onetouch
communication & security

HTTP API Manual v1.0

Content

Introduction

HTTP API is an application interface which is designed for function control of selected Onetouch intercom via the HTTP. It enables Onetouch intercoms to be integrated easily with third party products, such as home automation, security and monitoring systems, etc.

HTTP API Release Notes

Time	Version	Changes
2021-09	V1.0	New feature: 1. Add GET interface 2. Add http standard authentication method: Basic and Digest 3. Add system module: system, firmware, config, dialreplace, log, calllog, doorlog, pcap

HTTP API Description

Onetouch HTTP API is a standard protocol based on the HTTP GET/POST method. Any request can interact with the device via GET or POST. The format of the interactive message is JSON, and the file transfer mode uses the standard FORM upload mode.

HTTP API Request

All commands are sent via HTTP/HTTPS to the intercom address with absolute path completed with the /api prefix. Which protocol you choose depends on the current intercom settings in the Services / HTTP API section. The HTTP API functions are assigned to services with defined security levels including the TLS connection request (i.e. HTTPS).
The absolute path includes the target system module (system, firmware, config, relay, etc.) and the action name (status, get, add, etc.). To be accepted by the intercom, a request has to include the method and absolute path specification followed by the Host header.

Example:

http://IP address/api/Target/Action?Param

HTTP API Method

Onetouch intercom applies the following two HTTP methods: GET and POST. The GET and POST methods are equivalent from the viewpoint of HTTP API but use different parameter transfers (refer to the next subsection).

Method: GET

To request intercom content download or general command execution via path.
Example Request:

```
http://192.168.1.163/api/contact/add?name=Testing&phone=110
```

Example Return:

```
{
   "retcode": 0,
   "action": 0,
   "message": "OK",
  "data": {
     "id": 0,
      "name": "Testing",
      "phone": "110"
   }
}
```

Method: POST

Post mode is the same as Get mode that requests intercom content download or general command execution, but Post moves the Query String parameters into the json for processing

Example Request:

```
http://Device-IP/api/
{
     "target": "contact",
     "action": "add",
     "data" : {
          "name": "Testing",
          "phone": "110"
     }
}
```

Example Return:

```
{
   "retcode": 0,
   "action": 0,
   "message": "OK",
  "data": {
     "id": 0,
      "name": "Testing",
      "phone": "110"
   }
}
```

HTTP API Reply

Replies to requests are mostly in the JSON format, and Files exported according to requests are saved in file format without compression (e.g config is saved in .tgz file type , rfkey is saved in .xml file type, etc). The Content-Type header specifies the response format. Three basic reply types are defined for JSON.

Positive Reply without Parameters

This reply is sent in case a request has been executed successfully for functions that do not return any parameters. This reply is always combined with the HTTP status code 200OK.

```
{
      "retcode": 0,
      "action": "xxxx",
      "message": "OK"
}
```

Positive Reply with Parameters

This reply is sent in case a request has been executed successfully for functions that return supplementary parameters. The result item includes other reply parameters related to the function. This reply is always combined with the HTTP status code 200OK.

```
{
    "retcode": 0,
    "action": "xxxxx",
    "message": "OK",
    "data":
    {
        ......
    }
}
```

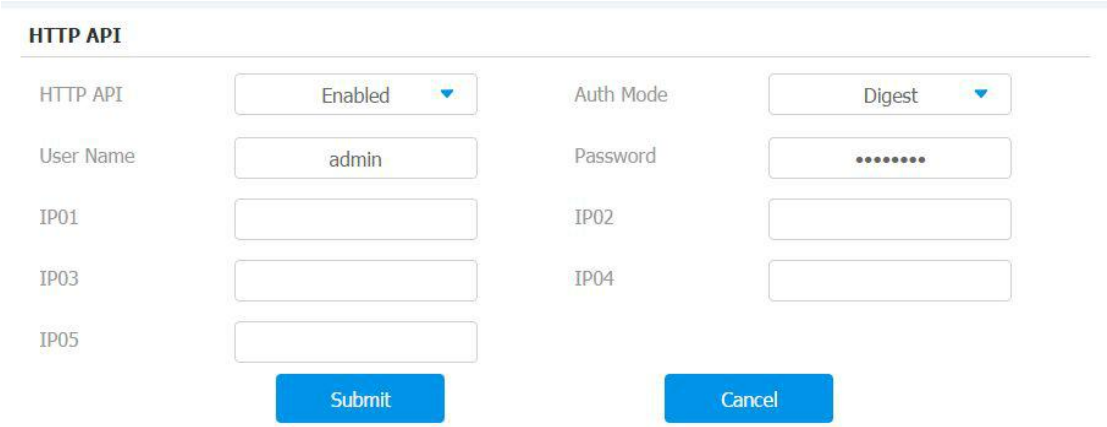
Negative Reply

This reply is sent in case an error occurs during request processing. The reply specifies the error code (code), text description (description) and error details if necessary (param). The reply can be combined with the HTTP status code 200 OK or 401 Unauthorized Required.

```
{
    "retcode": -1,
    "action": "unknow",
    "message": "unsupport action"
}
```

HTTP API Configuration

HTTP API configurations for Onetouch intercom are defined via web interface PATH: Intercom->HTTP API to disable/enable a service and select the user authentication method



Item	Description
HTTP API	Disabled: system allways return HTTP 403 Forbidden status Enabled：enable http api
Auth Mode	None: no authentication method Normal: (reserved) WhiteList: Whitelist model Basic: http Basic standard authentication method Digets: http Digets standard authentication method Token : (reserved)
User Name	UserName: admin (default), used in basic and digest auth mode
Password	Password: httpapi (default), used in basic and digest auth mode
IP01	White List IP
IP02	White List IP
IP03	White List IP
IP04	White List IP
IP05	White List IP

HTTP API Authentication Method

None

No authentication is required for http api, it is only used by demo testing.

WHITE LIST

The whitelist is suitable for operation in the LAN, by judging the IP address of the visitor to confirm whether to allow access to the HTTP API.

Basic

In Authorization field of Http request header, use Base64 encode method for the information of username and password.

