



LAN Platform Services

LAN System Interface Manual

Version and modification records

version	description	date
V1.0	create	2019-07-26
V1.1	Increase body temperature interface	2020-02-28

Interface operation instructions

Interface root address : [http:// device ip address : 8080 /](http://device_ip_address:8080/)

Interface form : Provide external services through HTTP requests.

Interface security : The device password (P) needs to be set for the first call to the interface , and any subsequent calls to the interface need to be passed in

The device password (pass) is used as the interface security verification key.

Interface returns

Interface general return description :

```

public class ResultInfo<T> {
private int result;// Indicates whether the interface is connected, 1 is
successful, 0 is failed, usually as long as the device server can respond,
the value is 1
private Boolean success;// Whether the operation was successful, success is true,
failure is false private T
data;// Business data returned by the interface. The type can be numeric,
string, or collection.

```

The interface return examples involved in the document, the return data of individual interfaces will be slightly adjusted, and the actual return results shall prevail.

Interface operation

Set device password

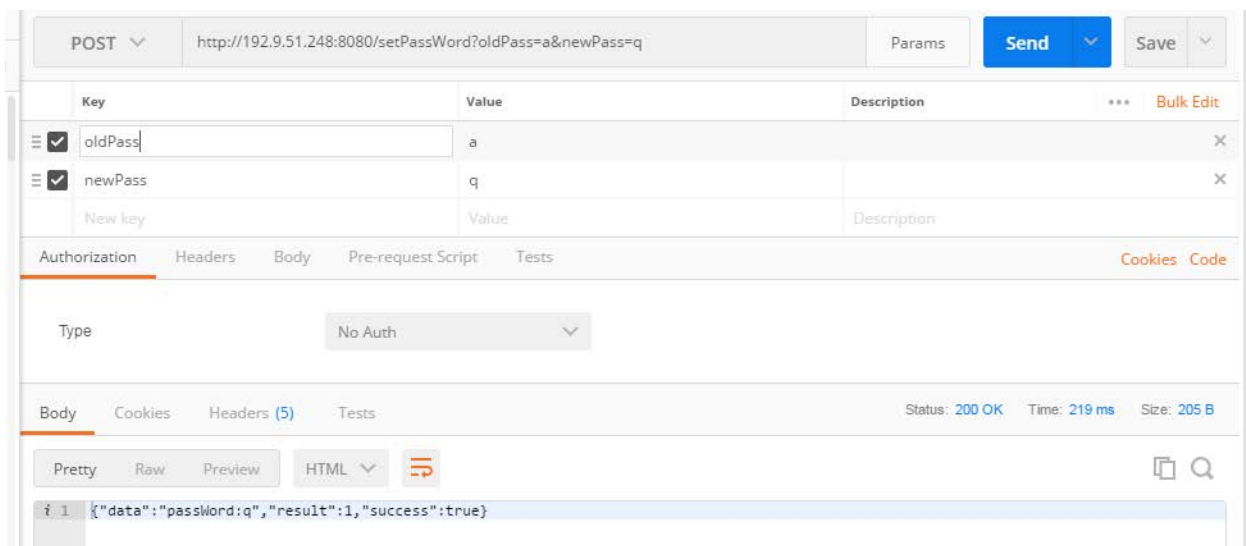
Way: POST request

url: [http:// device IP: 8080 / setPassWord](http://device IP: 8080 / setPassWord)

parameter

Field	Types of	Must pass	description
oldPass	String	Y	old password
newPass	String	Y	New password, password cannot be empty or space

Postman example:



Return example:

```
{
  "data": "passWord: q", // Device password
  "result": 1, // Interface communication
  "success": true // The device password is set successfully
}
```

Parameter configuration

Way: POST request

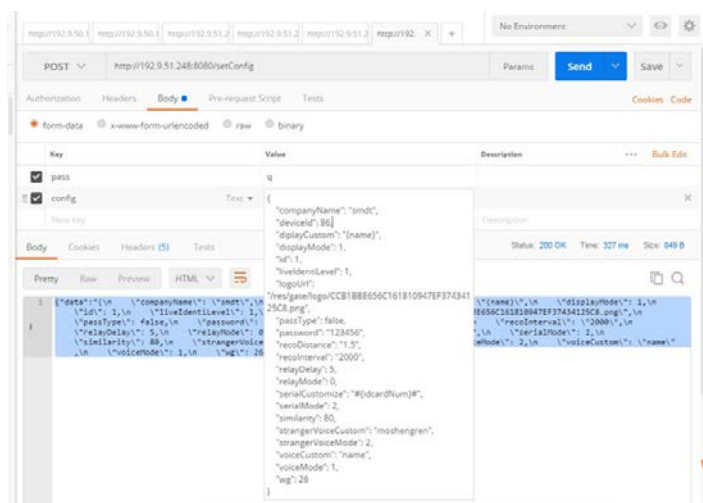
URL: [http:// device IP: 8080 / setConfig](http://device IP: 8080 / setConfig)

