result
Detail	String (optional)		Error message when Result is "Fail"

6. Example of Reply Message

```
{
    "operator": "SetSysTime",
    "code": 200,
    "info": {
        "Result": "Ok"
    }
}
```

6.6 NTP Parameters

Get or set the NTP parameters of the device. Network Time Protocol (NTP) is an application layer protocol inside the TCP/IP family of protocols used to synchronize clocks between clients and servers to provide highly accurate time correction.

6.6.1 Getting NTP Parameters

1. Description

   Get the NTP parameters of the device.

2. API Description

Items	Description
Operator	`GetSysNTP`
Request URL	`http://<server ipaddr>/action/GetSysNTP` (Where `<server ipaddr>` is the device IP, for example:192.168.1.10)

3. Example of Request Message

```
URL of the request:http://192.168.2.10/action/GetSysNTP
Content of the request:（NULL）
```

4. Explanation of the Paragraph of the Reply Message

   Parameter information(Note:`Optional` is optional):

Key	Type	Values	Description
operator	string	GetSysNTP	Get the NTP parameters
info	json object		Concrete content
NtpEnable	int	0~1	Whether to Enable NTP Timing 0: Not Enabled 1: Enabled
NtpUrl	string		NTP server ip address (up to 64 bytes)
NtpPort	int		NTP server port

Key	Type	Values	Description
NtpTimeZone	int	1~30	Time Zone 1: (GMT-12:00) International Date Line 2: (GMT-11:00) Midway, Samoa 3:(GMT-10:00) Hawaii 4:(GMT-09:00) Alaska 5:(GMT-08:00) Pacific(USA and Canada) 6:(GMT-07:00) Mountain(USA and Canada) 7:(GMT-06:00) Central (USA and Canada) 8:(GMT-05:00) Eastern(USA and Canada) 9:(GMT-04:30) Caracas 10:(GMT-04:00) Atlantic(Canada) 11:(GMT-03:30) Newfoundland 12:(GMT-03:00) Georgetown,Brasilia 13:(GMT-02:00) Mid-Atlantic 14:(GMT-01:00) Cape Verde islands, Azores 15:(GMT+00:00) Dublin, Edinburgh, London 16:(GMT+01:00) Amsterdam, Berlin, Rome, Paris 17:(GMT+02:00) Athens, Jerusalem, Istanbul 18:(GMT+03:00) Baghdad, Kuwait, Moscow 19:(GMT+03:30) Teheran 20:(GMT+04:00) Caucasian standard time 21:(GMT+04:30) Kabul 22:(GMT+05:00) Islamabad, Karachi, Tashkent 23:(GMT+05:30) Madras, Mumbai, New Delhi 24:(GMT+05:45) Katmandu 25:(GMT+06:00) Almaty, Novosibirsk, Dhaka 26:(GMT+06:30) Rangoon 27:(GMT+07:00) Bangkok, Hanoi, Jakarta 28:(GMT+08:00) Beijing, urumqi, Singapore 29:(GMT+09:00) Seoul, Tokyo, Osaka, Sapporo 30:(GMT+09:30) Adelaide, Darwin 31:(GMT+10:00) Melbourne, Sydney, Canberra 32:(GMT+11:00) Magadan, Solomon islands 33:(GMT+12:00) Oakland, Wellington 34:(GMT+13:00) Nukualofa
NtpInterval	unsigned int		Ntp calibration time interval in hours

5. Example of Reply Message

```
{
    "operator": "GetSysNTP",
    "info": {
        "NtpEnable": 1,
        "NtpUrl": "clock.isc.org",
        "NtpPort": 123,
        "NtpTimeZone": 25,
        "NtpInterval": 2
    }
}
```

6.6.2 Setting NTP Parameters

1. Description

   Set the NTP parameters of the device.

2. API Description

Items	Description
Operator	`SetSysNTP`
Request URL	`http://<server ipaddr>/action/SetSysNTP` (Where `<server ipaddr>` is the device IP, for example:192.168.1.10)

3. Explanation of the Paragraph of the Requested Message

   Parameter information(Note: `Optional` is optional):

Key	Type	Values	Description
operator	string	SetSysNTP	Set the NTP parameters
info	json object		Concrete content
NtpEnable	int	0~1	Whether to Enable NTP Timing 0: Not Enabled 1: Enabled
NtpUrl	string		NTP server ip address (up to 64 bytes)
NtpPort	int		NTP server port

Key	Type	Values	Description
NtpTimeZone	int	1~30	Time Zone 1: (GMT-12:00) International Date Line 2: (GMT-11:00) Midway, Samoa 3:(GMT-10:00) Hawaii 4:(GMT-09:00) Alaska 5:(GMT-08:00) Pacific(USA and Canada) 6:(GMT-07:00) Mountain(USA and Canada) 7:(GMT-06:00) Central (USA and Canada) 8:(GMT-05:00) Eastern(USA and Canada) 9:(GMT-04:30) Caracas 10:(GMT-04:00) Atlantic(Canada) 11:(GMT-03:30) Newfoundland 12:(GMT-03:00) Georgetown,Brasilia 13:(GMT-02:00) Mid-Atlantic 14:(GMT-01:00) Cape Verde islands, Azores 15:(GMT+00:00) Dublin, Edinburgh, London 16:(GMT+01:00) Amsterdam, Berlin, Rome, Paris 17:(GMT+02:00) Athens, Jerusalem, Istanbul 18:(GMT+03:00) Baghdad, Kuwait, Moscow 19:(GMT+03:30) Teheran 20:(GMT+04:00) Caucasian standard time 21:(GMT+04:30) Kabul 22:(GMT+05:00) Islamabad, Karachi, Tashkent 23:(GMT+05:30) Madras, Mumbai, New Delhi 24:(GMT+05:45) Katmandu 25:(GMT+06:00) Almaty, Novosibirsk, Dhaka 26:(GMT+06:30) Rangoon 27:(GMT+07:00) Bangkok, Hanoi, Jakarta 28:(GMT+08:00) Beijing, urumqi, Singapore 29:(GMT+09:00) Seoul, Tokyo, Osaka, Sapporo 30:(GMT+09:30) Adelaide, Darwin 31:(GMT+10:00) Melbourne, Sydney, Canberra 32:(GMT+11:00) Magadan, Solomon islands 33:(GMT+12:00) Oakland, Wellington 34:(GMT+13:00) Nukualofa
NtpInterval	unsigned int		Ntp calibration time interval in hours.

4. Example of Request Message

```
URL of the request:http://192.168.2.10/action/SetSysNTP
Content of the request:
{
    "operator":"SetSysNTP",
    "info":{
        "NtpEnable":0,
        "NtpUrl":"time.windows.com",
        "NtpPort":123,
        "NtpTimeZone":14,
        "NtpInterval":1
    }
}
```

5. Explanation of the Paragraph of the Reply Message

   Parameter information(Note: `Optional` is optional):

Key	Type	Values	Description
operator	string	SetSysNTP	Set the NTP parameters
info	JSON Object		Concrete content
code	int		Command execution error code 200-successes,see also7.7 设置NTP参数错误码
Result	String	“Ok”/“Fail”	Operating result
Detail	String (optional)		Error message when Result is "Fail"

6. Example of Reply Message

```
{
    "operator": "SetSysNTP",
    "code": 200,
    "info": {
        "Result": "Ok"
    }
}
```

6.7 Central Connection Parameters

Get and set the parameters of the device central connection.

6.7.1 Get Central Connection Parameters

1. Description

   Get central connection parameters.

2. API Description

Items	Description
Operator	`GetCenter`
Request URL	`http://<server ipaddr>/action/GetCenter` (Where `<server ipaddr>` is the device IP, for example:192.168.1.10)

3. Example of Request Message

```
URL of the request:http://192.168.2.10/action/GetCenter
Content of the request:（NULL）
```

4. Explanation of the Paragraph of the Reply Message

   Parameter information(Note: `Optional` is optional):

Key	Type	Values	Description
operator	string	GetCenter	Get central connection parameters
info	json object		Concrete content
ConnectCenter	int	0~1	Whether to enable the center connection 0: Disable 1: Enable
CenterUrl	string		Central server ip address (max 64 bytes)
CenterPort	int		Central server port
CenterUser	string		Username, maximum length 64 character length (including terminator)
CenterPwd	string		Password, maximum length 64 character length (including terminator)
SendSnapImage	int	0~1	Whether to send live images 0: Do not send 1: Do send
SendVerifyResult	int	0~1	Whether to send authentication results 0: Do not send 1: Do send

5. Example of Reply Message

```
{
    "operator": "GetCenter",
    "info": {
        "ConnectCenter": 1,
        "CenterUrl": "172.168.5.99",
        "CenterPort": 6200,
        "CenterUser": "admin",
        "CenterPwd": "admin",
        "SendSnapImage": 1,
        "SendVerifyResult": 0
    }
}
```

6.7.2 Setting the Central Connection Parameters

1. Description

   Set the central connection parameters. Parameters not included in the request message follow the original configuration by default.

2. API Description

Items	Description
Operator	`SetCenter`
Request URL	`http://<server ipaddr>/action/SetCenter` (Where `<server ipaddr>` is the device IP, for example:192.168.1.10)

3. Explanation of the Paragraph of the Requested Message

   Parameter information(Note: `Optional` is optional):

Key	Type	Values	Description
operator	string	SetCenter	Set the central connection parameters.
info	json object		Concrete content
ConnectCenter	int (optional)	0~1	Set the central connection parameters.
CenterUrl	string (optional)		Central server ip address (max 64 bytes)
CenterPort	int (optional)		Central server port
CenterUser	string (optional)		Username, maximum length 64 character length (including terminator)
CenterPwd	string (optional)		Password, maximum length 64 character length (including terminator)
SendSnapImage	int (optional)	0~1	Whether to send live images 0: Do not send 1: Do send
SendVerifyResult	int (optional)	0~1	Whether to send authentication results 0: Do not send 1: Do send

4. Example of Request Message

```
URL of the request:http://192.168.2.10 /action/SetCenter
Content of the request:
{
    "operator": "SetCenter",
    "info": {
        "ConnectCenter": 0,
        "CenterUrl": "192.168.2.10",
        "CenterPort": 6666,
        "CenterUser": "admin",
        "CenterPwd": "admin",
        "SendSnapImage": 1,
        "SendVerifyResult": 1
    }
}
```

5. Explanation of the Paragraph of the Reply Message

   Parameter information(Note: `Optional` is optional):

Key	Type	Values	Description
operator	string	SetCenter	Set the central connection parameters.
info	JSON Object		Concrete content
code	int		Command execution error code 200-successes,see also7.8 Setting the Central Connection Parameters Error Codes
Result	String	“Ok”/“Fail”	Operating result
Detail	String (optional)		Error message when Result is "Fail"

6. Example of Reply Message

```
{
    "operator": "SetCenter",
    "code": 200,
    "info": {
        "Result": "Ok"
    }
}
```

6.8 Device Information

Get and set device information.