For more detailed information, please refer to RFC2671

Digest

Password encryption method, only supports MD5.
In Authorization field of Http request header:
WWW-Authenticate: Digest realm="HTTPAPI",qop="auth,auth-int",nonce="xx", opaque="xx"
For more detailed information, please refer to RFC2671

Token

(Reserved)

HTTP API Functions

The table below provides a list of all available HTTP API functions including:

- **the HTTP request absolute path**
- **the supported HTTP methods**

Absolute path	API Method
/api/system/info	GET/POST
/api/system/status	GET/POST
/api/system/reboot	GET/POST
/api/firmware/status	GET/POST
/api/firmware/prepare	GET/POST
/api/firmware/upload	POST
/api/firmware/upgrade	GET/POST
/api/config/get	GET/POST
/api/config/set	POST
/api/config/export	GET
/api/config/import	POST
/api/config/reset_factory	GET/POST
/api/relay/get	GET/POST
/api/relay/set	POST
/api/relay/status	GET/POST
/api/relay/trig	GET/POST
/api/input/get	GET/POST
/api/input/set	POST
/api/input/status	GET/POST
/api/publiccode/get	GET/POST
/api/publiccode/set	POST
/api/privatekey/get	GET/POST
/api/privatekey/add	POST
/api/privatekey/set	POST
/api/privatekey/del	POST
/api/privatekey/clear	GET/POST
/api/privatekey/export	GET
/api/privatekey/import	POST
/api/rfkey/get	GET/POST
/api/rfkey/add	POST
/api/rfkey/set	POST
/api/rfkey/del	POST
/api/rfkey/clear	GET/POST
/api/rfkey/export	GET
/api/rfkey/import	POST

/api/contact/get	GET/POST
/api/contact/add	POST
/api/contact/set	POST
/api/contact/del	POST
/api/contact/clear	GET/POST
/api/contact/export	GET
/api/contact/import	POST
/api/group/get	GET/POST
/api/group/add	POST
/api/group/set	POST
/api/group/clear	GET/POST
/api/call/status	GET/POST
/api/call/dial	GET/POST
/api/call/hangup	GET/POST
/api/sip/get	GET/POST
/api/sip/set	POST
/api/sip/status	GET/POST
/api/dialreplace/get	GET/POST
/api/dialreplace/add	POST
/api/dialreplace/set	POST
/api/dialreplace/del	POST
/api/dialreplace/clear	GET/POST
/api/dialreplace/export	GET
/api/dialreplace/import	GET/POST
/api/log/get	GET/POST
/api/log/set	POST
/api/log/export	GET
/api/log/clear	GET/POST
/api/calllog/get	GET/POST
/api/calllog/del	POST
/api/calllog/clear	GET/POST
/api/calllog/export	GET
/api/doorlog/get	GET/POST
/api/doorlog/del	POST
/api/doorlog/clear	GET/POST
/api/doorlog/export	GET
/api/pcap/status	GET/POST
/api/pcap/start	GET/POST
/api/pcap/stop	GET/POST
/api/pcap/export	GET

Api system

Section	Param	Description
Status	Model	ProductModel
	MAC	MAC address
	FirmwareVersion	Firmware version:
	HardwareVersion	Hardware version
Lan	PortType	0: DHCP 1: Static
	LinkStatus	0: Disconnect 1: Connected 2: IP Conflict
	IPAddress	192.168.1.78
	SubnetMask	255.255.255.0
	Gateway	192.168.1.1
	DNS1	192.168.1.1
	DNS2	192.168.1.1
Account1	UserName	114
	SipServer	192.168.1.115
	Status	-1: unknow 0: disable 1: registering 2: registed 3: registe failed
Account2	UserName	115
	SipServer	192.168.1.115
	Status	-1: unknow 0: disable 1: registering 2: registed 3: registe failed

Api system info

The GET or POST method can be used for this function.

The reply is in the application/json format and includes the following information on the device:

Example Request:

```
GET: /api/system/info
```

```
POST:
{
"target": "system",
"action": "info"
}
```

Example Return:

```
{
    "retcode": 0,
    "action": "info",
    "message": "OK",
    "data": {
        "Status": {
            "Model": “OT-IP-LP-C9",
            "MAC": "0C:11:05:08:93:C6",
            "FirmwareVersion": "29.31.1.617",
            "HardwareVersion": "29.3.0"
        },
        "Lan": {
            "PortType": "0",
            "LinkStatus": "1",
            "IPAddress": "192.168.35.29",
            "SubnetMask": "255.255.255.0",
            "Gateway": "192.168.35.1",
            "DNS1": "192.168.1.1",
            "DNS2": "192.168.1.1"
        },
        "Account1": {
            "UserName": "",
            "SipServer": "",
            "Status": "0"
        },
        "Account2": {
            "UserName": "",
            "SipServer": "",
            "Status": "0"
        }
    }
}
```

Api system status

The GET or POST method can be used for this function.

Param	Description
SystemTime	Unix time
UpTime	Device online time

Example Request:

```
GET: /api/system/status
```

```
POST:
{
    "target": "system",
     "action": "status"
}
```

Example Return:

```
{
     "retcode": 0,
     "action": "status",
     "message": "OK",
     "data": {
          "SystemTime": 1544336389,
          "UpTime": 137
     }
}
```

Api system reboot

Example Request:

```
GET: /api/system/reboot
```

```
POST:
{
    "target": "system",
     "action": "reboot"
}
```

Example Return:

```
{
      "retcode": 0,
      "action": "status",
      "message": "OK",
}
```

Api firmware param

Param	Description
md5	string, optional md5sum of the firmware only use in action “upgraer”
silent	silent 0 - ui will display upgrade process 1 - upload in backgroup only use in action “upload” default: 0

Api firmware status

The GET or POST method can be used for this function.

Example Request:

```
GET: /api/firmware/status
```

```
POST:
{
     "target": "firmware",
     "action": "status",
}
```

Example Return:

```
{
      "retcode": 0,
      "action": "status",
      "message": "OK",
      "data": {
            "FirmwareVersion": "29.31.101.624",
            "HardwareVersion": "29.3.0",
            "MAC": "0C:11:05:07:8B:71",
            "Model": "OT-IP-LP-C9"
      }
}
```


Api firmware prepare

Tell device into upgrade mode, this must do before upload and upgrade action The GET or POST method can be used for this function.

Example Request:

```
GET: /api/firmware/prepare
```

```
POST:
{
     "target": "firmware",
     "action": "prepare",
}
```

Example Return:


```
{
      "retcode": 0,
      "action": "prepare",
      "message": "OK"
}
```


Api firmware upload

Upload firmware, Only POST method can be used for this function.
PATH: /api/firmware/upload

PARAM: /api/firmware/upload?silent=0

Api firmware upgrade

The GET or POST method can be used for this function.

