



# HTTP API manual for Akuvox Intercom(Android)

Time	Version	Changes
2023.05.30	V1.0	Interface organization

# Content

Introduction .....	5
HTTP API Functions Description .....	1
HTTP API Configuration .....	4
HTTP API Authentication Method .....	6
None .....	6
WHITE LIST .....	6
Basic .....	6
Digest .....	6
HTTP API Request .....	7
HTTP API Method .....	8
api system .....	10
api system info .....	10
api system status .....	14
api system reboot .....	14
api firmware .....	16
api firmware status .....	16
api firmware prepare .....	16
api firmware upload .....	17
api firmware upgrade .....	19
api config .....	21
api config get .....	21
api config set .....	24
api config export .....	24
api config import .....	25
api config reset_factory .....	26
api config reset_factory_except_data .....	26
api relay .....	28
api relay get .....	29
api relay set .....	29
api relay status .....	30
api relay trig .....	31
api input .....	33
api input get .....	34
api input set .....	35
api input status .....	36
api call .....	37
api call status .....	38
api call dial .....	38
api call hangup .....	39
api sip .....	40
api sip get .....	40
api sip set .....	41

api sip status .....	43
api dialreplace .....	44
api dialreplace get .....	45
api dialreplace add .....	48
api dialreplace set .....	50
api dialreplace del .....	52
api dialreplace clear .....	53
api dialreplace import .....	53
api dialreplace export .....	54
api log .....	55
api log get .....	56
api log set .....	57
api log export .....	58
api log clear .....	58
api calllog .....	60
api calllog get .....	60
api calllog del .....	62
api calllog clear .....	63
api calllog export .....	64
api doorlog .....	66
api doorlog get .....	66
api doorlog del .....	67
api doorlog clear .....	68
api doorlog export .....	68
api pcap .....	70
api pcap status .....	70
api pcap start .....	71
api pcap stop .....	73
api pcap export .....	73
api contact .....	75
api contact get .....	75
api contact add .....	78
api contact set .....	80
api contact del .....	82
api contact clear .....	83
api contact export .....	84
api contact import .....	85
api group .....	87
api group get .....	87
api group add .....	90
api group set .....	91
api group clear .....	91
api time .....	93
api time get .....	93

api time set .....	94
api rfkey .....	96
api rfkey get .....	96
api rfkey add .....	99
api rfkey set .....	101
api rfkey del .....	102
api rfkey clear .....	103
api schedule .....	105
api schedule get .....	106
api schedule add .....	109
api schedule set .....	113
api schedule del .....	115
api schedule clear .....	116
api schedule export .....	117
api schedule import .....	117
api user .....	118
api user get .....	118
api user info .....	126
api user add .....	126
api user set .....	137
api user del .....	137
api user clear .....	139
api user export .....	139
api user import .....	141
api user status .....	142
api commontool .....	146
api commontool get .....	146
api commontool set .....	148
api filetool .....	150
api filetool import .....	150
api filetool del .....	151

# Introduction

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

**Note:**

1. If the request returns 401 or other authentication related values, check in **HTTP API Configuration**
2. View descriptions in the table to find request you want. **HTTP API Functions Description**

# HTTP API Functions Description

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

- the HTTP request absolute path
- the supported HTTP methods

<b>Absolute path</b>	<b>API Method</b>	<b>Description</b>
/api/system/info	GET/POST	Obtain basic status information of the device
/api/system/status	GET/POST	Obtain Unix time and online seconds
/api/system/reboot	GET/POST	Reboot the device
/api/firmware/status	GET/POST	Obtain firmware information
/api/firmware/prepare	GET/POST	Prepare for software upgrade
/api/firmware/upload	POST	Upload upgrade file
/api/firmware/upgrade	GET/POST	Start upgrade
/api/config/get	GET/POST	Obtain device configuration
/api/config/set	POST	Specify a configuration for the device
/api/config/export	GET/POST	Export .tgz file of configuration
/api/config/import	POST	Import configuration
/api/config/reset_factory	GET/POST	Reset to factory and clean data
/api/config/reset_factory_except_data	GET/POST	Reset configuration to factory but keep data
/api/relay/get	GET/POST	Obtain some configurations about relay
/api/relay/set	POST	Set some specified configurations about relay
/api/relay/status	GET/POST	Obtain the current status of relay
/api/relay/trig	GET/POST	Trig relay over api
/api/input/get	GET/POST	Obtain some configurations about input
/api/input/set	POST	Set some specified configurations about input
/api/input/status	GET/POST	Obtain the current status of input
/api/call/status	GET/POST	Obtain device's status
/api/call/dial	GET/POST	Make call by request
/api/call/hangup	GET/POST	Hangup current call
/api/sip/get	GET/POST	Obtain configurations about sip
/api/sip/set	POST	Set configurations about sip
/api/sip/status	GET/POST	Obtain sip account registration