6.8.1 Getting Device Information

1. Description

   Get device information.

2. API Description

Items	Description
Operator	`GetSysParam`
Request URL	`http://<server ipaddr>/action/GetSysParam` (Where `<server ipaddr>` is the device IP, for example:192.168.1.10)

3. Example of Request Message

```
URL of the request:http://192.168.2.10/action/GetSysParam
Content of the request:（NULL）
```

4. Explanation of the Paragraph of the Reply Message

   Parameter information(Note: `Optional` is optional):

Key	Type	Values	Description
operator	string	GetSysParam	Getting system device information
info	json object		Concrete content
Name	string		Device name, supports up to 64 characters in length (including terminator)
DeviceID	int		Device ID (not configurable)
Version	string		System software version (not configurable)

5. Example of Reply Message

```
{
    "operator": "GetSysParam",
    "info": {
        "Name": "FaceGate",
        "DeviceID": 1743725,
        "Version": "v8.52.11.0-L-M-I7-2.5W"
    }
}
```

6.8.2 Setting Device Information

1. Description

   Set the device information.

2. API Description

Items	Description
Operator	`SetSysParam`
Request URL	`http://<server ipaddr>/action/SetSysParam` (Where `<server ipaddr>` is the device IP, for example:192.168.1.10)

3. Explanation of the Paragraph of the Requested Message

   Parameter information(Note: `Optional` is optional):

Key	Type	Values	Description
operator	string	SetSysParam	Set the device information.
info	json object		Concrete content
Name	string (optional)		Device name, supports up to 64 characters in length (including terminator)

4. Example of Request Message

```
URL of the request:http://192.168.2.10/action/SetSysParam
Content of the request:
{
    "operator": "SetSysParam",
    "info": {
        "Name": "facegate_test"
    }
}
```

5. Explanation of the Paragraph of the Reply Message

   Parameter information(Note: `Optional` is optional):

Key	Type	Values	Description
operator	string	SetSysParam	Set the device information.
info	JSON Object		Concrete content
code	int		Command execution error code 200-successes,see also7.10 Setting Device Information Error Codes
Result	String	“Ok”/“Fail”	Operating result
Detail	String (optional)		Error message when Result is "Fail"

6. Example of Reply Message

```
{
    "operator": "SetSysParam",
    "code": 200,
    "info": {
        "Result": "Ok"
    }
}
```

6.9 Face Recognition Parameters

The face recognition parameters are mainly related to the recognition parameters of the Face Recognition All-in-One Machine. The recognition distance (corresponding to the minimum face pixel) can be modified through the interface. The X and Y values of the face detection area represent the starting position of the detection area of the video screen, and the width and height of the face detection area represent the width and height of the video screen detected based on X and Y. The width and height of the face detection area should be matched with the width and height of the face detection area. The face detection width and height should be used in conjunction with the X and Y values of the face detection area. When X>0, the face detection width should be less than (**maximum** face detection width - X). **Generally only the recognition distance (corresponding to the smallest pixel of the face) needs to be changed, and other parameters do not need to be changed again.**

6.9.1 Getting Face Recognition Parameters

1. Description

   Get the face recognition parameters.

2. API Description

Items	Description
Operator	`GetFaceParam`
Request URL	`http://<server ipaddr>/action/GetFaceParam` (Where `<server ipaddr>` is the device IP, for example:192.168.1.10)

3. Example of Request Message

```
URL of the request:http://192.168.2.10/action/GetFaceParam
Content of the request:（NULL）
```

4. Explanation of the Paragraph of the Reply Message

Key	Type	Values	Description
operator	string	GetFaceParam	Get the face recognition parameters.
info	JSON object		Concrete content
recDistance	Int	30~200(unit:cm)	recognition distance(cm)
faceMinPixel	Int	0~960	Minimum pixel of the face (deprecated, use recognition distance instead)
detectArea_x	Int	0~960(default 0)	Face detection area X start point (even)
detectArea_y	Int	0~1280(default 0)	Face detection area Y start point (even)
detectArea_w	Int	0~960 (default 960)	Face detection area width (even)
detectArea_h	string	0~1280 (default 1280)	Face detection area height (even)

5. Example of Reply Message

```
{
    "operator": "GetFaceParam",
    "info": {
        "recDistance": "200",
        "faceMinPixel": "122",
        "detectArea_x": 0,
        "detectArea_y": 0,
        "detectArea_w": 960,
        "detectArea_h": "1280"
    }
}
```

6.9.2 Setting Face Recognition Parameters

1. Description

   Set the face recognition parameters.

2. API Description

Items	Description
Operator	`SetFaceParam`
Request URL	`http://<server ipaddr>/action/SetFaceParam` (Where `<server ipaddr>` is the device IP, for example:192.168.1.10)

3. Explanation of the Paragraph of the Requested Message

   Parameter information(Note: `Optional` is optional):

Key	Type	Values	Description
operator	string	SetFaceParam	Set the face recognition parameters.
info	JSON object		Concrete content
recDistance	Int (optional)	30~200(unit:cm)	recognition distance(cm)
faceMinPixel	Int (optional)	0~960	Minimum pixel of the face (deprecated, use recognition distance instead)
detectArea_x	Int (optional)	0~960 (default 0)	Face detection area X start point (even)
detectArea_y	Int (optional)	0~1280 (default 0)	Face detection area Y start point (even)
detectArea_w	Int (optional)	0~960 (default 960)	Face detection area width (even)
detectArea_h	Int (optional)	0~1280 (default 1280)	Face detection area height (even)

4. Example of Request Message

```
URL of the request:http://192.168.2.10/action/SetFaceParam
Content of the request:
{
    "operator": "SetFaceParam",
    "info": {
        "recDistance": "100",
        "detectArea_x": 0,
        "detectArea_y": 0,
        "detectArea_w": 960,
        "detectArea_h": 1280
    }
}
```

5. Explanation of the Paragraph of the Reply Message

   Parameter information(Note:`Optional`is optional):

Key	Type	Values	Description
operator	string	SetFaceParam	Set the face recognition parameters.
info	JSON Object		Concrete content
code	int		Command execution error code 200-successes,see also 7.11 Setting Face Recognition Parameters Error Codes
Result	String	“Ok”/“Fail”	Operating result
Detail	String (optional)		Error message when Result is "Fail"

6. Example of Reply Message

```
{
    "operator": "SetFaceParam",
    "code": 200,
    "info": {
        "Result": "Ok"
    }
}
```

6.10 Common Configuration Settings

Get or set common parameter configuration information, including black and white list comparison threshold, ID card comparison threshold, control type, device IP address, subnet mask, gateway and other parameters.

6.10.1 Getting Common Configuration Parameters

1. Description

   Get information about common configuration parameters.

2. API Description

Items	Description
Operator	`GetGeneralParam`
Request URL	`http://<server ipaddr>/action/GetGeneralParam` (Where `<server ipaddr>` is the device IP, for example:192.168.1.10)

3. Example of Request Message

```
URL of the request:http://172.168.2.10/action/GetGeneralParam
Content of the request:（NULL）
```

4. Explanation of the Paragraph of the Reply Message

   Parameter information(Note:`Optional`is optional):

Key	Type	Values	Description
operator	string	GetGeneralParam	Get information about common configuration parameters.
DeviceID	int		ID number of the device to be operated See 6.8.1 Getting Device Information
info	json object		Concrete content
FaceThreshold	int	50 ~ 100	Black and White List comparison threshold
IDCardThreshold	int	50 ~ 100	ID card comparison threshold
ControlType	int	0~2	0:Switching; 1:Wiegand interface 26 bits; 2:Wiegand interface 34 bits
LiveThreshold	double	0~1	Live detection threshold (not settable)
LiveFrameNum	int		Number of consecutive frames for live body detection (not settable)
AutoRebootDay	int	0~8	Device AutoReboot Setting Date 0:No reboot; 1:Daily reboot; 2~8:Weekly AutoRebootDay-1 reboot
AutoRebootHour	int	0~23	Device Reboot Integral Time
Direction	int	0~2	Entrance/Exit Direction 0:Unidirectional 1:Entrance 2:Exit
IPAddr	string		Device Ip address (64 bytes)
Submask	string		Device subnet mask (32 bytes)
Gateway	string		Device gateway (32 bytes)
DNS	string		Device DNS (32 bytes)

5. Example of Reply Message

```
{
    "operator": "GetGeneralParam",
    "DeviceID": 1743725,
    "info": {
        "FaceThreshold": 85,
        "IDCardThreshold": 50,
        "ControlType": 0,
        "AutoRebootDay": 0,
        "AutoRebootHour": 0,
        "Direction": 0,
        "IPAddr": "172.168.5.134",
        "Submask": "255.255.255.0",
        "Gateway": "172.168.5.1",
        "DNS": "202.96.134.33"
    }
}
```

6.10.2 Setting Common Configuration Parameters

1. Description

   Get information about common configuration parameters.