Example Request(without MD5 param):

```
GET: /api/firmware/upgrade
```

```
POST:
{
    "target": "firmware",
    "action": "upgrade",
}
```

Example Return(with MD5 param):

```
{
    "retcode": 0,
    "action": "upgrade",
    "message": "OK"
}
```

Example Request(with MD5 param):

```
GET: /api/firmware/upgrade?md5=25e317773f308e446cc84c503a6d1f85
```

```
POST:
{
    "target": "firmware",
    "action": "upgrade",
    "data": {
        "md5" : "25e317773f308e446cc84c503a6d1f85"
    }
}
```

Example Return(with MD5 param):

```
{
    "retcode": -1,
    "action": "upgrade",
    "message": "Wrong MD5",
    "data": {
        "md5": "25e317773f308e446cc84c503a6d1f83",
        "md5_calc": "25e317773f308e446cc84c503a6d1f85"
    }
}
```

Api config get

Note: The configuration for CardSetting、PrivateKey、DialReplace、Contact、Group are not included in this section.

The GET or POST method can be used for this function.

To get all configurations.
Example Request 1:

```
GET: /api/config/get
```

```
POST:
{
"target": "config",
"action": "get"
}
```

Example Return 1:

```
{
      "retcode": 0,
      "action": "get",
      "message": "OK",
      "data": {
            "Config.Account1.ADVANCED.ShowMissedCall": "1",
            "Config.Account1.ANONYMOUS_CALL.Enable": "0",
            "Config.Account1.ANONYMOUS_CALL.OffCode": "",
            "Config.Account1.ANONYMOUS_CALL.OnCode": "",
            .......
"Config.Account1.ANONYMOUS_CALL.PrivacyHeaderValue": "id",
     },
}
```

Api config set

Api config export

To export the configuration, equivalent to operate via web interface path: Upgrader- > Advanced-> Config File->Export, and the exported configuration is encrypted which cannot be modified.

The GET method can be used for this function.

Example Request:

```
GET: /api/config/export
```

Example Return:

```
Config.tgz file type
```

Api config import

To import the configuration, equivalent to operate via web interface path: Upgrader-> Advanced-> Config File->Import, and only tgz file can be accepted.

Note: Device will reboot after import the configuration successfully.

Only POST method can be used for this function.

Api config reset_factory

The GET or POST method can be used for this function.

Example Request:

```
GET: /api/config/reset_factory
```

```
POST:
{
    "target": "config",
    "action": "reset_factory"
}
```

Example Return:

```
{
    "retcode": 0,
    "action": "reset_factory",
    "message": "OK"
}
```

Api relay

Params	Description
Config.DoorSetting.RELAY.TypeA	Relay Open Type 0 - NC - COM Open 1- NO -COM Open
Config.DoorSetting.RELAY.TypeB	
Config.DoorSetting.RELAY.TypeC	
Config.DoorSetting.RELAY.HoldDelayA	Relay Hold Delay 0 ~ 60 second
Config.DoorSetting.RELAY.HoldDelayB	

Config.DoorSetting.RELAY.HoldDelayC	
Config.DoorSetting.RELAY.TriggerDelayA	Relay Trigger Delay 0 ~ 60 second Delay X second to trigger relay
Config.DoorSetting.RELAY.TriggerDelayB	
Config.DoorSetting.RELAY.TriggerDelayC	
Config.DoorSetting.DTMF.Enable	Config.DoorSetting.DTMF.Option n=1: 1 digit dtmf, refer to Config.DoorSetting.DTMF.CodeX n=2~4: 2~4 digits dtmf, refer to Config.DoorSetting.DTMF.MultiCodeX
Config.DoorSetting.DTMF.Code1	one dtmf code
Config.DoorSetting.DTMF.Code2	
Config.DoorSetting.DTMF.Code3	
Config.DoorSetting.DTMF.MultiCode1	2 ~ 4 digits dtmf code
Config.DoorSetting.DTMF.MultiCode2	
Config.DoorSetting.DTMF.MultiCode3	

Api relay get

The GET or POST method can be used for this function.

Example Request:

```
GET: /api/relay/get
```

```
POST:
{
    "target": "relay",
    "action": "get"
}
```

Example Return:

```
{
    "retcode": 0,
    "action": "get",
    "message": "OK",
    "data": {
        "Config.DoorSetting.RELAY.TypeA": "0",
        "Config.DoorSetting.RELAY.TypeB": "0",
        "Config.DoorSetting.RELAY.TypeC": "0",
        "Config.DoorSetting.RELAY.HoldDelayA": "5",
        "Config.DoorSetting.RELAY.HoldDelayB": "5",
        "Config.DoorSetting.RELAY.HoldDelayC": "5",
        "Config.DoorSetting.RELAY.TriggerDelayA": "0",
        "Config.DoorSetting.RELAY.TriggerDelayB": "0",
        "Config.DoorSetting.RELAY.TriggerDelayC": "0",
        "Config.DoorSetting.RELAY.NameA": "RelayA",
        "Config.DoorSetting.RELAY.NameB": "RelayB",
        "Config.DoorSetting.RELAY.NameC": "RelayC",
        "Config.DoorSetting.RELAY.HttpTrig": "0",
        "Config.DoorSetting.RELAY.UserName": "",
        "Config.DoorSetting.RELAY.Password": "",
        "Config.DoorSetting.DTMF.Enable": "1",
        "Config.DoorSetting.DTMF.Option": "0",
        "Config.DoorSetting.DTMF.Code1": "0",
        "Config.DoorSetting.DTMF.Code2": "1",
        "Config.DoorSetting.DTMF.Code3": "2",
        "Config.DoorSetting.DTMF.MultiCode1": "010",
        "Config.DoorSetting.DTMF.MultiCode2": "012",
        "Config.DoorSetting.DTMF.MultiCode3": "013"
    }
}
```


Api relay set

Only POST method can be used for this function.

Example Request:

```
POST:
{
      "target": "relay",
      "action": "set",
      "data" : {
          "Config.DoorSetting.RELAY.HoldDelayA": "8"
      }
}
```

Example Return:

```
{
      "retcode": 0,
      "action": "set",
      "message": "OK"
}
```

Api relay status

The GET or POST method can be used for this function.

Example Request:

```
GET: /api/relay/status
```

```
POST:
{
      "target": "relay",
      "action": "status"

}
```

Example Return:

```
{
      "retcode": 0,
      "action": "status",
      "message": "OK",
      "data": {
            "RelayA": 0,
            "RelayB": 0,
            "RelayC": 0
      }
}
```


Api relay

Params	Description
mode	0:Auto Close 1: Manual
num	1 Relay A 2 Relay B 3 Relay C
level	0 NO - COM 1NC-COM
delay	0~65535(Second)

Api relay trig

The GET or POST method can be used for this function.