Field	Types of	Must pass	description
pass	String	Y	Device password
config	Json	Y	Parameter configuration Json The following are the parameters of json

Note: POST request parameter in the body in Parameters.

Field	type	description
companyName	String	company name
deviceId	int	Associated device id
diplayCustom	String	Display customization, default is {name}
displayMode	int	Display mode, 1: display name, 100: custom
id	int	
liveIdentiLevel	int	Living body recognition level, default quick mode, 0 no live body, 1 fast mode cannot reject photos, 2 can reject some photos, 3 can reject photos and adaptation
passType	boolean	Access type false in true out
password	String	Passcode
recoDistance	String	Recognition distance (0.5 ~ 3)
recoInterval	String	Identification interval (2000 ~ 10000)
relayDelay	Int	Relay delay x seconds automatically closed (5 ~ 63)
relayMode	Int	Relay mode, 0: indicates the automatic closing mode, that is, the relay will automatically close after a delay of X seconds after opening the relay; 1: indicates the non-automatic closing mode, that is, open Does not close automatically after the relay
serialCustomize	String	Serial port definition, default is "# {idcardNum} #", unlimited length
similarity	Int	Similarity, default 80,30 ~ 100 adjustment
strangerVoiceCustom	String	Strange voice customization, default is stranger recognition, within 32 characters
strangerVoiceMode	Int	Stranger voice mode, default stranger alarm, 1 no voice announcement required, 2 stranger alarm, 100 custom
voiceCustom	String	Voice customization, default is "name", within 32 characters
voiceMode	Int	Voice mode, 1 does not need to broadcast, 2 announces the name, 100 custom
wg	String	Wigan 26, 34

Pastman example



Return example:

```

" data ": { //Setparameters"companyName": "smdt", "deviceId": 86, "diplayCustom": "{name}",
" displayMode ": 1,

  " id ": 1,

  " livIdentiLevel ": 1, " passType ":
    false,

  " password ": " 123456 ",

  " recoDistance ": " 1.5 ",

  " recoInterval ": " 2000 ",

  " relayDelay ":5,

  " relayMode ":0,

  " serialCustomize ": "#{idcardNum}#",

  " serialMode ": 2,

  " similarity ": 80,

  " strangerVoiceCustom ": "moshengren",

  " strangerVoiceMode ": 2, " voiceCustom ": "name",

  " voiceMode ": 1,

  " wg ": 26

  " result ":1,// interface connected

  " success ":true //Set up
  successfully

```

Person registration

Way: POST request

url : http://device_ip:8080/person/create

parameter:

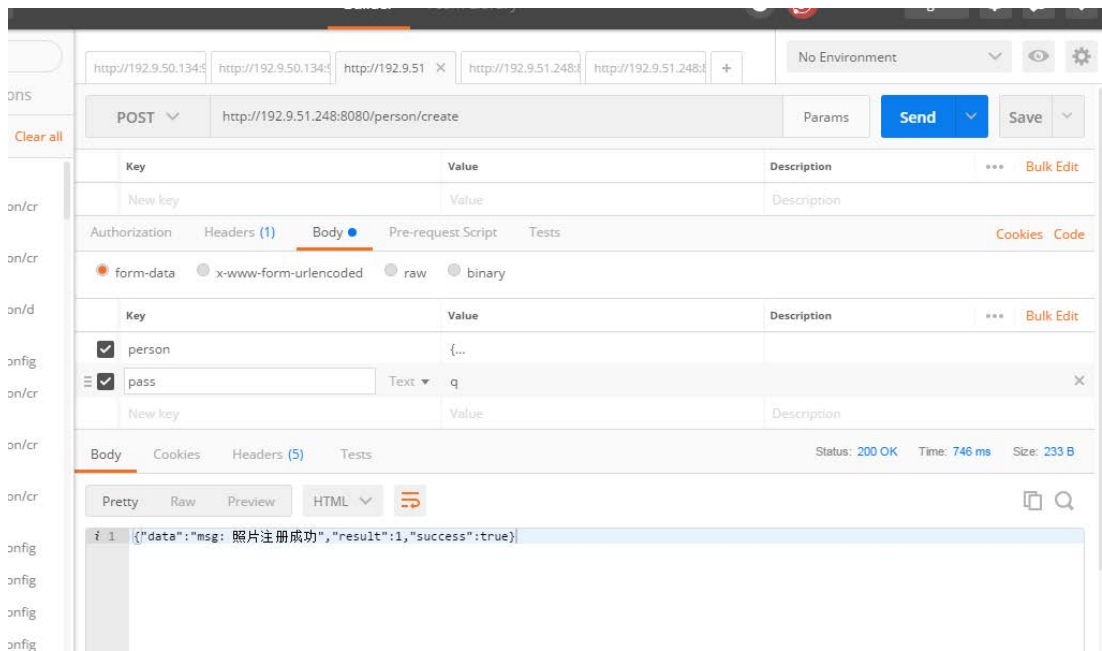
Field	Types of	Must pass	description
pass	String	Y	Device password
person	Json	Y	Personnel information Json The following are the parameters of json

Note: POST request parameters are placed in the body parameters

Parameter description

field	tpye	description
age	int	age
imgBase64	String	Base64 string for photos
name	String	name
prescription	String	Passage time
sex	sex	Gender
type	Int	The default type is 1, 1 is visitor, 2 employees,
vipID	Int	Personnel ID, the assigned ID must be unique
welCome	String	Identification message prompt displayed
wn	String	Reserved, optional

Pastman example



Return example

```
{
  "data": "msg: person
registration successful",
  "result": 1,
  "success": true
}
```

Personnel information editing

Way: POST request

url : http:// device ip: 8080 / person / update

parameter:

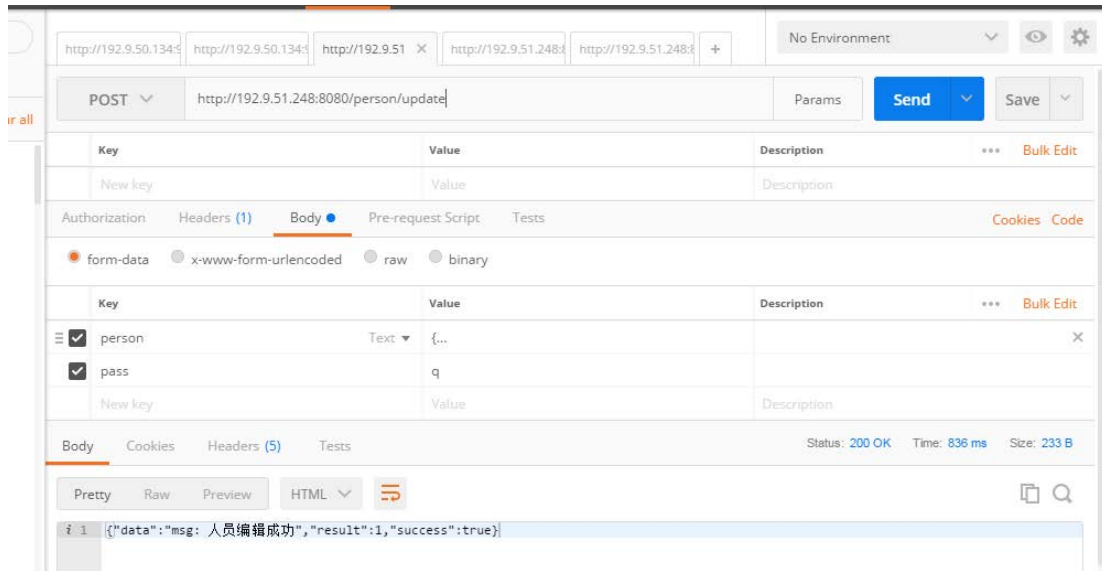
Field	Types of	Must pass	description
pass	String	Y	Device password
person	Json	Y	Personnel information Json The following are the parameters of json