2. API Description

Items	Description
Operator	`SetGeneralParam`
Request URL	`http://<server ipaddr>/action/SetGeneralParam` (Where `<server ipaddr>` is the device IP, for example:192.168.1.10)

3. Explanation of the Paragraph of the Requested Message

Key	Type	Values	Description
operator	string	SetGeneralParam	Set information about common configuration parameters.
DeviceID	int		ID number of the device to be operated See 6.8.1 Getting Device Information
info	json object		Concrete content
FaceThreshold	int (optional)	50 ~ 100	Black and White List comparison threshold
IDCardThreshold	int (optional)	50 ~ 100	ID card comparison threshold
ControlType	int (optional)	0~2	0:Switching; 1:Wiegand interface 26 bits; 2:Wiegand interface 34 bits
AutoRebootDay	int (optional)	0~8	Device AutoReboot Setting Date 0:No reboot; 1:Daily reboot; 2~8:Weekly AutoRebootDay-1 reboot
AutoRebootHour	int (optional)	0~23	Device Reboot Integral Time
Direction	int (optional)	0~2	Entrance/Exit Direction 0:Unidirectional 1:Entrance 2:Exit
IPAddr	string (optional)		Device Ip address (64 bytes)
Submask	string (optional)		Device subnet mask (32 bytes)
Gateway	string (optional)		Device gateway (32 bytes)
DNS	string (optional)		Device DNS (32 bytes)

4. Example of Request Message

```
URL of the request:http://172.168.2.10/action/SetGeneralParam
Content of the request:
{
```

```
    "operator": "SetGeneralParam",
    "DeviceID": 1305269,
    "info": {
        "FaceThreshold": 75,
        "IDCardThreshold": 50,
        "ControlType": 0,
        "Direction": 0,
        "AutoRebootDay": 0,
        "AutoRebootHour": 0,
        "IPAddr": "172.168.2.130",
        "Submask": "255.255.255.0",
        "Gateway": "172.168.2.1",
        "DNS": "202.96.134.33"
    }
}
```

5. Explanation of the Paragraph of the Reply Message

   Parameter information(Note: `Optional` is optional):

Key	Type	Values	Description
operator	string	SetGeneralParam	Set information about common configuration parameters.
info	JSON Object		Concrete content
code	int		Command execution error code 200-successes,see also 7.12 Setting Common Configuration Parameters Error Codes
Result	String	“Ok”/“Fail”	Operating result
Detail	String (optional)		Error message when Result is "Fail"

6. Example of Reply Message

```
{
    "operator": "SetGeneralParam",
    "code": 462,
    "info": {
        "Result": "Fail",
        "Detail": "Unknow DeviceID"
    }
}
```

6.11 System Parameters

The system parameters mainly include the system operation parameters of the face recognition machine, such as the device language, ID card reader type, the start and end time point of the vivo detection mode and so on.

6.11.1 Getting System Parameters

1. Description

   Get the system parameters of the device.

2. API Description

Items	Description
Operator	`GetSysCfg`
Request URL	`http://<server ipaddr>/action/GetSysCfg` (Where `<server ipaddr>` is the device IP, for example:192.168.1.10)

3. Example of Request Message

```
URL of the request:http://192.168.2.10/action/GetSysCfg
Content of the request:（NULL）
```

4. Explanation of the Paragraph of the Reply Message

   Parameter information(Note: `Optional` is optional)

Key	Type	Values	Description
operator		GetSysCfg	Get the system parameters of the device.
info			Concrete content
Language	int	0~9	Device Language: 0:English 1:Chinese Simplified 2:Chinese Traditional 3:Portuguese 4:Korean 5:Russian (supported by special versions) 6:Italian 7:French 8:Spanish 9:Japanese Changing the device language will reboot the machine.
DataBaseEnable	int	0~1	Whether to save record capture+record ID+record authentication 0:No 1:Yes Default:1, if not turned on, the capture, ID and authentication image data will not be saved.
IDcardType	int	Reserve	Reserve
CardReaderType	int	Reserve	Reserve
CardOpenDoorWay	int	Reserve	Swipe card door opening method 0:Face + card 1:Face or card
FaceDisplay	int	0~1，default 1	Whether to display face frame 0:No 1:Yes
LiveDetectType	int	0~2，default 1	Live detection mode 0: live detection always on 1: live detection off 2: live detection on by time period
LiveDetectTimeBeg	String	e.g.,07:00:00	Live detect daily start point Required when LiveDetectType=2
LiveDetectTimeEnd	String	e.g.,19:00:00	Live detect daily end point Required when LiveDetectType=2
LiveThreshold	String	e.g.,90.000000	Live detection threshold (not settable)

Key	Type	Values	Description
LiveFrameNum	int		The number of consecutive frames for live body detection, the larger the number of frames, the longer the recognition occupation time increases (not settable).
LedLightType	int	0~5	Enable White Light Type 0:Never 1:Time Control 2:Light Sensitive Control 3:Face Sense 4:Face or Time 5:Face or Light Sense
LedBrightness	int		Use when white light is enabled
LedTimeBeg	String	e.g.,19:00:00	White light activation start time per day (white light activation time), required when LedLightType=1,4
LedTimeEnd	String	e.g.,07:00:00	White light activation end time per day (white light off time), required when LedLightType=1, 4
LedDisableAfterSec	int		How many seconds after no one is there to turn off the white light (face sensing), required when LedLightType=3,4,5.
LcdBLDisable	int		Whether to turn off the screen when no one is present 0: Never 1:Turn off the screen display when no one is around
LcdBLDisableAfterSec	int		How many seconds to turn off the screen after no one is present, required when LcdBLDisable=1
ScreenBrightness	int	Default:50	Screen brightness (reserved)
WebTimeOut	int	2~10 Default:5	Web page login timeout (minutes)
SceneSnap	int	0~1	Whether to enable scene image capture
NightMode	int		Night mode on type 0: Off 1: On by Time 2: All Day On

Key	Type	Values	Description
NightTimeBeg	String	e.g.,19:00:00	Night start-up time point
NightTimeEnd	String	e.g.,07:00:00	End-of-night time point

5. Example of Reply Message

```
{
    "operator": "GetSysCfg",
    "info": {
        "Language": 1,
        "DataBaseEnable": 1,
        "IDcardType": 0,
        "CardReaderType": 0,
        "CardOpenDoorWay": 0,
        "FaceDisplay": 0,
        "LiveDetectType": 0,
        "LiveDetectTimeBeg": "07:00:00",
        "LiveDetectTimeEnd": "19:00:00",
        "LiveThreshold": "90.000000",
        "LiveFrameNum": 5,
        "LedLightType": 0,
        "LedTimeBeg": "19:00:00",
        "LedTimeEnd": "07:00:00",
        "LedBrightness": 50,
        "LedDisableAfterSec": 30,
        "LcdBLDisable": 0,
        "LcdBLDisableAfterSec": 30,
        "ScreenBrightness": 50,
        "WebTimeOut": 10,
        "SceneSnap": 0,
        "NightMode": 1,
        "NightTimeBeg": "18:00:00",
        "NightTimeEnd": "21:00:00"
    }
}
```

6.11.2 Setting System Parameters

1. Description

   Set the system parameters. the fields not included in the request message remain unchanged from the original configuration.

2. API Description

Items	Description
Operator	`SetSysCfg`
Request URL	`http://<server ipaddr>/action/SetSysCfg` (Where `<server ipaddr>` is the device IP, for example:192.168.1.10)

3. Explanation of the Paragraph of the Requested Message

   Parameter information(Note: `Optional` is optional):

Key	Type	Values	Description
operator	string	SetSysCfg	Set the system parameters
info			Concrete content
Language	int (optional)	0~9	Device Language: 0:English 1:Chinese Simplified 2:Chinese Traditional 3:Portuguese 4:Korean 5:Russian (supported by special versions) 6:Italian 7:French 8:Spanish 9:Japanese Changing the device language will reboot the machine.
DataBaseEnable	int (optional)	0~1	Whether to save record capture+record ID+record authentication 0:No 1:Yes Default:1, if not turned on, the capture, ID and authentication image data will not be saved.
IDcardType	int (optional)	Reserve	Reserve
CardReaderType	int (optional)	Reserve	Reserve
CardOpenDoorWay	int (optional)	Reserve	Swipe card door opening method 0:Face + card 1:Face or card
FaceDisplay	int (optional)	0~1,default 1	Whether to display face frame 0:No 1:Yes
LiveDetectType	int (optional)	0~2,default 1	Live detection mode 0: live detection always on 1: live detection off 2: live detection on by time period

Key	Type	Values	Description
LiveDetectTimeBeg	String **Required when LiveDetectType=2**	e.g.,07:00:00	Live detect daily start point
LiveDetectTimeEnd	String **Required when LiveDetectType=2**	e.g.,19:00:00	Live detect daily end point
LedLightType	int (optional)	0~5	Enable White Light Type 0:Never 1:Time Control 2:Light Sensitive Control 3:Face Sense 4:Face or Time 5:Face or Light Sense
LedBrightness	int (optional)		White light brightness
LedTimeBeg	String **required when LedLightType=1,4**	e.g.,19:00:00	White light activation start time per day (white light activation time)
LedTimeEnd	String **required when LedLightType=1, 4**	e.g.,07:00:00	White light activation end time per day (white light off time)
LedDisableAfterSec	int **required when LedLightType=3,4,5**		How many seconds after no one is there to turn off the white light (face sensing).
BodySensorEnable	int (optional)	0~1	Whether to turn on the body sensor
LcdBLDisable	int (optional)		Whether to turn off the screen when no one is present 0: Never 1: Turn off the screen display when no one is around
LcdBLDisableAfterSec	int **required when LcdBLDisable=1**	10~600	Disable screen when no pass after seconds
ScreenBrightness	int (optional)	Default:50	Screen brightness (reserved)

Key	Type	Values	Description
WebTimeOut	int (optional)	2~10 default:5	Web page login timeout (minutes)
SceneSnap	int (optional)	0~1	Whether to enable scene image capture
NightMode	int (optional)		Night mode on type 0: Off 1: On by Time 2: All Day On
NightTimeBeg	String (optional)	e.g.,19:00:00	Night start-up time point
NightTimeEnd	String (optional)	e.g.,07:00:00	End-of-night time point

4. Example of Request Message

```
URL of the request:http://192.168.2.10/action/SetSysCfg
Content of the request:
{
    "operator": "SetSysCfg",
    "info": {
        "LedLightType": 4,
        "LedTimeBeg": "19:00:00",
        "LedTimeEnd": "06:00:00",
        "ScreenBrightness": 90,
        "WebTimeOut": 10
    }
}
```

5. Explanation of the Paragraph of the Reply Message

   Parameter information(Note: `Optional` is optional):

Key	Type	Values	Description
operator	string	SetSysCfg	Setting System Parameters
info	JSON Object		Concrete content
code	int		Command execution error code 200-successes,see also7.13 Setting System Parameters Error Codes
Result	String	“Ok”/“Fail”	Operating result
Detail	String (optional)		Error message when Result is "Fail"

6. Example of Reply Message

```
{
    "operator": "SetSysCfg",
    "code": 200,
    "info": {
        "Result": "Ok"
    }
}
```

6.12 MQTT Parameters

MQTT parameters are mainly related to the MQTT service operation parameters of the facial recognition machine, setting the important parameters of this interface will make the facial recognition machine and MQTT server reconnect.

6.12.1 Getting MQTT Parameters

1. Description

   Get MQTT parameters.