Example Request:

```
GET: /api/relay/trig?mode=1&num=1&level=1&delay=5
```

```
POST:
{
      "target": "relay",
      "action": "trig",
      "data" : {
          "mode": 1,
          "num": 1,
          "level": 1,
          "delay": 5
      }
}
```

Example Return:

```
{
      "retcode": 0,
      "action": "trig",
      "message": "OK"
}
```

Api input

Params	Description
Config.DoorSetting.INPUT.EnableA	Input Enable
Config.DoorSetting.INPUT.EnableB	
Config.DoorSetting.INPUT.EnableC	
Config.DoorSetting.INPUT.OptionA	Trig Option: 0 Low Trig 1 High Trig
Config.DoorSetting.INPUT.OptionB	
Config.DoorSetting.INPUT.OptionC	
Config.DoorSetting.INPUT.DelayA	Trig Delay 0 ~ 65536 second 0: trig now 1 ~ 65536: trig after X second
Config.DoorSetting.INPUT.DelayB	
Config.DoorSetting.INPUT.DelayC	
Config.DoorSetting.INPUT.RelayIdA	Trig which relay id 1 - Relay A 2 - Relay B 3 - Relay C
Config.DoorSetting.INPUT.RelayIdB	
Config.DoorSetting.INPUT.RelayIdC	
Config.DoorSetting.INPUT.ActionUrlA	Action URL More Action URL Param see Action URL document
Config.DoorSetting.INPUT.ActionUrlB	
Config.DoorSetting.INPUT.ActionUrlC	
Config.DoorSetting.INPUT.FtpEnableA	when input trig, send capture files to

Config.DoorSetting.INPUT.FtpEnableA	ftp server.
Config.DoorSetting.INPUT.FtpEnableA	0 - Disable 1 - Enable
Config.DoorSetting.INPUT.HttpEnableA	when input trig, send a http get requst
Config.DoorSetting.INPUT.HttpEnableB	with url
Config.DoorSetting.INPUT.HttpEnableC	0 - Disable 1 - Enable
Config.DoorSetting.INPUT.SmtpEnableA	when input trig, send capture files with
Config.DoorSetting.INPUT.SmtpEnableB	smtp
Config.DoorSetting.INPUT.SmtpEnableC	0 - Disable 1 - Enable
Config.DoorSetting.INPUT.SipEnableA	when input trig, start a sip call
Config.DoorSetting.INPUT.SipEnableB	
Config.DoorSetting.INPUT.SipEnableC	
Config.DoorSetting.Action.CallNum	sip call num
Config.DoorSetting.Action.CallName	sip call remote display name

Api input get

The GET or POST method can be used for this function.

Example Request:

```
GET: /api/input/get
```

```
POST:
{
    "target": "input",
    "action": "get"
}
```

Example Return:

```
{
    "retcode": 0,
    "action": "get",
    "message": "OK",
    "data": {
        "Config.DoorSetting.INPUT.EnableA": "1",
        "Config.DoorSetting.INPUT.EnableB": "1",
        "Config.DoorSetting.INPUT.EnableC": "1",
        "Config.DoorSetting.INPUT.OptionA": "0",
        "Config.DoorSetting.INPUT.OptionB": "0",
        "Config.DoorSetting.INPUT.OptionC": "0",
        "Config.DoorSetting.INPUT.DelayA": "0",
        "Config.DoorSetting.INPUT.DelayB": "0",
        "Config.DoorSetting.INPUT.DelayC": "0",
        "Config.DoorSetting.INPUT.RelayIdA": "1",
        "Config.DoorSetting.INPUT.RelayIdB": "2",
        "Config.DoorSetting.INPUT.RelayIdC": "3",
        "Config.DoorSetting.INPUT.FtpEnableA": "0",
        "Config.DoorSetting.INPUT.FtpEnableB": "0",
        "Config.DoorSetting.INPUT.FtpEnableC": "0",
        "Config.DoorSetting.INPUT.HttpEnableA": "0",
        "Config.DoorSetting.INPUT.HttpEnableB": "0",
        "Config.DoorSetting.INPUT.HttpEnableC": "0",
        "Config.DoorSetting.INPUT.SipEnableA": "0",
        "Config.DoorSetting.INPUT.SipEnableB": "0",
        "Config.DoorSetting.INPUT.SipEnableC": "0",
        "Config.DoorSetting.INPUT.SmtpEnableA": "0",
        "Config.DoorSetting.INPUT.SmtpEnableB": "0",
        "Config.DoorSetting.INPUT.SmtpEnableC": "0",
        "Config.DoorSetting.INPUT.ActionUrlA": "",
        "Config.DoorSetting.INPUT.ActionUrlB": "",
        "Config.DoorSetting.INPUT.ActionUrlC": ""
    }
}
```


Api input set

Only POST method can be used for this function.

Example Request:

```
POST:
{
      "target": "input",
      "action": "set",
      "data" : {
          "Config.DoorSetting.INPUT.EnableA": "0"
      }
}
```

Example Return:

```
{
      "retcode": 0,
      "action": "set",
      "message": "OK"
}
```

Api input status

Params	Description
InputA	Door A GPIO Level
InputB	Door B GPIO Level
InputC	Door C GPIO Level

The GET or POST method can be used for this function.

Example Request:

```
GET: /api/input/status
```

```
POST:
{
      "target": "input",
      "action": "status",
}
```

Example Return:

```
{
      "retcode": 0,
      "action": "status",
      "message": "OK",
      "data": {
            "InputA": 1,
            "InputB": 1,
            "InputC": 1
      }
}
```


Api rfkey

Param	Description
ID	e.g. “ID”:”1”
Code	RF SN e.g. “Code”:”12345678”
DoorNum	1 : Door A 2 : Door B 3 : Door C e.g. “DoorNum”: “123”
Mon	0 - Disable 1 - Enable
Tue	0 - Disable 1 - Enable
Wed	0 - Disable 1 - Enable
Thur	0 - Disable 1 - Enable
Fri	0 - Disable 1 - Enable
Sat	0 - Disable 1 - Enable
Sun	0 - Disable 1 - Enable
TimeStart	“00:00” - “23:59” “00:00” - “00:00”: key will always work
TimeEnd	“00:00” - “23:59”
Tags	0: ALLOWED 1: FREQUENCY

	2: FROBIDDEN **3: SCHEDULE**
Frequency	**0 ~ 65535**
DoorWebRelay	**(Reserved) for WebRelay Action**

Api rfkey get

The GET or POST method can be used for this function.