Note: POST request parameter in the body in

Person parameter description

Field	Types of	description
age	int	age
Field	Types of	description
imgBase64	String	Base64 string of photos
name	String	first name
prescription	String	Traffic period
sex	sex	gender
type	Int	The default type is 1 , 1 is a visitor, 2 employees, 3 for blacklist
vipID	Int	Create a person's id , find the person based on the idedit
welCome	String	Display of identification information
wn	String	Reserved, not required

Pastman example:



Return example:

```
{
  "data": "msg: 人员编辑成功",
  "result": 1,
  "success": true
}
```

Delete person

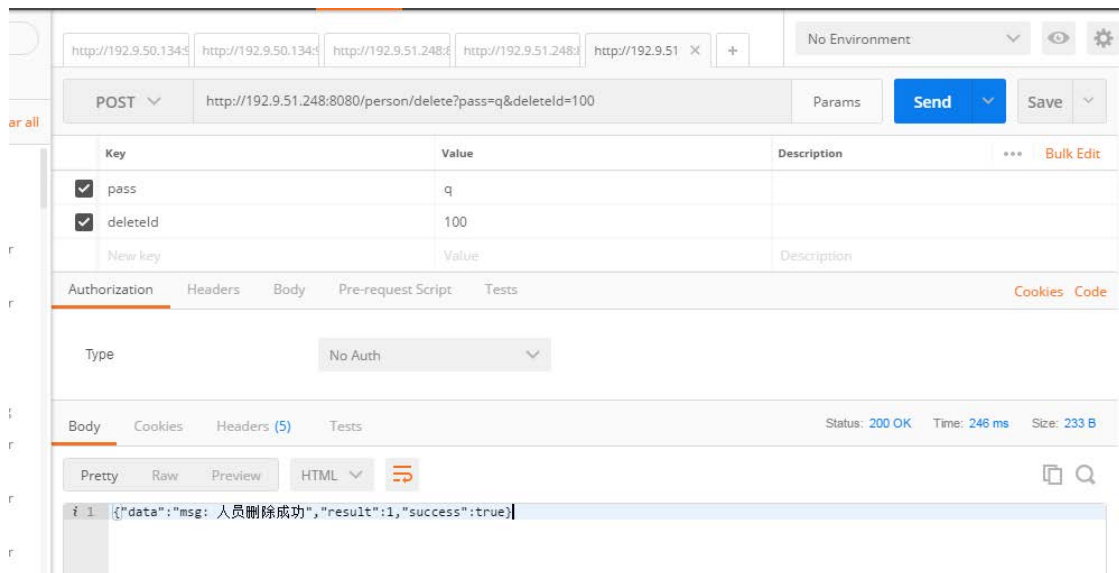
Way: POST request

url : [http:// device_ip: 8080 / person / delete](http://device_ip: 8080 / person / delete)

parameter

Field	Types of	description
pass	String	Equipment moma
deleteld	String	Deleted person id

Pastman example



Return example

```
{
  "data": "msg: 人员删除成功",
  "result": 1,
  "success": true
}
```

Access record

Way: POST request

url : http://device ip: 8080 / newFindRecords

parameter

Field	Types of	description
pass	String	Device password
startTime	String	Starting time: 2019-07-11 12:00
endTime	String	End time: 2019-07-11 17:00

Pastman example

The screenshot shows a REST client interface with the following details:

- Method:** POST
- URL:** http://192.9.51.248:8080/newFindRecords?pass=q&startTime=2019-07-11 12:00&endTime=...
- Params:** A table with the following entries:

Key	Value	Description
pass	q	
startTime	2019-07-11 12:00	
endTime	2019-07-11 17:00	
New key	Value	Description
- Form Data:** form-data (selected), x-www-form-urlencoded, raw, binary
- Body:** A JSON array of objects, each containing fields like name, birthDate, idCardNum, currentTime, imageFlag, and imageName. The objects are separated by commas and enclosed in square brackets.

```
1 [{"name": "\pdf", "birthDate": "\", "idCardNum": "\", "currentTime": 1562817881485, "imageFlag": 0, "imageName": "\1_0_1562817881485.jpg", "type": 1}, {"name": "\pdf", "birthDate": "\", "idCardNum": "\", "currentTime": 1562817883491, "imageFlag": 0, "imageName": "\1_0_1562817883491.jpg", "type": 1}, {"name": "\pdf", "birthDate": "\", "idCardNum": "\", "currentTime": 1562817888554, "imageFlag": 0, "imageName": "\1_0_1562817888554.jpg", "type": 1}, {"name": "\pdf", "birthDate": "\", "idCardNum": "\", "currentTime": 1562817890577, "imageFlag": 0, "imageName": "\1_0_1562817890577.jpg", "type": 1}, {"name": "\pdf", "birthDate": "\", "idCardNum": "\", "currentTime": 1562817892610, "imageFlag": 0, "imageName": "\1_0_1562817892610.jpg", "type": 1}, {"name": "\pdf", "birthDate": "\", "idCardNum": "\", "currentTime": 1562817894657, "imageFlag": 0, "imageName": "\1_0_1562817894657.jpg", "type": 1}, {"name": "\pdf", "birthDate": "\", "idCardNum": "\", "currentTime": 1562819076594, "imageFlag": 0, "imageName": "\1_0_1562819076594.jpg", "type": 1}, {"name": "\pdf", "birthDate": "\", "idCardNum": "\", "currentTime": 1562819084252, "imageFlag": 0, "imageName": "\1_0_1562819084252.jpg", "type": 1}, {"name": "\pdf", "birthDate": "\", "idCardNum": "\", "currentTime": 1562819086284, "imageFlag": 0, "imageName": "\1_0_1562819086284.jpg", "type": 1}]}]
```
- Status:** 200 OK, Time: 366 ms, Size: 2.63 KB
- Response Body:** Pretty, Raw, Preview, HTML (selected)

Return example

```
{
  "data": [
    {
      "birthDate": "",
      "currentTime": 1562819094334, //通行时间, 毫秒
      "idCardNum": "",
      "imageFlag": 0,
      "imageName": "1_0_1562819094334.jpg", //通行拍的相片
      "name": "pdf", //通行人员名字
      "type": 1 //类型
      "temprature": "36.5" //通行时的体温
    },
    {
      "birthDate": "",
      "currentTime":
      1562824109687,
      "idCardNum": "",
```

Modify logo

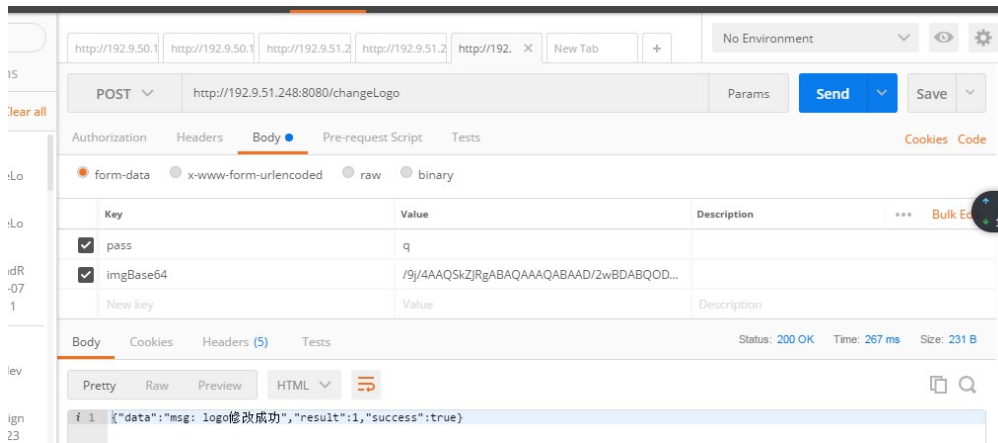
Way: POST request

url : http://device_ip:8080/changeLogo

parameter

Field	Types of	description
pass	String	Device password
imgBase64	String	Logo icon Base64 , only supports png , jpg

Pastman example



Return example

```
{  
  \"data\": \"msg: logo 修改成功\",  
  \"result\": 1,  
  \"success\": true  
}
```