2. API Description

Items	Description
Operator	`GetMQTTParam`
Request URL	`http://<server ipaddr>/action/GetMQTTParam` (Where `<server ipaddr>` is the device IP, for example:192.168.1.10)

3. Example of Request Message

```
URL of the request:http://192.168.2.10/action/GetMQTTParam
Content of the request:（NULL）
```

4. Explanation of the Paragraph of the Reply Message

   Parameter information(Note:`Optional`is optional)

Key	Type	Values	Description
operator		GetMQTTParam	Get MQTT parameters.
info			Concrete content
MQEnable	int	0~1	Enable/disable MQTT service 0:Disable 1:Enable
MQCloudID	String		MQTT gate cloud ID number Recommended native ID (e.g.: 1300001)
MQAddr	String		MQTT cloud address
MQPort	int		MQTT cloud port
MQUser	String		MQTT cloud username
MQPwd	String		MQTT cloud password
MQTopic	String		MQTT cloud Topic
StrangerUploadType	int	0~1	Whether to upload stranger capture information (default:0) **0:upload** **1:no upload**
RecordUploadType	int	0~2	Authentication Recognition Information Upload or Not (Default:1) 0:No upload 1:Recognition Record Upload with Snapshot 2:Recognition Record Upload without Snapshot Image
Direction	int	0~2	Entrance/Exit Direction 0:Unidirectional 1:Entrance 2:Exit
QRCode	int	0~1	Whether to upload QR code scanning result (default:0) 0:Not upload 1:Upload

Key	Type	Values	Description
ResumefromBreakpoint	int	0~1	MQTT stranger capture information and identification records whether to enable `Continue Transmitting After Disconnection` (under enable stranger capture upload or identification records upload mode) 0:Not enable 1:Enable Without `Continue Transmitting After Disconnection` function, it only guarantees to push the data, and it does not need to be returned by the platform (cloud server). Turn on the function, then push the stranger capture information and authentication information need to be returned by the server (see MQTT protocol 9.7 `Continue Transmitting After Disconnection` instructions) 10 seconds after the all-in-one machine does not receive the correct data packet from the server, then continue to push this information.
BeginTime	String	YYYY-MM-DDThh:mm:ss	The default start time of MQTT `Continue Transmitting After Disconnection`, you can leave it blank. Default is the time to set up the stranger capture upload or authentication record upload and enable the `Continue Transmitting After Disconnection` function. If the stranger capture and authentication record information before the breakpoint is pushed, this time will be changed to the time to finish the `Continue Transmitting After Disconnection`.e.g., 2020-05-25T09:10:00
Alarm	int	0~1	Whether alarms are uploaded (default:1) 0:not uploaded 1:uploaded

Key	Type	Values	Description
KeepAliveInterval	int	10~300(Sec)	Application layer heartbeat interval Interval
OnlineTopic	string		Topics for online notification
HeartbeatTopic	string		Heartbeat Reporting Topic
IDCard	int	0~1	Whether to upload ID information (default:1) 0:Not uploaded 1:Uploaded
Card	int	0~1	Whether IC or RF card number is uploaded (default:1) 0:Not uploaded 1:Uploaded

5. Example of Reply Message

```
{
    "operator": "GetMQTTParam",
    "info": {
        "MQEnable": 0,
        "MQAddr": "172.168.2.90",
        "MQPort": 61613,
        "MQUser": "admin",
        "MQPwd": "password",
        "MQTopic": "mqtt/face/1743725",
        "MQCloudID": "1743725",
        "StrangerUploadType": 0,
        "RecordUploadType": 1,
        "Direction": 2,
        "QRCode": 0,
        "IDCard": 1,
        "Card": 1,
        "Alarm": 1,
        "KeepAliveInterval": 60,
        "OnlineTopic": "mqtt/face/basic",
        "HeartbeatTopic": "mqtt/face/heartbeat",
        "ResumefromBreakpoint": 0,
        "BeginTime": "2022-12-14T11:24:07"
    }
}
```

6.12.2 Setting MQTT Parameters

1. Description

   Set MQTT parameters.

2. API Description

Items	Description
Operator	`SetMQTTParam`
Request URL	`http://<server ipaddr>/action/SetMQTTParam` (Where `<server ipaddr>` is the device IP, for example:192.168.1.10)

3. Explanation of the Paragraph of the Requested Message

   Parameter information(Note: `Optional` is optional)

Key	Type	Values	Description
operator		SetMQTTParam	Set MQTT parameters.
info			Concrete content
MQEnable	int (optional)	0~1	Enable/disable MQTT service 0:Disable 1:Enable
MQCloudID	String (optional)		MQTT gate cloud ID number Recommended native ID(e.g:1300001)
MQAddr	String (optional)		MQTT cloud address
MQPort	int (optional)		MQTT cloud port
MQUser	String (optional)		MQTT cloud username
MQPwd	String (optional)		MQTT cloud password
MQTopic	String (optional)		MQTT cloud Topic
StrangerUploadType	int (optional)	0~1	Whether to upload stranger capture information (default:0) **0:upload** **1:no upload**
RecordUploadType	int (optional)	0~2	Authentication Recognition Information Upload or Not (Default:1) 0:No upload 1:Recognition Record Upload with Snapshot 2:Recognition Record Upload without Snapshot Image
Direction	int (optional)	0~2	Entrance/Exit Direction 0:Unidirectional 1:Entrance 2:Exit

Key	Type	Values	Description
QRCode	int (optional)	0~1	Whether to upload QR code scanning result (default:0) 0:Not upload 1:Upload
ResumefromBreakpoint	int (optional)	0~1	MQTT stranger capture information and identification records whether to enable `Continue Transmitting After Disconnection` (under enable stranger capture upload or identification records upload mode) 0:Not enable 1:Enable Without `Continue Transmitting After Disconnection` function, it only guarantees to push the data, and it does not need to be returned by the platform (cloud server). Turn on the function, then push the stranger capture information and authentication information need to be returned by the server (see MQTT protocol 9.7 `Continue Transmitting After Disconnection` transmission instructions) 10 seconds after the all-in-one machine does not receive the correct data packet from the server, then continue to push this information.

Key	Type	Values	Description
BeginTime	String (optional)	YYYY-MM-DDThh:mm:ss	The default start time of MQTT `Continue Transmitting After Disconnection`, you can leave it blank. Default is the time to set up the stranger capture upload or authentication record upload and enable the `Continue Transmitting After Disconnection` function. If the stranger capture and authentication record information before the breakpoint is pushed, this time will be changed to the time to finish the `Continue Transmitting After Disconnection`.e.g., 2020-05-25T09:10:00
Alarm	int	0~1	Whether alarms are uploaded (default:1) 0:not uploaded 1:uploaded
KeepAliveInterval	int	10~300(Sec)	Bottom heartbeat interval Interval, defaults to 20 when out of range
OnlineTopic	string		Topics for online notification
HeartbeatTopic	string		Heartbeat Reporting Topic
IDCard	int	0~1	Whether to upload ID information (default:1) 0:Not uploaded 1:Uploaded
Card	int	0~1	Whether IC or RF card number is uploaded (default:1) 0:Not uploaded 1:Uploaded

4. Example of Request Message

```
URL of the request:http://192.168.2.10/action/SetMQTTParam
Content of the request:
{
    "operator": "SetMQTTParam",
    "info": {
        "StrangerUploadType": 0,
        "RecordUploadType": 1
    }
}
```

5. Explanation of the Paragraph of the Reply Message

   Parameter information(Note: `Optional` is optional):

Key	Type	Values	Description
operator	string	SetMQTTParam	Set MQTT parameters.
info	JSON Object		Concrete content
code	int		Command execution error code 200-successes,see also7.14 Set MQTT Parameters Error Codes
Result	String	“Ok”/“Fail”	Operating result
Detail	String (optional)		Error message when Result is "Fail"

6. Example of Reply Message

```
{
    "operator": "SetMQTTParam",
    "code": 200,
    "info": {
        "Result": "Ok"
    }
}
```

6.13 WiFi Parameters

WiFi parameter mainly involves the device amount WiFi parameter settings, need to be used with the default route in 6.4 Network Parameters, **support WiFi models of the device model and version only have this parameter.**

6.13.1 Getting WiFi Parameters

1. Description

   Get the WiFi parameters of the device.

2. API Description

Items	Description
Operator	`GetNetWiFi`
Request URL	`http://<server ipaddr>/action/GetNetWiFi` (Where `<server ipaddr>` is the device IP, for example:192.168.1.10)

3. Example of Request Message

```
URL of the request:http://192.168.2.10/action/GetNetWiFi
Content of the request:（NULL）
```

4. Explanation of the Paragraph of the Reply Message

   Parameter information(Note: `Optional` is optional):

Key	Type	Values	Description
operator	string	GetNetWiFi	Get the WiFi parameters .
info	json object		Concrete content
Enable	int		Whether WiFi is enabled or not 0:Not enabled 1:Enabled
WorkMode	int	Default:0	Operating mode 0:Sta mode
SecurityMode	int	0~2	Secure mode 0: no encryption 1:WPA-PSK/WPA2-PSK 2:WPA/WPA2
SSID	String		SSID(up to 32 bytes)
PassWord	String		Password (up to 128 bytes)
DHCP	int	0~1	DHCP enabled or not 0:not enabled 1:enabled
IPAddr	string		Device ip address (up to 16 bytes), required when DHCP=0
NetMask	string		Device ip subnet mask (up to 16 bytes), required when DHCP=0
GateWay	string		Device ip gateway (up to 16 bytes), required when DHCP=0
DNS1	string		DNS1
DNS2	string		DNS2

5. Example of Reply Message

```
{
    "operator": "GetNetWiFi",
    "info": {
        "Enable": 1,
        "WorkMode": 0,
        "DHCP": 1,
        "SecurityMode": 1,
        "SSID": "TP-LINK_4755",
        "PassWord": "testpassword",
```

```
        "IPAddr": "192.168.1.104",
        "NetMask": "255.255.255.0",
        "GateWay": "0.0.0.0",
        "DNS1": "202.96.134.33",
        "DNS2": "202.96.134.33"
    }
}
```

6.13.2 Setting WiFi Parameters

1. Description

   Set the WiFi parameters of the device.