Example Request:

```
GET: /api/rfkey/get
```

```
POST:
{
    "target": "rfkey",
    "action": "get",
```

Example Return:

```
{
    "retcode": 0,
    "action": "get",
    "message": "OK",
    "data": {
        "num": 2,
        "item": [
            {
                "Code": "00B4161D",
                "DoorNum": "0",
                "Frequency": "0",
                "Fri": "1",
                "ID": "1",
                "Mon": "1",
                "Name": "Test",
                "Sat": "1",
                "Sun": "1",
                "Tags": "0",
                "Thur": "1",
                "TimeEnd": "00:00",
                "TimeStart": "00:00",
                "Tue": "1",
                "Wed": "1"
            },
            {
                "Code": "00B4161E",
                "DoorNum": "0",
                "Frequency": "0",
                "Fri": "1",
                "ID": "2",
                "Mon": "1",
                "Name": "Test2",
                "Sat": "1",
                "Sun": "1",
                "Tags": "0",
                "Thur": "1",
                "TimeEnd": "00:00",
                "TimeStart": "00:00",
                "Tue": "1",
                "Wed": "1"
            }
        ]
    }
}
```

Api rfkey add

Only POST method can be used for this function.

Example Request:

POST:

```
{
    "target": "rfkey",
    "action": "add",
    "data" : {
        "item": [
            {
                "ID": "0",
                "Name": "Test",
                "Code": "00B4161D",
                "DoorNum": "1",
                "Tags": "0",
                "Frequency": "0",
                "Mon": "1",
                "Tue": "1",
                "Wed": "1",
                "Thur": "1",
                "Fri": "1",
                "Sat": "1",
                "Sun": "1",
                "TimeEnd": "00:00",
                "TimeStart": "00:00"
            },
            {
                "ID": "0",
                "Name": "Test2",
                "Code": "00B4161E",
                "DoorNum": "1",
                "Tags": "0",
                "Frequency": "0",
                "Mon": "1",
                "Tue": "1",
                "Wed": "1",
                "Thur": "1",
                "Fri": "1",
                "Sat": "1",
                "Sun": "1",
                "TimeEnd": "00:00",
                "TimeStart": "00:00"
            }
        ]
    }
}
```

Example Return:


```
{
    "retcode": 0,
    "action": "add",
    "message": "OK",
    "data": {
        "num": 2,
        "item": [
            {
                "Ret": 1,
                "Code": "00B4161D",
                "ID": "1",
                "Name": "Test"
            },
            {
                "Ret": 2,
                "Code": "00B4161E",
                "ID": "2",
                "Name": "Test2"
            }
        ]
    }
}
```


Api rfkey set

Only POST method can be used for this function.

Example Request:

POST:

```
{
    "target": "rfkey",
    "action": "set",
    "data" : {
        "item": [
            {
                "ID": "1",
                "Name": "Test",
                "Code": "00B4161D",
                "DoorNum": "123",
                "Tags": "0",
                "Frequency": "0",
                "Mon": "1",
                "Tue": "1",
                "Wed": "1",
                "Thur": "1",
                "Fri": "1",
                "Sat": "1",
                "Sun": "1",
                "TimeEnd": "00:00",
                "TimeStart": "00:00"
            }
        ]
    }
}
```

Example Return:

```
{
	"retcode": 0,
	"action": "set",
	"message": "OK",
	"data": {
		"num": 1,
		"item": [
			{
				"Ret": 0,
				"Code": "00B4161D",
				"ID": "1",
				"Name": "Test"
			}
		]
	}
}
```


Api rfkey del

Only POST method can be used for this function.

Example Request:

```
POST:
{
      "target": "rfkey",
      "action": "del",
      "data" : {
            "item": [
                  {
                     "ID": "1"
                  },
                  {
                     "Name": "Test2"
                  },
                  {
                     "ID": "5",
                     "Code": "88888888",
                     "Name": "Test"
                  }
            ]
      }
}
```

Example Return:

```
{
    "retcode": 0,
    "action": "del",
    "message": "OK",
    "data": {
        "num": 3,
        "item": [
            {
                "Ret": 0,
                "Code": "00B4161D",
                "ID": "1",
                "Name": "Test"
            },
            {
                "Ret": 0,
                "Code": "00B4161E",
                "ID": "2",
                "Name": "Test2"
            },
            {
                "Ret": -1,
                "Code": "88888888",
                "ID": "5",
                "Name": "Test"
            }
        ]
    }
}
```


Api rfkey clear

The GET or POST method can be used for this function.

Example Request:

```
GET: /api/rfkey/clear
```

```
POST:
{
    "target": "rfkey",
   "action": "clear",
}
```

Example Return:

```
{
      "retcode": 0,
      "action": "clear",
      "message": "OK"
}
```

Api rfkey export

The GET method can be used for this function.

Example Request:

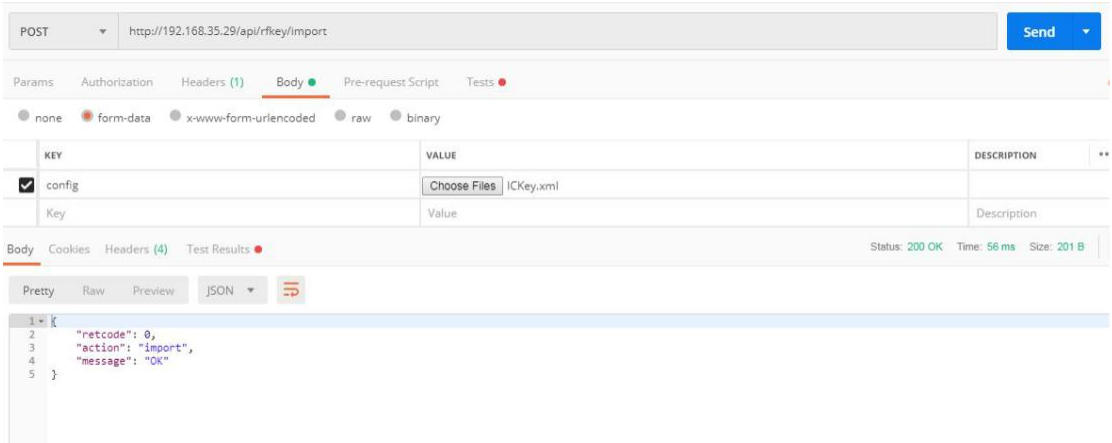
```
GET: /api/rfkey/export
```

Example Return:

```
Content-Type: application/octet-stream
Content-Disposition: attachment;filename=ICKey.xml
```

Api rfkey import

Only POST method can be used for this function.



Api call status

The GET or POST method can be used for this function.