Get device mac address

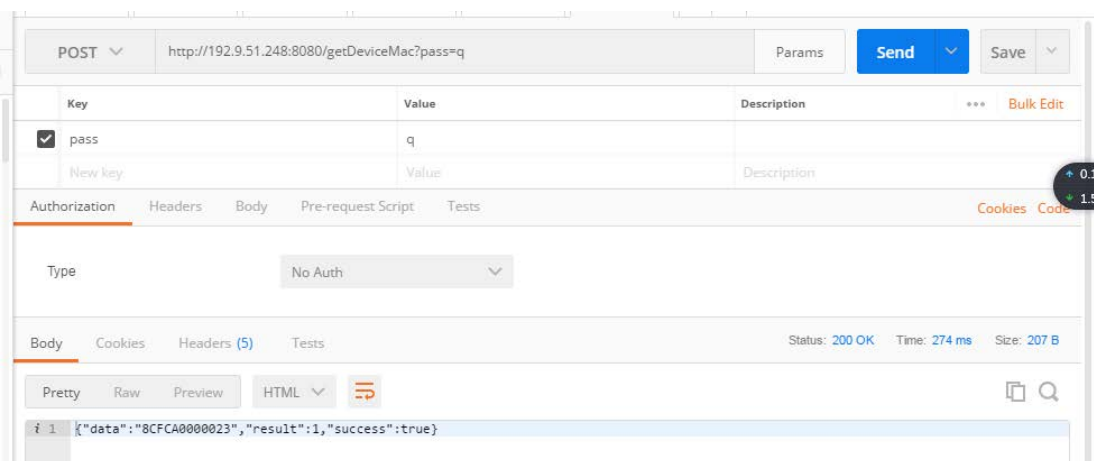
Way: POST request

url : [http:// device ip: 8080 / getDeviceMac](http://device ip: 8080 / getDeviceMac)

parameter

Field	Types of	description
Pass	String	Device password

Pastman example



Return example

```
{
  "data": "8CFCA0000023",
  "result": 1,
  "success":
```

Body temperature and mask parameter configuration

Way: POST request

url : http://device_ip:8080/tempAndMaskSetting

Note: POST request parameter in the body

field	tpye	description
pass	String	Device password
config	Json	Parameter information JSON is the parameter of JSON
isBodyTempAlarm	int	1 turn on temperature alarm 0 turn off temperature alarm
isBodyTempStart	Int	1 on body temperature 0 off body temperature
standardBodyTemp	String	The temperature threshold value is 37.3 (accurate to one decimal place). When the temperature detection and temperature alarm are turned on, it is detected that the temperature exceeds the threshold value, and the sound alarm is played
isHighFeverAdopt	int	High fever pass (0: no pass 1: pass) for door opening (range: 38.5-43.0)
isLowFeverAdopt	int	Whether low fever temperature pass (0: no pass 1: pass) is used for door opening (range 37.3-38.5)
isLowTempAdopt	int	Whether low temperature pass (0: no pass 1: pass) is used for door opening (range 30.0-below)
isStandardTempAdopt	int	Whether normal body temperature passes (0: no pass 1: pass) is used for door opening (range 36.1-37.2)
isWearingMask	int	Wear mask detection switch (0: No 1: Yes))

Pastman example

The screenshot shows a Postman interface for a POST request to `http://192.9.50.234:8080/tempAndMaskSetting?pass=123456&config={...}`. The request body is a JSON object with the following fields:

```
{
  "deviceId": 88,
  "id": 10,
  "isBodyTempAlarm": 1,
  "isBodyTempStart": 1,
  "isHighFeverAdopt": 0,
  "isLowFeverAdopt": 0,
  "isLowTempAdopt": 0,
  "isStandardTempAdopt": 1,
  "isWearingMask": 0,
  "standardBodyTemp": "37.3"
}
```

The response is a 200 OK status with a JSON body:

```
{
  "data": "{\n  \"deviceId\": 88,\n  \"id\": 10,\n  \"isBodyTempAlarm\": 1,\n  \"isBodyTempStart\": 1,\n  \"isHighFeverAdopt\": 0,\n  \"isLowFeverAdopt\": 0,\n  \"isLowTempAdopt\": 0,\n  \"isStandardTempAdopt\": 1,\n  \"isWearingMask\": 0,\n  \"standardBodyTemp\": \"37.3\"\n}",
  "result": 1,
  "success": true
}
```

Return example

```
{
  "data":
  "{ \"deviceId
  \": 88,
  \"id\": 10,
  \"isBodyTempAlarm\": 1,
  \"isBodyTempStart\": 1,
  \"isHighFeverAdopt\": 0,
  \"isLowFeverAdopt\":0,
  \"isLowTempAdopt\":0,
  \"isStandardTempAdopt\": 1,
  \"isWearingMask\": 0,
  \"standardBodyTemp\": \"37.3\"
}",
  "result": 1,
  "success":
  true
}
```