2. API Description

Items	Description
Operator	`SetNetWiFi`
Request URL	`http://<server ipaddr>/action/SetNetWiFi` (Where `<server ipaddr>` is the device IP, for example:192.168.1.10)

3. Explanation of the Paragraph of the Requested Message

   Parameter information(Note: `Optional` is optional):

Key	Type	Values	Description
operator	string	SetNetWiFi	Set the WiFi parameters of the device.
info	json object		Concrete content
Enable	int (optional)		Whether WiFi is enabled or not 0:Not enabled 1:Enabled
WorkMode	int (optional)	Default:0	Operating mode 0:Sta mode
SecurityMode	int (optional)	0~2	Secure mode 0: no encryption 1:WPA-PSK/WPA2-PSK 2:WPA/WPA2
SSID	String (optional)		SSID(up to 32 bytes)
PassWord	String (optional)		Password (up to 128 bytes)
DHCP	int (optional)	0~1	DHCP enabled or not 0:not enabled 1:enabled
IPAddr	string (optional)		Device ip address (up to 16 bytes), required when DHCP=0
NetMask	string (optional)		Device ip subnet mask (up to 16 bytes), required when DHCP=0
GateWay	string (optional)		Device ip gateway (up to 16 bytes), required when DHCP=0
DNS1	string (optional)		DNS1
DNS2	string (optional)		DNS2

4. Example of Request Message

```
URL of the request:http://192.168.2.10/action/SetNetWiFi
Content of the request:
{
    "operator": "SetNetWiFi",
    "info": {
```

```
        "Enable": 1,
        "WorkMode": 0,
        "DHCP": 1,
        "SecurityMode": 1,
        "SSID": "TP-LINK_4755",
        "PassWord": "testpassword",
        "IPAddr": "192.168.1.119",
        "NetMask": "255.255.255.0",
        "GateWay": "192.168.1.0",
        "DNS1": "202.96.134.33",
        "DNS2": "202.96.134.33"
    }
}
```

5. Explanation of the Paragraph of the Reply Message

   Parameter information(Note: `Optional` is optional):

Key	Type	Values	Description
operator	string	SetNetWiFi	Set the WiFi parameters of the device.
info	JSON Object		Concrete content
code	int		Command execution error code 200-successes,see also 7.15 Setting WiFi Parameters Error Codes
Result	String	“Ok”/“Fail”	Operating result
Detail	String (optional)		Error message when Result is "Fail"

6. Example of Reply Message

```
{
    "operator": "SetNetWiFi",
    "code": 200,
    "info": {
        "Result": "Ok"
    }
}
```

6.14 RTSP Parameters

RTSP parameters are mainly the RTSP operation parameters of the device, set the RTSP parameters successfully **the device will reboot**, currently this interface is only supported in some models.

6.14.1 Getting RTSP Parameters

1. Description

   Get the RTSP parameters of the device.

2. API Description

Items	Description
Operator	`GetRTSPCfg`
Request URL	`http://<server ipaddr>/action/GetRTSPCfg` (Where `<server ipaddr>` is the device IP, for example:192.168.1.10)

3. Example of Request Message

```
URL of the request:http://192.168.2.10/action/GetRTSPCfg
Content of the request:（NULL）
```

4. Explanation of the Paragraph of the Reply Message

   Parameter information(Note: `Optional` is optional):

Key	Type	Values	Description
operator	string	GetRTSPCfg	Get the RTSP parameters of the device
info	json object		Concrete content
OpenVerify	Int	0~1	Whether to enable RTSP service 0:Not enabled 1:Enabled
PackSize	Int	1~1500 （Default:1500）	Packing size byte
RTSPPort	Int		RTSP Port

5. Example of Reply Message

```
{
    "operator": "GetRTSPCfg",
    "info": {
        "OpenVerify": 0,
        "PackSize": 1500,
        "RTSPPort": 554
    }
}
```

6.14.2 Setting RTSP Parameters

1. Description

   Set the RTSP parameters of the device.

2. API Description

Items	Description
Operator	`SetRTSPCfg`
Request URL	`http://<server ipaddr>/action/SetRTSPCfg` (Where `<server ipaddr>` is the device IP, for example:192.168.1.10)

3. Explanation of the Paragraph of the Requested Message

   Parameter information(Note: `Optional` is optional):

Key	Type	Values	Description
operator	string	SetRTSPCfg	Set the RTSP parameters of the device.
DeviceID	int		ID number of the device to be operated See 6.8.1 Getting Device Information
info	json object		Concrete content
OpenVerify	Int (optional)	0~1	Whether to enable RTSP service 0:Not enabled 1:Enabled
PackSize	Int (optional)	1~1500 （Default:1500）	Packing size byte
RTSPPort	Int (optional)		RTSP port

4. Example of Request Message

```
URL of the request:http://192.168.2.10/action/SetRTSPCfg
Content of the request:
{
    "operator": "SetRTSPCfg",
    "DeviceID":1743725,
    "info": {
        "OpenVerify": 1,
        "PackSize": 1600,
        "RTSPPort": 500
    }
}
```

5. Explanation of the Paragraph of the Reply Message

   Parameter information(Note:`Optional` is optional):

Key	Type	Values	Description
operator	string	SetRTSPCfg	Set the RTSP parameters of the device.
info	JSON Object		Concrete content
code	int		Command execution error code 200-successes,see also7.16 Setting RTSP Parameters Error Codes
Result	String	“Ok”/“Fail”	Operating result
Detail	String (optional)		Error message when Result is "Fail"

6. Example of Reply Message

```
{
    "operator": "SetRTSPCfg",
    "code": 465,
    "info": {
        "Result": "Fail",
        "Detail": "Device's model can not support this interface"
    }
}
```

6.15 Temperature Parameters

The temperature parameters is mainly related to the temperature operation parameter of the face recognition all-in-one machine,**only the models that support temperature detection support these parameter.**

6.15.1 Getting Temperature Parameters

1. Description

   Get the temperature parameters.

2. API Description

Items	Description
Operator	`GetTemperature`
Request URL	`http://<server ipaddr>/action/GetTemperature` (Where `<server ipaddr>` is the device IP, for example:192.168.1.10)

3. Example of Request Message

```
URL of the request:http://192.168.2.10/action/GetTemperature
Content of the request:（NULL）
```

4. Explanation of the Paragraph of the Reply Message

   Parameter information(Note: `Optional` is optional):

Key	Type	Values	Description
operator	string	GetTemperature	Get temperature parameters.
info	json object		Concrete content
TemperatureMode	int	0~1	Temperature Display Mode 0:Celsius 1:Fahrenheit
ShowAbnormalTemp	int	0~1	Display abnormal temperature 0:Not display 1:Display
TemperatureCheck	double	Default :0.00	Temperature calibration
TempAutoLow	int	0~1(Default 1)	Automatically adjusts the low temperature threshold (Default on will reduce parameter setting problems due to ambient temperature changes) 0:No 1:Yes
TemperatureLow	double	34.00(Default for C models); 28.00(Default except for C models)	Low Temperature Threshold (Detected temperatures below the Low Temperature Threshold will be filtered, when TempAutoLow=1, the Low Temperature Threshold will be adaptive according to the outside ambient temperature)
TemperatureHigh	double	Default :37.30	High Temperature Threshold (detected temperatures above the high temperature threshold will be considered abnormal for body temperature)
EnvTemperature	double		Ambient temperature (reserved)
EnvTemperatureCheck	double		Ambient temperature calibration value (reserved)
TempFaceMinW	int	100~500	Minimum face recognition pixel width for temperature measurement The smaller the value, the farther the recognition distance for temperature measurement (default 350).

Key	Type	Values	Description
TempFaceMinH	int	50~1000	Minimum face recognition pixel height, limited to forehead. The smaller the value the smaller the recognition area, the larger the value the larger the recognition area (default 560).
bTempTimeHigh	int	0~1	Whether to limit the time period for high temperature detection 0: No time period limitation 1: Time period limitation
bTempokSwitch	int	0~1	Whether to turn on high-temperature alert 0: No 1: On
nFaceLeftUpX	int		Face position X coordinate
nFaceLeftUpY	int		Face position Y coordinate
BDHC	double	40.00	BDHC:(Supported by special model)
BDLC	double	28.00	BDLC:(Supported by special model)
nTempWay	int	0~1	Temperature measurement mode: (Supported by special model) 0:Precision mode 1:Fast mode
nTempTimeBeg	int		High Temperature Detection Start Point Seconds counted from 0:00 of the day
nTempTimeEnd	int		High temperature detection end point Seconds counted from 0:00 of the day

5. Example of Reply Message


```
{
    "operator": "GetTemperature",
    "info": {
        "TemperatureMode": 0,
        "ShowAbnormalTemp": 0,
        "TemperatureCheck": 0.00,
        "TemperatureLow": 28.00,
        "TemperatureHigh": 37.30,
        "EnvTemperature": 0.00,
        "EnvTemperatureCheck": 0.00,
```

```
        "OpenLaser": 0,
        "TempFaceMinW": 350,
        "TempFaceMinH": 560
    }
}
```

6.15.2 Setting Temperature Parameters

1. Description

   Set the temperature parameters. Fields not included in the request message follow the original configuration by default.

2. API Description

Items	Description
Operator	`SetTemperature`
Request URL	`http://<server ipaddr>/action/SetTemperature` (Where `<server ipaddr>` is the device IP, for example:192.168.1.10)

3. Explanation of the Paragraph of the Requested Message

   Parameter information(Note: `Optional` is optional):

Key	Type	Values	Description
operator	string	SetTemperature	Set the temperature parameters.
info	json object		Concrete content
DeviceID	int		ID number of the device to be operated See 6.8.1 Getting Device Information
TemperatureMode	int (optional)	0~1	Temperature display mode 0:Celsius 1:Fahrenheit
ShowAbnormalTemp	int (optional)	0~1	Display abnormal temperature 0:Not display 1:Display
TemperatureCheck	double (optional)	0.0	Temperature calibration
TemperatureLow	double (optional)	34.00(Default for C models); 28.00(Default except for C models)	Low Temperature Threshold
TemperatureHigh	double (optional)	37.3	High Temperature Threshold ,Default :37.30
EnvTemperature	double (optional)		Ambient temperature (reserved)
EnvTemperatureCheck	double (optional)		Ambient temperature calibration value (reserved)
TempFaceMinW	int (optional)	100~500	Minimum face recognition pixel width for temperature measurement The smaller the value, the farther the recognition distance for temperature measurement (default 350).
TempFaceMinH	int (optional)	50~1000	Minimum face recognition pixel height, limited to forehead. The smaller the value the smaller the recognition area, the larger the value the larger the recognition area (default 560).