Example Request:

```
GET: /api/call/status
```

```
POST:
{
    "target": "call",
    "action": "status"
}
```

Example Return:

```
{
    "retcode": 0,
    "action": "status",
    "message": "OK",
    "data": {
        "Status": "READY"
    }
}
```

Api call dial

The GET or POST method can be used for this function.

Example Request:

```
GET: /api/call/dial?Phone=192.168.1.58&DisplayName=Test3
```

```
POST:
{
    "target": "call",
    "action": "dial",
    "data" : {
        "Phone": "192.168.1.58",
        "DisplayName": "Test3"
    }
}
```

Example Return:

```
{
    "retcode": 0,
    "action": "dial",
    "message": "OK"
}
```

Api call hangup

The GET or POST method can be used for this function.

Example Request:

```
GET: /api/call/hangup
```

```
POST:
{
    "target": "call",
    "action": "hangup",
}
```

Example Return:

```
{
    "retcode": 0,
    "action": "hangup",
    "message": "OK"
}
```

Api sip

Param	Description
Account 1	
Config.Account1.GENERAL.Enable	0 Disable 1 Enable
Config.Account1.SIP.Server	SIP Server Ex:192.168.1.115
Config.Account1.SIP.Port	SIP Server Port Ex:5060
Config.Account1.GENERAL.AuthName	Sip Server Auth Name
Config.Account1.GENERAL.UserName	Sip Server Register Name
Config.Account1.GENERAL.Pwd	Sip Server Password
Config.Account1.GENERAL.DisplayName	Remote Display Name
Account 2	
Config.Account2.GENERAL.Enable	0 Disable 1 Enable
Config.Account2.SIP.Server	SIP Server Ex:192.168.1.115
Config.Account2.SIP.Port	SIP Server Port Ex:5060
Config.Account2.GENERAL.AuthName	Sip Server Auth Name
Config.Account2.GENERAL.UserName	Sip Server Register Name
Config.Account2.GENERAL.Pwd	Sip Server Password
Config.Account2.GENERAL.DisplayName	Remote Display Name

Api sip get

The GET or POST method can be used for this function.

Example Request:

```
GET: /api/sip/get
```

```
POST:
{
    "target": "sip",
    "action": "get"
}
```

Example Return:

```
{
    "retcode": 0,
    "action": "get",
    "message": "OK",
    "data": {
        "Config.Account1.ANONYMOUS_CALL.Enable": "0",
        "Config.Account1.ANONYMOUS_CALL.OffCode": "",
        "Config.Account1.ANONYMOUS_CALL.OnCode": "",
        "Config.Account1.ANONYMOUS_CALL.PrivacyHeaderValue": "id",
        "Config.Account1.AUTO_ANSWER.Enable": "1",
        "Config.Account1.GENERAL.AuthName": "",
        "Config.Account1.GENERAL.DisplayName": "",
        "Config.Account1.GENERAL.EditEnable": "",
        "Config.Account1.GENERAL.Enable": "0",
        "Config.Account1.GENERAL.Label": "",
        "Config.Account1.GENERAL.Pwd": "",
        "Config.Account1.GENERAL.UserAgent": "",
        "Config.Account1.GENERAL.UserName": "",
        "Config.Account1.NAT.Rport": "0",
        "Config.Account1.NAT.UdpKeepEnable": "1",
        "Config.Account1.NAT.UdpKeepInterval": "30",
        "Config.Account1.OUTPROXY.BakPort": "5060",
        "Config.Account1.OUTPROXY.BakServer": "",
        "Config.Account1.OUTPROXY.DHCPOption": "0",
        "Config.Account1.OUTPROXY.Enable": "0",
        "Config.Account1.OUTPROXY.Port": "5060",
        "Config.Account1.OUTPROXY.Server": "",
        "Config.Account1.REJECT_ANONYMOUSCALL.Enable": "0",
        "Config.Account1.REJECT_ANONYMOUSCALL.OffCode": "",
        "Config.Account1.REJECT_ANONYMOUSCALL.OnCode": "",
        "Config.Account1.SIP.ListenPortMax": "5062",
        "Config.Account1.SIP.ListenPortMin": "5062",
        "Config.Account1.SIP.Port": "5060",
        "Config.Account1.SIP.Port2": "5060",
        "Config.Account1.SIP.Server": "",
        "Config.Account1.SIP.Server2": "",
        "Config.Account1.SIP.TransType": "0",
        "Config.Account1.STUN.Enable": "0",
        "Config.Account1.STUN.Port": "3478",
        "Config.Account1.STUN.Server": "",
        "Config.Account2.ANONYMOUS_CALL.Enable": "0",
        "Config.Account2.ANONYMOUS_CALL.OffCode": "",
        "Config.Account2.ANONYMOUS_CALL.OnCode": "",
```

```
            "Config.Account2.ANONYMOUS_CALL.PrivacyHeaderValue": "id",
            "Config.Account2.AUTO_ANSWER.Enable": "1",
            "Config.Account2.GENERAL.AuthName": "",
            "Config.Account2.GENERAL.DisplayName": "",
            "Config.Account2.GENERAL.EditEnable": "",
            "Config.Account2.GENERAL.Enable": "0",
            "Config.Account2.GENERAL.Label": "",
            "Config.Account2.GENERAL.Pwd": "",
            "Config.Account2.GENERAL.UserAgent": "",
            "Config.Account2.GENERAL.UserName": "",
            "Config.Account2.NAT.Rport": "0",
            "Config.Account2.NAT.UdpKeepEnable": "1",
            "Config.Account2.NAT.UdpKeepInterval": "30",
            "Config.Account2.OUTPROXY.BakPort": "5060",
            "Config.Account2.OUTPROXY.BakServer": "",
            "Config.Account2.OUTPROXY.DHCPOption": "0",
            "Config.Account2.OUTPROXY.Enable": "0",
            "Config.Account2.OUTPROXY.Port": "5060",
            "Config.Account2.OUTPROXY.Server": "",
            "Config.Account2.REJECT_ANONYMOUSCALL.Enable": "0",
            "Config.Account2.REJECT_ANONYMOUSCALL.OffCode": "",
            "Config.Account2.REJECT_ANONYMOUSCALL.OnCode": "",
            "Config.Account2.SIP.ListenPortMax": "5063",
            "Config.Account2.SIP.ListenPortMin": "5063",
            "Config.Account2.SIP.Port": "5060",
            "Config.Account2.SIP.Port2": "5060",
            "Config.Account2.SIP.Server": "",
            "Config.Account2.SIP.Server2": "",
            "Config.Account2.SIP.TransType": "0",
            "Config.Account2.STUN.Enable": "0",
            "Config.Account2.STUN.Port": "3478",
            "Config.Account2.STUN.Server": ""
        }
}
```


Api sip set

Only POST method can be used for this function.