		status and information	
/api/dialreplace/get	GET/POST	Obtain dialreplace(dial plan)	
/api/dialreplace/add	POST	Add dialreplace(dial plan)	
/api/dialreplace/set	POST	Modify dialreplace(dial plan)	
/api/dialreplace/del	POST	Delete dialreplace(dial plan)	
/api/dialreplace/clear	GET/POST	Clear dialreplace(dial plan)	
/api/dialreplace/export	GET/POST	Export dialreplace(dial plan) to file	
/api/dialreplace/import	GET/POST	Import dialreplace(dial plan)	
/api/log/get	GET/POST	Obtain log level of device	
/api/log/set	POST	Set log level of device	
/api/log/export	GET/POST	Export log, kmsg, log_debug	
/api/log/clear	GET/POST	Clear log, kmsg, debug log	
/api/calllog/get	GET/POST	Obtain call log	
/api/calllog/del	GET/POST	Del call log by id	
/api/calllog/clear	GET/POST	Clear call log	
/api/calllog/export	GET/POST	Export call log	
/api/doorlog/get	GET/POST	Obtain door log	
/api/doorlog/del	GET/POST	Del door log	
/api/doorlog/clear	GET/POST	Clear door log	
/api/doorlog/export	GET/POST	Export door log	
/api/pcap/status	GET/POST	Check if the device is capturing packets now	
/api/pcap/start	GET/POST	Start capture packets	
/api/pcap/stop	GET/POST	Stop capture packets	
/api/pcap/export	GET/POST	Get captured packets	
/api/rfkey/get	GET/POST	Obtain rfkey in device	<b>R29/X916</b> <b>Abandoned</b>
/api/rfkey/add	POST	Add rfkey to device	
/api/rfkey/set	POST	Modify rfkey in device	
/api/rfkey/del	POST	Del rfkey in device	
/api/rfkey/clear	GET/POST	Clear rfkey	
/api/rfkey/export	GET/POST	Export rfkey to file	
/api/rfkey/import	POST	Import rfkey in file	
/api/contact/get	GET/POST	Get local contacts info	<b>Abandoned in S539</b>
/api/contact/add	POST	Add local contact	
/api/contact/set	POST	Modify local contact	
/api/contact/del	POST	Delete local contact by id	
/api/contact/clear	GET/POST	Clear local contact	
/api/contact/export	GET/POST	Export local contact to file	
/api/contact/status /api/contact/import	POST	Import contact from file	

/api/group/get	GET/POST	Get local group list
/api/group/add	POST	App local group
/api/group/set	POST	Modify local group
/api/group/clear	GET/POST	Clear local group
api/time/get	GET/POST	Get timezone list supported
api/time/set	GET/POST	Set timezone and other time configurations about time.
api/rfkey/get	GET/POST	Get super card in device
api/rfkey/add	POST	Add super card
api/rfkey/set	POST	Modify super card
api/rfkey/del	GET/POST	Delete super card by ID
api/rfkey/clear	GET/POST	Clear super card
api/schedule/get	GET/POST	Get schedule in device
api/schedule/add	POST	Add local schedule
api/schedule/set	POST	Modify local schedule
api/schedule/del	POST	Delete local schedule by ID
api/schedule/clear	GET/POST	Clear local schedule
api/schedule/export	GET/POST	Export local schedule
api/schedule/import	POST	Import schedule in XML file
api/user/get	GET/POST	Get user in device
api/user/info	GET/POST	Get the authorization mode of pin and recommend id before add and set
api/user/add	POST	Add local user
api/user/set	POST	Modify local user
api/user/del	POST	Delete local user by ID
api/user/clear	GET/POST	Clear local user
api/user/export	GET/POST	Export local user
api/user/import	POST	Import user in .TGZ file
api/user/status	GET/POST	Check device status, is importing user data?is export user data? Get swiping card.
api/commontool/get	GET/POST	Get some configurations that could not get by /api/config/get
api/commontool/set	POST	Some test interface. Modify some configurations that could not set by /api/config/set
api/filetool/import	POST	
api/filetool/del	GET/POST	
api/filetool/clear	GET/POST	



api/filetool/status	GET/POST	
---------------------	----------	--

# HTTP API Configuration

HTTP API configurations for Akuvox intercom are defined via web interface  
PATH:

R29\X916 Intercom->HTTP API

X915\S539 Aetting->HTTP API

to disable/enable a service and select the user authentication method

**HTTP API**

HTTP API	<input type="text" value="Enabled"/>	Auth Mode	<input type="text" value="Digest"/>
User Name	<input type="text" value="admin"/>	Password	<input type="password" value="....."/>
IP01	<input type="text"/>	IP02	<input type="text"/>
IP03	<input type="text"/>	IP04	<input type="text"/>
IP05	<input type="text"/>		

**HTTP API**

Enabled	<input checked="" type="checkbox"/>	Authorization Mode	<input type="text" value="Allowlist"/>
Username	<input type="text" value="admin"/>	Password	<input type="password" value="....."/>
1st IP	<input type="text"/>	2nd IP	<input type="text"/>
3rd IP	<input type="text"/>	4th IP	<input type="text"/>
5th IP	<input type="text"/>		

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

# HTTP API Description

Akuvox 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.

## HTTP API Method

Akuvox 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.

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

http://IP address/api/Target/Action?Param1=xxx

http://IP address/api/Target/Action?Param1=xxx&Param2=xxx

Example Request:

<http://192.168.1.163/api/config/get>

Example Return:

```
{
  "retcode": 0,
  "action": "get",
  "message": "OK",
  "data": {
    "Config.Autoprovision.GENERAL.Url": "",
    ...
  }
}
```

### 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

http://IP address/api/Target/Action with params in body

Example Request:


**POST** ▼ http://192.168.1.144/api/config/set

Params   Authorization   Headers (8)   **Body** ●   Pre-request Script   Tests   Set

none    form-data    x-www-form-urlencoded    raw    binary    GraphQL

```
1  .....
2  ..... "target": "config",
3  ..... "action": "set", .....
4  ..... "data": {
5  .....     "Config.DoorSetting.LOGACTIVE.DoorLogActive": "1"
6  ..... }
7  .....
```

**Body**   Cookies   Headers (5)   Test Results

**Pretty**   Raw   Preview   Visualize   JSON ▼   

```
1  .....
2  ..... "retcode": 0,
3  ..... "action": "set",
4  ..... "message": "OK"
5  .....
```

## api system

### api system info

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

Section	Return Param	Description
Status	Model	ProductModel
	MAC	MAC address
	FirmwareVersion	Firmware version
	HardwareVersion	Hardware version
	WebLang	Web language id
	RemoteDebugStatus	Remote debug serve status 0: DisConnected 1: Connected
	Uptime	how long the system has been running Up 0 Weeks, 2 Days, 21 Hours, 10 Minutes
TelnetActive	Ssh server enable 0: disable 1: enable	
Location	Device location set by Cloud server、SDMC	
Lan	PortType	0: DHCP 1: Static      2: Static
	LinkStatus	0: Disconnect 1: Connected 2: IP Conflict
	IPAddress	Ip address
	SubnetMask	Subnet Mask
	Gateway	Gateway
	DNS1	Preferred DNS Server
	DNS2	Alternate DNS Server
Account1 Accounr2	UserName	Sip account name
	SipServer	Sip server address
	Status	-1: unknow 0: disable 1: registering 2: registered 3: registe failed

DoorSetting	LocationCommunity	Community set by cloud server
	LocationStreet	Street set by cloud server

This request does not require parameters

GET http://192.168.1.144/api/system/info

Params Authorization Headers (6) **Body** Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary GraphQL

Body Cookies Headers (5) Test Results

Pretty Raw Preview Visualize JSON

```

1  {
2    "retcode": 0,
3    "action": "info",
4    "message": "OK",
5    "data": {
6      "Status": {
17     },
18     "Lan": {
28     },
29     "Account1": {
33     },
34     "Account2": {
38     },
39     "DoorSetting": {
43     }
44   }
45 }