Key	Type	Values	Description
TempAutoLow	int (optional)	0~1(default 1)	Whether to limit the time period for high temperature detection 0: No time period limitation 1: Time period limitation
bTempTimeHigh	int (optional)	0~1	Whether to limit the time period for high temperature detection 0: No time period limitation 1: Time period limitation
bTempokSwitch	int (optional)	0~1	Whether to turn on high-temperature alert 0: No 1: On
nFaceLeftUpX	int (optional)		Face position X coordinate
nFaceLeftUpY	int (optional)		Face position Y coordinate
BDHC	double (optional)	40.00	BDHC:(Supported by special model)
BDLC	double (optional)	28.00	BDLC:(Supported by special model)
nTempWay	int (optional)	0~1	Temperature measurement mode: (Supported by special model) 0:Precision mode 1:Fast mode
nTempTimeBeg	int (optional)		High temperature detection start point Seconds counted from 0:00 of the day
nTempTimeEnd	int (optional)		High temperature detection end point Seconds counted from 0:00 of the day

4. Example of Request Message

```
{
    "operator": "SetTemperature",
    "info": {
        "DeviceID": 1305433,
        "TemperatureMode": 0,
        "ShowAbnormalTemp": 0,
        "TemperatureCheck": 0.00,
        "TemperatureLow": 28.00,
```

```
        "TemperatureHigh": 37.30,
        "EnvTemperature": 0.00,
        "EnvTemperatureCheck": 0.00,
        "OpenLaser": 0,
        "TempFaceMinW": 350,
        "TempFaceMinH": 560
    }
}
```

5. Explanation of the Paragraph of the Reply Message

   Parameter information(Note: `Optional` is optional):

Key	Type	Values	Description
operator	string	SetTemperature	Set the temperature parameters.
info	JSON Object		Concrete content
code	int		Command execution error code 200-successes,see also7.17 设置温度参数错误码
Result	String	“Ok”/“Fail”	Operating result
Detail	String (optional)		Error message when Result is "Fail"

6. Example of Reply Message

```
{
    "operator": "SetTemperature",
    "code": 200,
    "info": {
        "Result": "Ok"
    }
}
```

6.16 Handshake Protocol Data

Handshake protocol data is mainly related to the customer docking platform can write some data content to the face recognition all-in-one machine to save, generally is the third party's encrypted data content or third-party authentication data, etc., the size of the data content **not more than 4K bytes**.

6.16.1 Getting Handshake Protocol Data

1. Description

   Get handshake protocol data.

2. API Description

Items	Description
Operator	`GetHandSharkData`
Request URL	`http://<server ipaddr>/action/GetHandSharkData` (Where `<server ipaddr>` is the device IP, for example:192.168.1.10)

3. Example of Request Message

```
URL of the request:http://192.168.2.10/action/GetHandSharkData
Content of the request:（空）
```

4. Explanation of the Paragraph of the Reply Message

   Parameter information(Note: `Optional` is optional):

Key	Type	Values	Description
operator	string	GetHandSharkData	Get handshake protocol data.
info	json object		Concrete content
HandSharkInfo	string		Data content, not to exceed 4K bytes.

5. Example of Reply Message

```
{
    "operator": "GetHandSharkData",
    "info": {
        "HandSharkInfo": "new HandSharkInfo!"
    }
}
```

6.16.2 Setting Handshake Protocol Data

1. Description

   Set handshake protocol data.

2. API Description

Items	Description
Operator	`SetHandSharkData`
Request URL	`http://<server ipaddr>/action/SetHandSharkData` (Where `<server ipaddr>` is the device IP, for example:192.168.1.10)

3. Explanation of the Paragraph of the Requested Message

   Parameter information(Note: `Optional` is optional):

Key	Type	Values	Description
operator	string	SetHandSharkData	Set handshake protocol data.
info	json object		Concrete content
DeviceID	int		ID number of the device to be operated See 6.8.1 Getting Device Information
HandSharkInfo	string		Data content, not to exceed 4K bytes.

4. Example of Request Message

```
URL of the request:http://192.168.2.10/action/SetHandSharkData
Content of the request:
{
    "operator":"SetHandSharkData",
    "info":{
        "DeviceID":1743726,
        "HandSharkInfo":"this is new HandSharkInfo!"
    }
}
```

5. Explanation of the Paragraph of the Reply Message

   Parameter information(Note: `Optional` is optional):

Key	Type	Values	Description
operator	string	HttpSetHandSharkData	Set handshake protocol data.
info	JSON Object		Concrete content
code	int		Command execution error code 200-successes,see also7.42 Setting Handshake Protocol Data Error Codes
Result	String	“Ok”/“Fail”	Operating result
Detail	String (optional)		Error message when Result is "Fail"

6. Example of Reply Message

```
{
    "operator": "HttpSetHandSharkData",
    "code": 200,
    "info": {
        "Result": "Ok"
    }
}
```

6.17 GPS Location Information

Only models with GPS modules support interfaces related to GPS location information.

6.17.1 Getting GPS Location Information

1. Description

   Get the current GPS location information of the device.

2. API Description

Items	Description
Operator	`GetGpsInfo`
Request URL	`http://<server ipaddr>/action/GetGpsInfo` (Where `<server ipaddr>` is the device IP, for example:192.168.1.10)

3. Example of Request Message

```
URL of the request:http://172.168.5.227/action/GetGpsInfo
Content of the request:（NULL）
```

4. Explanation of the Paragraph of the Reply Message

   Parameter information(Note: `Optional` is optional):

Key	Type	Values	Description
operator	string	GetGpsInfo	Get the current GPS location information of the device.
info	json object		Concrete content
GpsState	string		GPS module status.: "NotConfigGPSType": The current version of the device does not have a GPS module set "Initiating": GPS module is initializing. "OK": GPS module working properly "NotConncet": GPS module not connected "BadSignal": No signal from GPS module
GpsNum	int		Number of satellites captured
CoordinateSystem	int		Coordinate system 0: WGS84 1: GCJ02 Default 0
Longitude	double(Retain 5 decimal places)		Longitude
Latitude	double(Retain 5 decimal places)		Latitude
EorW	string	"E"/"W"	East/West
NorS	string	"N"/"S"	North/South
Altitude	double(Retain 1 decimal places)		Height above sea level
UtcDate	string	YYYY-MM-DD	UTC Date
UtcTime	string	hh:mm::ss	UTC Time

5. Example of Reply Message

```
{
    "operator": "GetGpsInfo",
    "info": {
        "GpsState": "NotConnect"
    }
}
```

6.17.2 Setting GPS Parameters

1. Description

   Set the current GPS location information of the device.

2. API Description

Items	Description
Operator	`SetGpsInfo`
Request URL	`http://<server ipaddr>/action/SetGpsInfo` (Where `<server ipaddr>` is the device IP, for example:192.168.1.10)

3. Explanation of the Paragraph of the Requested Message

   Parameter information(Note: `Optional` is optional):

Key	Type	Values	Description
operator	string	SetGpsInfo	Set the current GPS location information of the device.
info	json object		Concrete content
CoordinateSystemType	int (optional)	0~1	Coordinate system 0: WGS84 1: GCJ02

4. Example of Request Message

```
URL of the request:http://192.168.2.10/action/SetGpsInfo
Content of the request:
{
    "operator":"SetGpsInfo",
    "info":{
        "CoordinateSystemType":0
    }
}
```

5. Explanation of the Paragraph of the Reply Message

Parameter information(Note: `Optional` is optional):

Key	Type	Values	Description
operator	string	SetGpsInfo	Setting GPS parameters.
info	JSON Object		Concrete content
code	int		Command execution error code 200-successes,see also7.44 Setting GPS location Information Error Codes
Result	String	"Ok"/"Fail"	Operating result
Detail	String (optional)		Error message when Result is "Fail"

6. Example of Reply Message

```
{
    "operator": "SetGpsInfo"
    "code": 200,
    "info": {
        "Result": "Ok"
    }
}
```

7、Error Code Cross-Reference Table

7.1 Setting of Door Opening Conditions and Output Control Parameters Error Codes

code	Type	String	Description
461	int	"Unknow operator"	"operator" error
462	int	"Parameter error"	Failed to get keyword “info”.
463	int	"VerifyMode error"	The "VerifyMode" parameter is set incorrectly.

7.2 Setting the Prompt Sound and Interface Display Parameters Error Codes

code	Type	String	Description
461	int	"Unknow operator"	"operator" error
462	int	"Parameter error"	Failed to get keyword “info”.

7.3 Playing Audio Files Error Codes

code	Type	String	Description
461	int	"Unknow operator"	"operator" error
462	int	"Parameter error"	Failed to get keyword “info”.
463	int	"Unknown DeviceID"	The "DeviceID" does not match the device ID.
464	int	"Remote audio control is not enabled"	Remote control playback sound not enabled
465	int	"AudioType error"	The "AudioType" parameter is set incorrectly.
466	int	"Unknown File"	Can't find the audio file to play

7.4 Set Username and Password Error Codes

code	Type	String	Description
461	int	"Unknow operator"	"operator" error
462	int	"Parameter error"	Failed to get keyword “info”.
463	int	"Unknown User"	The "User" field is empty.
464	int	"Unknown Pwd"	The "Pwd" field is empty.

7.5 Setting Network Parameters Error Codes

code	Type	String	Description
461	int	"Unknow operator"	"operator" error
462	int	"Parameter error"	Failed to get keyword “info”.
463	int	"IP addr error"	Invalid IP address in the “IPAddr” field.
464	int	"DHCP enabled, cannot set IP"	The device has DHCP enabled and cannot set IP address

7.6 Setting the Time Parameters Error Codes

code	Type	String	Description
461	int	"Unknow operator"	"operator" error
462	int	"Parameter error"	Failed to get keyword “info”.
463	int	"Invalid Time"	Invalid Time

7.7 Setting NTP Parameters Error Codes

code	Type	String	Description
461	int	"Unknow operator"	"operator" error
462	int	"Parameter error"	Failed to get keyword “info”.

7.8 Setting the Central Connection Parameters Error Codes

code	Type	String	Description
461	int	"Unknow operator"	"operator" error
462	int	"Parameter error"	Failed to get keyword “info”.

7.9 Restore Factory Settings Error Codes

code	Type	String	Description
461	int	"Unknow operator"	"operator" error
462	int	"Parameter error"	Failed to get keyword “info”.

7.10 Setting Device Information Error Codes

code	Type	String	Description
461	int	"Unknow operator"	"operator" error
462	int	"Parameter error"	Failed to get keyword “info”.

7.11 Setting Face Recognition Parameters Error Codes

code	Type	String	Description
461	int	"Unknow operator"	"operator" error
462	int	"Parameter error"	Failed to get keyword “info”.

7.12 Setting Common Configuration Parameters Error Codes

code	Type	String	Description
461	int	"Unknow operator"	"operator" error
462	int	"Unknow DeviceID"	The "DeviceID" does not match the device ID.
463	int	"Unknow info"	Failed to get keyword “info”.
464	int	"IP addr error"	Invalid IP address in the “IPAddr” field.

7.13 Setting System Parameters Error Codes

code	Type	String	Description
461	int	"Unknow operator"	"operator" error
462	int	"Parameter error"	Failed to get keyword “info”.
463	int	"Russian language is not supported in this version"	Russian is not supported in the current version
464	int	"HTTP change LiveDetectType is not authorized"	No permission to modify LiveDetectType via HTTP.