Example Request:

```
POST:
{
    "target": "sip",
    "action": "set",
    "data" : {
        "Config.Account1.GENERAL.AuthName" : "130",
        "Config.Account1.GENERAL.DisplayName": "130",
        "Config.Account1.GENERAL.Enable": "1",
        "Config.Account1.GENERAL.Label": "130",
        "Config.Account1.GENERAL.Pwd": "130",
        "Config.Account1.GENERAL.UserName":"130",
        "Config.Account1.SIP.Port":"5060",
        "Config.Account1.SIP.Server": "192.168.1.115"
    }
}
```

Example Return:

```
{
     "retcode": 0,
     "action": "set",
     "message": "OK"
}
```

Api sip status

The GET or POST method can be used for this function.

Example Request:

```
GET: /api/sip/status
```

```
POST:
{
    "target": "sip",
    "action": "status",
}
```

Example Return:

```
{
	"retcode": 0,
	"action": "status",
	"message": "OK",
	"data": {
		"Account1": {
			"UserName": "130",
			"SipServer": "192.168.1.115",
			"Status": "2"
		},
		"Account2": {
			"UserName": "",
			"SipServer": "",
			"Status": "0"
		}
	}
}
```


Api dialreplace

Param	Description
ID	ID
Prefix	number
Line	0 Auto 1 Account 1 2 Account 2
Replace1	number
Replace2	number
Replace3	number
Replace4	number
Replace5	number

Api dialreplace get

The GET or POST method can be used for this function.

Example Request:

```
GET: /api/dialreplace/get
```

```
POST:
{
    "target": "dialreplace",
    "action": "get"
}
```

Example Return:

```
{
    "retcode": 0,
    "action": "get",
    "message": "OK",
    "data": {
        "num": 1,
        "item": [
            {
                "ID": "1",
                "Line": "0",
                "Prefix": "110",
                "Replace1": "192.168.2.1",
                "Replace2": "192.168.2.4",
                "Replace3": "192.168.2.2",
                "Replace4": "192.168.2.5",
                "Replace5": "192.168.2.3"
            }
        ]
    }
}
```

Api dialreplace add

Only POST method can be used for this function.

Example Request:

```
POST:
{
    "target": "dialreplace",
    "action": "add",
    "data" : {
        "item": [
            {
                "ID": "1",
                "Line": "0",
                "Prefix" : "110",
                "Replace1": "192.168.2.1",
                "Replace2": "192.168.2.4",
                "Replace3": "192.168.2.2",
                "Replace4": "192.168.2.5",
                "Replace5": "192.168.2.3"
            }
        ]
    }
}
```

Example Return:

```
{
    "retcode": 0,
    "action": "add",
    "message": "OK",
    "data": {
        "num": 1,
        "item": [
            {
                "Ret": 0,
                "ID": "1",
                "Prefix": "110"
            }
        ]
    }
}
```

Api dialreplace set

Only POST method can be used for this function.

Example Request:

```
POST:
{
    "target": "dialreplace",
    "action": "set",
    "data" : {
        "item": [
            {
                "ID": "1",
                "Line": "0",
                "Prefix" : "110",
                "Replace1": "192.168.2.1",
                "Replace2": "192.168.2.4",
                "Replace3": "192.168.2.2",
                "Replace4": "192.168.2.5",
                "Replace5": "192.168.2.3"
            }
        ]
    }
}
```

Example Return:

```
{
    "retcode": 0,
    "action": "set",
    "message": "OK",
    "data": {
        "num": 1,
        "item": [
            {
                "Ret": 0,
                "ID": "1",
                "Prefix": "110"
            }
        ]
    }
}
```

Api dialreplace del

Only POST method can be used for this function.

Example Request:

```
POST:
{
    "target": "dialreplace",
    "action": "del",
    "data" : {
        "item": [
            {
                "ID": "1"
            }
        ]
    }
}
```

Example Return:

```
{
    "retcode": 0,
    "action": "del",
    "message": "OK",
    "data": {
        "num": 1,
        "item": [
            {
                "Ret": 0,
                "ID": "1"
            }
        ]
    }
}
```

Api dialreplace clear

Example Request:

```
GET: /api/dialreplace/clear
```

```
POST:
{
    "target": "dialreplace",
    "action": "clear",
}
```

Example Return:

```
{
      "retcode": 0,
      "action": "clear",
      "message": "OK"
}
```

Api dialreplace export

The GET method can be used for this function.

Example Request:

```
GET: /api/dialreplace/export
```

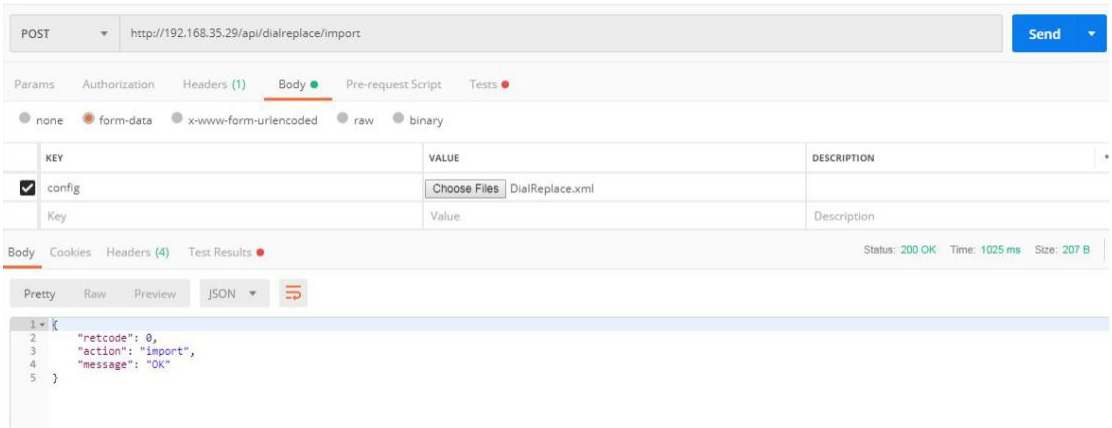
Example Return:

```
DialReplace.xml
```

Api dialreplace import

Only POST method can be used for this function.

Example Request&Return::



Api log

Param	Description
Config.Settings.LOGLEVEL.Level	log debug level 0 LOG_LEVEL_EMERG 1 LOG_LEVEL_ALERT 2 LOG_LEVEL_CRIT 3 LOG_LEVEL_ERR 4 LOG_LEVEL_WARNING 5 LOG_LEVEL_NOTICE 6 LOG_LEVEL_INFO 7 LOG_LEVEL_DEBUG

Api log get

The GET or POST method can be used for this function.