```



POST http://192.168.1.144/api/system/info

Params Authorization Headers (8) **Body** Pre-request Script

none form-data x-www-form-urlencoded **raw** binary

```
1 [
2   "target": "system",
3   "action": "info"
4 ]
```

Body Cookies Headers (5) Test Results

Pretty Raw Preview Visualize JSON

```
1 {
2   "retcode": 0,
3   "action": "info",
4   "message": "OK",
5   "data": {
6     "Status": {
7       "code": 0,
8       "message": "OK",
9       "type": "success"
10    },
11    "Lan": {
12      "code": 0,
13      "message": "OK",
14      "type": "success"
15    },
16    "Account1": {
17      "code": 0,
18      "message": "OK",
19      "type": "success"
20    },
21    "Account2": {
22      "code": 0,
23      "message": "OK",
24      "type": "success"
25    },
26    "DoorSetting": {
27      "code": 0,
28      "message": "OK",
29      "type": "success"
30    }
31  }
32 }
```

Example Return:

```

{
  "retcode": 0,
  "action": "info",
  "message": "OK",
  "data": {
    "Status": {
      "Model": "S539",
      "MAC": "A6:22:03:23:05:08",
      "FirmwareVersion": "539.30.10.1",
      "HardwareVersion": "539.1.0.0",
      "WebLang": 0,
      "RemoteDebugStatus": 0,
      "Uptime": "up 0 weeks, 2 days, 20 hours, 29 minutes",
      "ProductModeActive": 0,
      "TelnetActive": 0,
      "Location": "529-10"
    },
    "Lan": {
      "PortType": "2",
      "LinkStatus": "1",
      "IPAddress": "192.168.1.144",
      "SubnetMask": "255.255.255.0",
      "Gateway": "192.168.1.1",
      "DNS1": "192.168.1.1",
      "DNS2": "192.168.1.1",
      "DHCP": "1",
      "StaticIP": "10.0.0.10"
    },
    "Account1": {
      "UserName": "5115106660",
      "SipServer": "pbx.test.akuvox.com",
      "Status": "2"
    },
    "Account2": {
      "UserName": "",
      "SipServer": "",
      "Status": "-1"
    },
    "DoorSetting": {
      "LocationCommunity": "denynew",
      "LocationStreet": "gghghjk",
      "ImportWebLang": 0
    }
  }
}

```

## api system status

The GET or POST method can be used for this function. This request does not require parameters and returns information as follow

Return 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

### 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": "R29S"
  }
}
```

### 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.

Request Param	Description
upgradeType	Whether to restore after upgrade. 1: yes

0 or not set: no
------------------

Example Request:

GET: /api/firmware/prepare or /api/firmware/prepare?upgradeType=0
---

POST: { "target": "firmware", "action": "prepare" }	POST: { "target": "firmware", "action": "prepare", "data": { "upgradeType":0 } }
---	---

Example Return:

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

## api firmware upload

Upload firmware, Only POST method can be used for this function.

Request Param	Description
silent	Whether to show upgrading UI when uploading file. 1: no 0 or not set: yes

PATH: /api/firmware/upload

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

WITHOUT PARAM: /api/firmware/upload

http://192.168.1.78/ | http://192.168.1.78/ | + | ... |

POST | http://192.168.1.78/api/firmware/upload

Authorization | Headers (1) | **Body** | Pre-request Script | Tests

form-data |  x-www-form-urlencoded |  raw |  binary

Key	Value
<input checked="" type="checkbox"/>	<input type="text" value="选择文件"/> test.zip
New key	Value

Body | Cookies | Headers (4) | Test Results

Pretty | Raw | Preview | JSON |

```
1 {
2   "retcode": 0,
3   "action": "upload",
4   "message": "OK"
5 }
```

## api firmware upgrade

Request Param	Description
md5	Whether to check file md5. If you want to make sure that file has been uploaded successfully, request with the md5 value of the file. Not empty: yes not set or empty: no

Example Request(without MD5 param):

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

POST:

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

Example Return(with correct MD5 param):

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

Example Request(with wrong 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

### api config get

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

#### To get all configurations

Example Request:

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

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

Example Return:

```
{
  "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",
  },
}
```

#### To get the configurations for one section

Example Request:

```

POST:
{
  "target": "config",
  "action": "get",
  "data": {
    "filter": [
      "Config.DoorSetting.CLOUDSERVER"
    ]
  }
}

```

Example Return:

```

{
  "retcode": 0
  "action": "get",
  "message": "OK",
  "data": {
    "Config.DoorSetting.CLOUDSERVER.DeviceCommunity": "",
    "Config.DoorSetting.CLOUDSERVER.Enable": "0",
    "Config.DoorSetting.CLOUDSERVER.GatePort": "0",
    "Config.DoorSetting.CLOUDSERVER.GateServer": "",
    "Config.DoorSetting.CLOUDSERVER.OpenDoorLimit": "",
    "Config.DoorSetting.CLOUDSERVER.Port": "",
    "Config.DoorSetting.CLOUDSERVER.RpsEnable": "1",
    "Config.DoorSetting.CLOUDSERVER.RpsServer":
"http://rps.akuvox.com:8080/redirect?mac=",
    "Config.DoorSetting.CLOUDSERVER.Server": "",
    "Config.DoorSetting.CLOUDSERVER.Token": "",
    "Config.DoorSetting.CLOUDSERVER.Type": "",
    "Config.DoorSetting.CLOUDSERVER.UploadPassword": "",
    "Config.DoorSetting.CLOUDSERVER.UploadUrl": "",
    "Config.DoorSetting.CLOUDSERVER.UploadUser": ""
  },
  ,
}

```

## To get the specified configuration

Example Request:

POST:

```
{
  "action": "get",
  "target": "config",
  "data": {
    "item": [
      "Config.DoorSetting.CLOUDSERVER.Enable",
      "Config.DoorSetting.CLOUDSERVER.RpsServer"
    ]
  }
}
```

Example Return:

```
{
  "retcode": 0
  "action": "get",
  "message": "OK",
  "data": {
    "Config.DoorSetting.CLOUDSERVER.Enable": "0",
    "Config.DoorSetting.CLOUDSERVER.RpsServer":
    "http://rps.akuvox.com:8080/redirect?mac="
  },
}
```

## api config set

Only POST method can be used for this function.

Example Request:

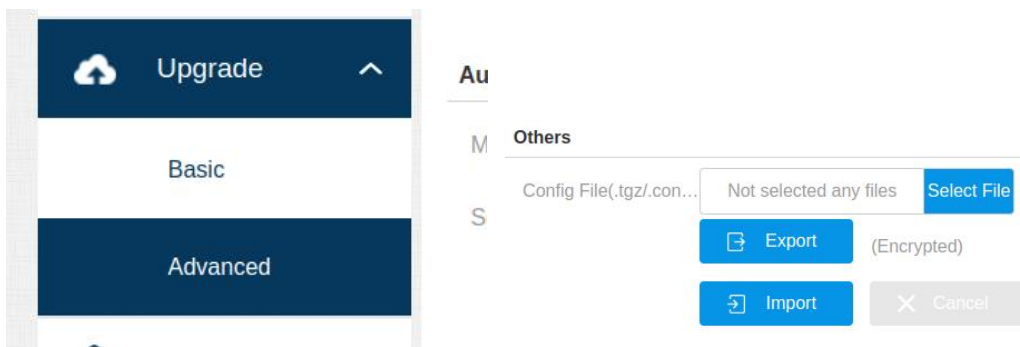
```
POST:
{
  "target": "config",
  "action": "set",
  "data": {
    "Config.DoorSetting.CLOUDSERVER.Enable": "0",
    "Config.DoorSetting.CLOUDSERVER.RpsServer": "192.168.1.163"
  }
}
```

Example Return:

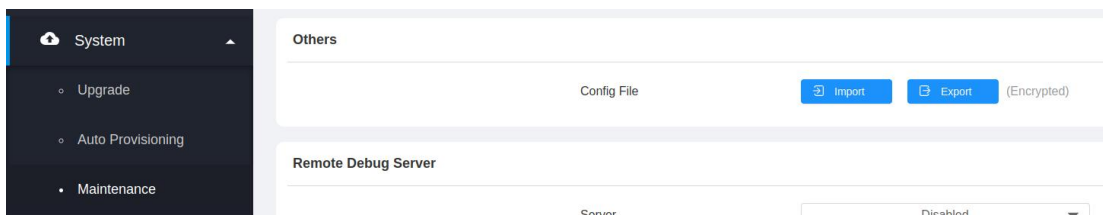
```
{
  "retcode": 0,
  "action": "set",
  "message": "OK"
}
```

## api config export

To export the configuration, equivalent to operate via web interface path:  
R29X916 Upgrader- > Advanced-> Config File->Export



X915S539 System->Maintenance->Others->Config File->Export



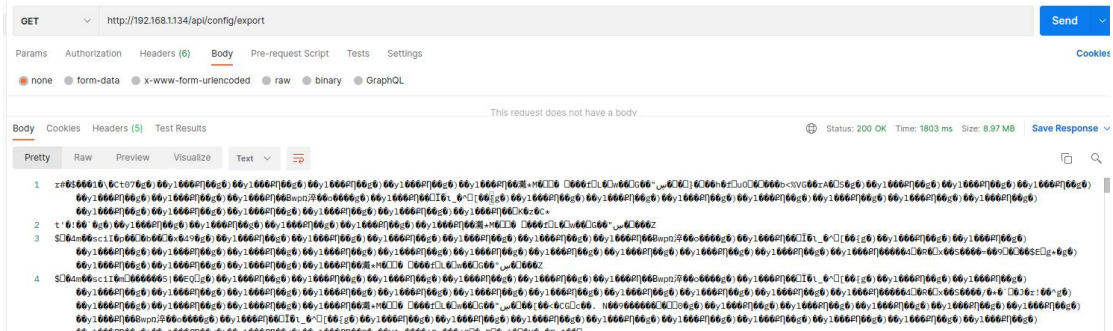
,and the exported configuration is encrypted which cannot be modified.

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

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

Example Return:

Old version will return Config.tgz file type



New version return the url to download the file.