7.14 Set MQTT Parameters Error Codes

code	Type	String	Description
461	int	"Unknow operator"	"operator" error
462	int	"Parameter error"	Failed to get keyword “info”.
463	int	"ResumefromBreakpoint need Topic Snap or Verify"	To enable `Continue Transmitting After Disconnection` mode, you need to upload at least one of “Strangers” and “Control Logs”.

7.15 Setting WiFi Parameters Error Codes

code	Type	String	Description
461	int	"Unknow operator"	"operator" error
462	int	"Parameter error"	Failed to get keyword “info”.
463	int	"IP addr error"	Invalid IP address in the “IPAddr” field.

7.16 Setting RTSP Parameters Error Codes

code	Type	String	Description
461	int	"Unknow operator"	"operator" error
462	int	"Parameter error"	Failed to get keyword “info”.
463	int	"Unknow DeviceID"	The "DeviceID" does not match the device ID.
464	int	"RTSPPort error"	RTSP port does not match webserver port
465	int	"Device's model can not support this interface"	This interface is not supported by non-Hisi models

7.17 Setting Temperature Parameters Error Codes

code	Type	String	Description
461	int	"Unknow operator"	"operator" error
462	int	"Parameter error"	Failed to get keyword “info”.
463	int	"Unknow DeviceID"	The "DeviceID" does not match the device ID.

7.18 Add or Modify Personnel in Bulk Error Codes

code	Type	String	Description
411	int	"Unknow operator"	"operator" error
412	int	"Parameter error"	Failed to get keyword “info”.
413	int	"Unknow DeviceID"	The "DeviceID" does not match the device ID.
414	int	"UnKnown Total"	Failed to get the keyword “Total”.
415	int	"Total out of range"	Total exceeds the scope limit.
416	int	"json of data and Total is not equal"	Get the value of the json array whose name is not equal to Total.
417	int	"UnKnown IdType"	Failed to get keyword “IdType”.
418	int	"IdType error"	Incorrect entry of "IdType" value.
461	int		Failed to get keyword "CustomizeID".
462	int		Failed to get keyword "PersonUUID".
464	int		Failed to get image address IP
465	int		Get URI image timeout or download image failed (note if the device's DNS is correct)
466	int		Failed to get URI image data content (data less than 1000 bytes)
467	int		Image data is too large (must not exceed 1M)
468	int		Failed to extract facial features from images
469	int		Failed to write image data to database
470	int		Database insertion list failure
471	int		Database query for location of deleted items fails
472	int		Database image data rewrite failed
474	int		The list of database personnel is full
475	int		Wiegand card number “MjCardNo” already exists.
476	int		Failed to get keyword "WGFacilityCode".
477	int		Failed to get keyword "MjCardNo".
479	int		Failure to modify personnel list information in the database
480	int		"RFIDCard" card number already exists.

code	Type	String	Description
481	int		Failed to get keyword "picURI".
482	int		Failed to read image of existing people
483	int		The person already exists

7.19 Add or Modify for Individual Lists Error Codes

code	Type	String	Description
461	int	"Unknow operator"	"operator" error
462	int	"Parameter error"	Failed to get keyword “info”.
463	int	"Unknow DeviceID"	The "DeviceID" does not match the device ID.
464	int	"Unkonw CustomizeID"	Failed to get keyword "customizeID".
465	int	"Unkonw LibID"	Failed to get keyword "LibID".
466	int	"Unkonw UUID"	Failed to get keyword "UUID".
467	int	"Unkonw IdType"	Failed to get keyword "IdType".
468	int	"Unkonw Picinfo or Picinfo Base64 decode error"	“picinfo” image base64 decoding failed
469	int	"Other"	Database-related operations fail
470	int	"Unkonw ChannelAuthority"	Unknown access rights (not in use at this time)
471	int	"database version does not support the temporary list 2"	Database version does not support temporary list 2
472	int	"database version does not support the temporary list 3"	Database version does not support temporary list 3
473	int	"IC_CardNo already exist"	RF(ID) card number already exists
474	int	"WG_CardNo already exist"	Wiegand card number already exists.
475	int	"Person already exist"	The person already exists.
477	int	"Face Undetected"	No face detected in the image
478	int	"Get URI server ip error"	Error getting server IP for “picURI”.
479	int	"Get picinfo and connect URI IP error"	Failed to connect to “picURI”.
480	int	"Get URI pic data too short"	Image data retrieved based on “picURI” is too short.
481	int	"get pic and get URI error"	Can't find “picinfo” or “picURI” field when adding a list of people.
482	int	"InsertPerson2DBWithUUID err"	Personnel insertion database error
484	int	"HTTP change list is not authorized"	No permission to change list via HTTP

7.20 Single or Multiple List Deletion Error Codes

code	Type	String	Description
461	int	"Unknow operator"	"operator" error
462	int	"Parameter error"	Failed to get keyword “info”.
463	int	"Unknow DeviceID"	The "DeviceID" does not match the device ID.
464	int	"Unkonw CustomizeID"	Failed to get keyword "customizeID".
465	int	"Unkonw LibID"	Failed to get keyword "LibID".
466	int	"Unkonw UUID"	Failed to get keyword "UUID".
467	int	"Unkonw TotalNum"	Failed to get keyword "TotalNum".
484	int	"HTTP change list is not authorized"	No permission to delete list via HTTP.

7.21 All Lists Delete Error Codes

code	Type	String	Description
461	int	"Unknow operator"	"operator" error
462	int	"Parameter error"	Failed to get keyword “info”.
484	int	"HTTP change list is not authorized"	No permission to delete list via HTTP

7.22 Software Version Upgrade Error Codes

code	Type	String	Description
461	int	"Parameter error"	"operator" error or failed to get keyword “info”.
462	int	"Can't find Name"	Failed to get keyword "Name".
463	int	"Upgrade file download fail"	Failed to download upgrade file
464	int	"Can't find Path"	Failed to get the keyword "Path" or could not find the upgrade file path.
465	int	"opendir Fail"	Failed to open directory
466	int	"Can't find upgrade file"	Upgrade file not found
467	int	"Can't find facegate_upgrade.sh"	Can't find "facegate_upgrade.sh" script file
468	int	"Unkonw DeviceID"	The "DeviceID" does not match the device ID.
469	int	"check upgrade fileInfo error"	Failed to verify upgrade file information.
470	int	"check Platform or Lcdtype error"	Error in calibrating platform parameters or LCD screen parameters.
471	int	"get FSVersion error"	Error getting "FSVersion"
472	int	"check ALG Version error"	Error in the version number of the calibration algorithm
473	int	"extract upgrade files error"	Failed to unzip upgrade zip file
474	int	"executable program not find"	Executable program not found in upgrade file
475	int	"Upgrade is not supported for this type's file"	Upgrade type not supported at this time
476	int	"Upgrade wav file failed,please check the fileName"	Failed to upgrade voice file, please check the file name.
477	int	"UpgradeType is inconsistent with the download file"	Inconsistency between upgrade type and file type.
478	int	"The number of customized voice files exceeds 41"	The number of customized voice files exceeds 41.
479	int	"The number of ShowInfo Pic files exceeds 10"	The number of remote door opening UI image files exceeds 10.
480	int	"Please chaeck this pic!"	Please check the image file.
482	int	"This version CustomID does not match"	Client code mismatch.

code	Type	String	Description
483	int	"This version is old"	Algorithm version too low

7.23 Setting HTTP Subscription Parameters Error Codes

code	Type	String	Description
461	int	"Unknow operator"	"operator" error
462	int	"Parameter error"	Failed to get keyword “info”.
463	int	"Unknow DeviceID"	The "DeviceID" does not match the device ID.
464	int	"Unkonw Topic"	Unkonw topic
465	int	"Wait server start....."	Waiting for the server to start up
466	int	"ResumefromBreakpoint need Topic Snap or Verify"	To enable the `Continue Transmitting After Disconnection` feature, you need to subscribe to the capture or authentication log.

7.24 Unsubscribe Error Codes

code	Type	String	Description
461	int	"Unknow operator"	"operator" error
462	int	"Parameter error"	Failed to get keyword “info”.
463	int	"Unknow DeviceID"	The "DeviceID" does not match the device ID.
464	int	"Unkonw Topic"	Unkonw topic

7.25 Manual Push Control Logging Error Codes

code	Type	String	Description
462	int	"Unknow DeviceID"	The "DeviceID" does not match the device ID.
463	int	"Unknow TimeS"	Failed to get keyword "TimeS".
464	int	"Unknow TimeE"	Failed to get keyword "TimeE".
465	int	"HTTP SubscribeInfo error"	HTTP subscription configuration information error (subscription IP, port address format is not correct; did not enable the authentication record subscription function)
466	int	"Can't Find any record issue, please check TimeS and TimeE"	Control record not found, check start time and end time

7.26 Device Reboot Error Codes

code	Type	String	Description
461	int	"Unknow operator"	"operator" error
462	int	"Unknow Parameter"	Failed to get keyword “info”.
463	int	"Unkonw DeviceID"	The "DeviceID" does not match the device ID.
464	int	"Parameter error"	Parameter error ("IsRebootDevice" field is missing, or value is not 0 or 1)

7.27 Individual List Search Error Codes

code	Type	String	Description
461	int	"Unknow operator"	"operator" error
462	int	"Parameter error"	Failed to get keyword “info”.
463	int	"Unknow DeviceID"	The "DeviceID" does not match the device ID.
464	int	"Unkonw SearchType"	Unknown value of "SearchType" (the value of search type is not 0~2)
465	int	"Unkonw picinfo"	Failed to get keyword "Picture".
466	int	"Other"	Reserve
467	int	"Unkonw SearchID"	The person corresponding to "SearchID" could not be found.

7.28 Total Number of List Queries Error Codes

code	Type	String	Description
461	int	"Unknow operator"	"operator" error
462	int	"Parameter error"	Failed to get keyword “info”.
463	int	"Unknow DeviceID"	The "DeviceID" does not match the device ID.