Example Request:

```
GET: /api/log/get
```

```
POST:
{
    "target": "log",
    "action": "get",
}
```

Example Return:

```
{
    "retcode": 0,
    "action": "get",
    "message": "OK",
    "data": {
        "Config.Settings.LOGLEVEL.Level": "3"
    }
}
```

Api log set

Only POST method can be used for this function.

Example Request:

```
POST:
{
    "target": "log",
    "action": "set",
    "data" : {
        "Config.Settings.LOGLEVEL.Level": "7"
    }
}
```

Example Return:

```
{
    "retcode": 0,
    "action": "get",
    "message": "OK",
    "data": {
        "Config.Settings.LOGLEVEL.Level": "7"
    }
}
```

Api log export

Param	Description
type	export log type log : default , application log(Android logcat or Linux syslog) kmsg: kernel log log_debug: support android only

The GET method can be used for this function.

Example Request:

```
GET: /api/log/export?type=log
```

Example Return:

```
log.txt
kmsg.txt
log_debug.tar.gz
```

Api log clear

Param	Description
type	export log type log : default , application log(Android logcat or Linux syslog) kmsg: kernel log log_debug: support android only

Example Request:

```
GET: /api/log/clear?type=kmsg
```

```
POST:
{
    "target": "log",
    "action": "clear",
    "data" : {
        "type" : "kmsg"
    }
}
```

Example Return:

```
{
    "retcode": 0,
    "action": "clear",
    "message": "OK"
}
```

Api calllog

Param	Description
ID	
Type	0 "All", 1 "Dialed", 2 "Received", 3 "Missed", 4 "Forwarded", 5 "Unknow",
Date	
Time	
Local Identity	Sip Server
Name	Remote DisplayName
Number	Call Num

Api calllog get

Param	Description
Type	Optional , Default “All” 0 "All", 1 "Dialed", 2 "Received", 3 "Missed", 4 "Forwarded", 5 "Unknow",

Example Request:

```
GET: /api/calllog/get
```

```
POST:
{
    "target": "calllog",
    "action": "get"
}
```

Example Return:

```
{
    "retcode": 0,
    "action": "get",
    "message": "OK",
    "data": {
        "num": 1,
        "item": [
            {
                "Date": "2018-12-13",
                "ID": "23",
                "LocalIdentity": "192.168.1.98@192.168.1.98",
                "Name": "192.168.1.48",
                "Num": "192.168.1.98@192.168.1.98",
                "Time": "05:55:59",
                "Type": "Received"
            }
        ]
    }
}
```


Api calllog del

Param	Description
ID	string, index

Only POST method can be used for this function.

Example Request:

```
POST:
{
    "target": "calllog",
    "action": "del",
    "data" : {
        "ID" : "23"
    }
}
```

Example Return:

```
{
      "retcode": 0,
      "action": "del",
      "message": "OK"
}
```

Api calllog clear

Param	Description
Type	string 0 "All", 1 "Dialed", 2 "Received", 3 "Missed", 4 "Forwarded", 5 "Unknow",

The GET or POST method can be used for this function.

Example Request:

```
GET: /api/calllog/clear?type=Missed
```

```
POST:
{
     "target": "calllog",
     "action": "clear",
     "data" : {
          "Type" : "Missed"
     }
}
```

Example Return:

```
{
      "retcode": 0,
      "action": "upgrade",
      "message": "OK"
}
```

Api calllog export

The GET method can be used for this function.

Example Request:

GET: /api/calllog/export

Example Return:

History.csv

Api doorlog

Param	Description
ID	
Type	0 “Unknow”, 1 "Card", 2 "Password", 3 "Public", 4 "Face",
Name	
Status	0 Success 1 Failed
Code	
Date	
Time	

Api doorlog get

The GET or POST method can be used for this function.

Example Request:

GET: /api/fe/upgrade

```
POST:
{
    "target": "doorlog",
    "action": "get"
}
```

Example Return:

```
{
     "retcode": 0,
     "action": "upgrade",
     "message": "OK"
}
```

Api doorlog del

Param	Description
ID	string, index

Only POST method can be used for this function.

Example Request:

```
POST:
{
    "target": "calllog",
    "action": "del",
    "data" : {
        "ID" : "1"
    }
}
```

Example Return:

```
{
     "retcode": 0,
     "action": "del",
     "message": "OK"
}
```

Api doorlog clear

The GET or POST method can be used for this function.

Example Request:

```
GET: /api/doorlog/clear
```

```
POST:
{
    "target": "doorlog",
    "action": "clear",
}
```

Example Return:

```
{
    "retcode": 0,
    "action": "clear",
    "message": "OK"
}
```

Api doorlog export

The GET method can be used for this function.

Example Request:

```
GET: /api/calllog/export
```

Example Return:

```
DoorLog.xml
```

Api pcap status

The GET or POST method can be used for this function.

Example Request:

```
GET: /api/pcap/status
```

```
POST:
{
    "target": "pcap",
    "action": "status"
}
```

Example Return:

```
{
    "retcode": 0,
    "action": "status",
    "message": "OK",
    "data": {
        "FileSize": 0,
        "Status": "stop"
    }
}
```

Api pcap start

Param	Description
type	capature type, optional auto auto stop (capture 60 second) manual
filter	pcap filter type, optional more info see libpcap default: “ip”
time	capture time 1-180 , optional default 60 second only work on type == auto
size	optional capture limit size, when reach size limit it will auto stop default: 1048576

The GET or POST method can be used for this function.

Example Request:

```
GET: /api//api/pcap/start
```

```
POST:
{
    "target": "pcap",
    "action": "start",
    "data" : {
        "type" : "auto",
        "filter" : "ip",
        "time": "180",
        "size": "1048576"
    }
}
```

Example Return:

```
{
     "retcode": 0,
     "action": "start",
     "message": "OK",
     "data": {
          "filter": "ip",
          "interface": "eth0",
          "size": "1048576",
          "time": "180",
          "type": "auto"
     }
}
```

Api pcap stop

The GET or POST method can be used for this function.

Example Request:

```
GET: /api/pcap/stop
```

```
POST:
{
    "target": "pcap",
    "action": "stop"
}
```

Example Return:

```
{
	"retcode": 0,
	"action": "stop",
	"message": "OK",
	"data": {
		"FileSize": "314407",
		"Time": "9"
	}
}
```


Api pcap export

The GET method can be used for this function.

Example Request:

```
GET: /api/pcap/export
```

Example Return:

```
phone.pcap
```