7.29 Multiple List Searches Error Codes

code	Type	String	Description
461	int	"Unknow operator"	"operator" error
462	int	"Parameter error"	Failed to get keyword “info”.
463	int	"Unknow DeviceID"	The "DeviceID" does not match the device ID.
464	int	"Unkonw BeginNO"	Failed to get keyword "BeginNo".
465	int	"Unkonw RequestCount"	Failed to get keyword "RequestCount".
466	int	"Unkonw Picture"	Failed to get keyword "Picture".
467	int	"Other"	Reserve
468	int	"can't find person"	No corresponding person information was searched

7.30 The Total Number of Control Log Queried Error Codes

code	Type	String	Description
461	int	"Unknow operator"	"operator" error
462	int	"Parameter error"	Failed to get keyword “info”.
463	int	"Unknow DeviceID"	The "DeviceID" does not match the device ID.
464	int	"Unkonw ID"	Unknown "ID" (there is no `CustomizeID`/`PersonUUID`/`LibID` counterpart in the device list, or the value of "IdType" is not 0~2)
465	int	"Unkonw IdType"	Failed to get keyword "IdType".
466	int	"Unkonw BeginTime or EndTime"	Failed to get keyword "BeginTime" or "EndTime".
467	int	"Request Number too large"	Excessive number of requests.
468	int	"No Control Record"	No control logs
469	int	"Other"	Other

7.31 Control Log Personnel Information Queries Error Codes

code	Type	String	Description
461	int	"Unknow operator"	"operator" error
462	int	"Parameter error"	Failed to get keyword “info”.
463	int	"Unknow DeviceID"	The "DeviceID" does not match the device ID.
464	int	"Unkonw ID"	Unknown "ID" (there is no `CustomizeID`/`PersonUUID`/`LibID` counterpart in the device list, or the value of "IdType" is not 0~2)
465	int	"Unkonw IdType"	Failed to get keyword "IdType".
466	int	"Unkonw BeginTime or EndTime"	Failed to get keyword "BeginTime" or "EndTime".
467	int	"Request Number too large"	Excessive number of requests.
468	int	"No Control Record"	No control logs
469	int	"Other"	Other

7.32 Search for the Total Number of Captured Logs Error Codes

code	Type	String	Description
461	int	"Unknow operator"	"operator" error
462	int	"Parameter error"	Failed to get keyword “info”.
463	int	"Unknow DeviceID"	The "DeviceID" does not match the device ID.

7.33 Capture Log Personnel Information Query Error Codes

code	Type	String	Description
461	int	"Unknow operator"	"operator" error
462	int	"Parameter error"	Failed to get keyword “info”.
463	int	"Unknow DeviceID"	The "DeviceID" does not match the device ID.
464	int	"Unkonw BeginNO"	Failed to get keyword "BeginNO".
465	int	"Unkonw RequestCount"	Failed to get keyword "RequestCount".
466	int	"Unkonw Picture"	Failed to get keyword "Picture".
467	int	"Other"	Other
468	int	"can't find person"	No capture logs were searched

7.34 Remote Open Door Error Codes

code	Type	String	Description
461	int	"Unknow operator"	"operator" error
462	int	"Parameter error"	Failed to get keyword “info”.
463	int	"Unknow DeviceID"	The "DeviceID" does not match the device ID.
464	int	"Unkonw Channel"	Failed to get the keyword "Channel".
465	int	"Not Remote Open Type"	The device is not in remote door-opening mode (the device is in face-opening mode).
466	int	"Parameter error no msg"	Failed to get keyword "msg".
467	int	"Parameter error no state"	Failed to get keyword "status".

7.35 Image Comparison Similarity Error Codes

code	Type	String	Description
461	int	"Unknow operator"	"operator" error
462	int	"Parameter error"	Failed to get keyword “info”.
463	int	"Unknow pic1info"	Failed to get keyword "picinfo1" (may be "picinfo1" field is missing or failed to decode image data base64 or failed to extract face feature value)
464	int	"Unknow pic2info"	Failed to get keyword "picinfo2" (may be "picinfo2" field is missing or image data base64 decoding failed or unable to extract face feature value)

7.36 Search for Local Face Database by Image Error Codes

code	Type	String	Description
461	int	"Unknow operator"	"operator" error
462	int	"Parameter error"	Failed to get keyword “info”.
463	int	"MaxSimilarity error"	Failed to get keyword "MaxSimilarity".
464	int	"MaxNum error"	Failed to get keyword "MaxNum".
465	int	"Unknow picinfo"	Failed to fetch keyword "picinfo" (may be "picinfo" field is missing or image data base64 decoding failed or face features could not be extracted)
466	int	"No Meet Conditions Person"	No eligible persons in the local face database

7.37 Detecting Image Face Error Codes

code	Type	String	Description
461	int	"Unknow operator"	"operator" error
462	int	"Parameter error"	Failed to get keyword “info”.
463	int	"Unknow picinfo"	Failed to get keyword "picinfo" (may be "picinfo" field is missing or image data base64 decoding failed)

7.38 Manual Snapshot Error Codes

code	Type	String	Description
461	int	"Unknow operator"	"operator" error
462	int	"Parameter error"	Failed to get keyword “info”.
463	int	"Unkonw DeviceID"	The "DeviceID" does not match the device ID.
464	int	"Server Error"	Server error (memory allocation failure)
465	int	"Timeout and can not get face pic"	Timeout, unable to get face map.
466	int	"This Version can not support this interface"	This interface is not supported in this version

7.39 Upload Ads Error Codes

code	Type	String	Description
461	int	"Unknow operator"	"operator" error
462	int	"cat not find info"	Failed to get keyword “info”.
463	int	"Unkonw DeviceID"	The "DeviceID" does not match the device ID.
464	int	"Unknow adsolt"	Failed to get keyword "adslot".
465	int	"adsolt out of range"	The value of "adslot" is out of range.
466	int	"Unknow picture type"	Unknown picture type
467	int	"AD picture download error"	Failed to download ad picture.

7.40 Drop Ads Error Codes

code	Type	String	Description
461	int	"Unknow operator"	"operator" error
462	int	"cat not find info"	Failed to get keyword “info”.
463	int	"Unkonw DeviceID"	The "DeviceID" does not match the device ID.
464	int	"Unknow adsolt"	Failed to get keyword "adslot".
465	int	"adsolt out of range"	The value of "adslot" is out of range.

7.41 Setting QR Code Image Error Codes

code	Type	String	Description
461	int	"Unknow operator"	"operator" error
462	int	"Parameter error"	Failed to get keyword “info”.
463	int	"can't find QRCodeData Param or Out of limit size"	Failed to get the keyword "QRCodeData", or the data exceeded the size limit.
464	int	"This Version can not support this interface"	This interface is not supported in this version

7.42 Setting Handshake Protocol Data Error Codes

code	Type	String	Description
461	int	"Unknow operator"	"operator" error
462	int	"Parameter error"	Failed to get keyword “info”.
463	int	"Unkonw DeviceID"	The "DeviceID" does not match the device ID.
464	int	"HttpSetHandSharkInfo Write Error"	Handshake protocol data write failure

7.43 Get Face Image Error Codes

code	Type	String	Description
461	int	"Unknow operator"	"operator" error
462	int	"cat not find info"	Failed to get keyword “info”.
463	int	"Unkonw Parameter"	Parameter error (May be failure to get keyword "dwfiletype", or failure to get keyword "dwfileindex", or failure to get keyword "dwfilepos", or "dwfileindex" exceeds maximum value, or "dwfiletype" is not 0)
464	int	"read picinfo fail"	Failed to read image data.

7.44 Setting GPS location Information Error Codes

code	Type	String	Description
461	int	"Device not support this interface"	This interface is not supported
462	int	"Can't get json key 'xxx'"	Parameter error, field 'xxx' not found
463	int	"Key 'xxx' value[yyy] is wrong"	Parameter error, field value error
464	int	"Save 'xxx' Error"	Failed to save configuration xxx

7.45 Device Keep Open Setting Error Codes

code	Type	String	Description
461	int	"Unknow operator"	"operator" error
462	int	"Parameter Info error"	Failed to get keyword “info”.
463	int	"Key[KeepOpen] value error"	Error in the value of the “KeepOpen” field.

7.46 Add/Modify Access Strategies Error Codes

code	Type	String	Description
461	int	"operator error"	"operator" error
462	int	"Parameter error"	Failed to get keyword “info”.
463	int	"can not find strategyID"	Failed to get keyword “strategyID”.
464	int	"strategyID can not be null or zero"	The value of strategyID is illegal.
465	int	"can not find startDate"	Failed to get keyword “startDate”.
466	int	"can not find endDate"	Failed to get keyword “endDate”.
467	int	"can not find allowCnt"	Failed to get keyword “allowCnt”.
468	int	"db update error"	Database update failure
469	int	"Time period setting greater than 6or3"	Number of time periods greater than 6

7.47 Delete Access Strategies Error Codes

code	Type	String	Description
461	int	"operator error"	"operator" error
462	int	"Parameter error"	Failed to get keyword “info”.
463	int	"can not find strategyID"	Failed to get keyword “strategyID”.
464	int	"strategyID can not be null or zero"	The value of strategyID is illegal.

errcode	Type	Description
461	int	The value of strategyID is illegal
462	int	Failed to delete the pass-through strategy corresponding to strategyID.

7.48 Query All Access Strategies IDs and Names Error Codes

code	Type	String	Description
461	int	"operator error"	"operator" error

7.49 Query Access Strategy Details by Access Strategy ID Error Codes

code	Type	String	Description
461	int	"operator error"	"operator" error
462	int	"Parameter error"	Failed to get keyword “info”.
463	int	"can not find strategyID"	Can't find the “strategyID” keyword.

7.50 Query All Associated Users by Access Strategy ID Error Codes

code	Type	String	Description
461	int	"operator error"	"operator" error
462	int	"Parameter error"	Failed to get keyword “info”.
463	int	"can not find strategyID"	Can't find the “strategyID” keyword.
464	int	"strategyID can not be null or zero"	The json array in “strategyID” is empty.

7.51 Personnel Binding Access Strategy Error Codes

code	Type	String	Description
461	int	"operator error"	"operator" error
462	int	"Parameter error"	Failed to get keyword “info”.
463	int	"can not find personInfo"	The keyword “personInfo” was not found.
464	int	"personsInfo Json num error"	The json array in “personInfo” is empty.

errcode	Type	Description
461	int	The person corresponding to PersonUUID/CustomizeID could not be found.
462	int	Can't find the “strategyID” keyword.
463	int	The json array in “strategyID” is empty.

7.52 Personnel Unbinding Access Strategy Error Codes

code	Type	String	Description
461	int	"operator error"	"operator" error
462	int	"Parameter error"	Failed to get keyword “info”.
463	int	"can not find personInfo"	The keyword “personInfo” was not found.
464	int	"personsInfo Json num error"	The json array in “personInfo” is empty.

errcode	Type	Description
461	int	The person corresponding to PersonUUID/CustomizeID could not be found.
462	int	Can't find the “strategyID” keyword.
463	int	The json array in “strategyID” is empty.