```
{
  "retcode": 0,
  "action": "export",
  "message": "OK",
  "data": {
    "url": "http://192.168.1.52/download/config.tgz"
  }
}
```

## api config import

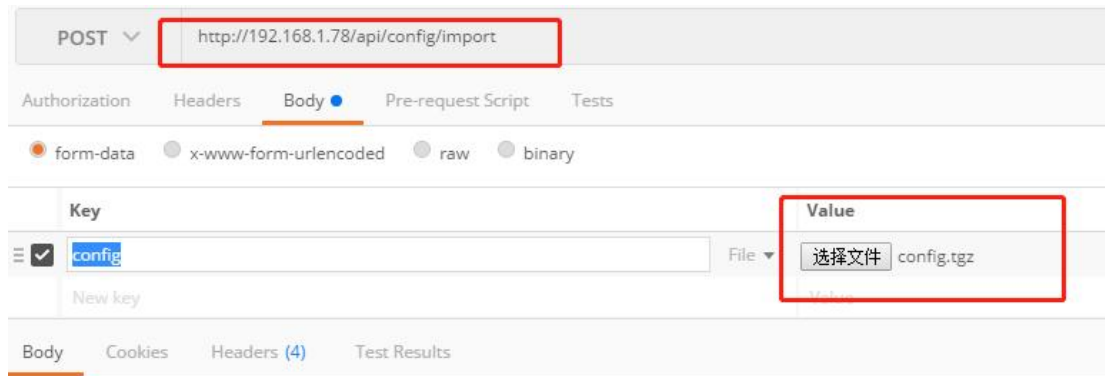
To import the configuration, equivalent to operate via web interface path:

R29X916 Upgrader-> Advanced-> Config File->Import

S539X915 System->Maintenance->Others->Config File->Import

Note: Device will reboot after import the configuration successfully.

Only POST method can be used for this function. only tgz file can be accepted.



## api config reset\_factory

Example Request:

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

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

Example Return:

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

## api config reset\_factory\_except\_data

Example Request:

```
GET: /api/config/reset_factory_except_data
```

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

Example Return:

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



## api relay

Through get and set, configurations as follow could be modified.

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	1: enable 0: disable
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	
Config.DoorSetting.RELAY.HttpTrig	Enable open door over http request
Config.DoorSetting.RELAY.UserName	Username for authentication
Config.DoorSetting.RELAY.Password	Due to safety reasons, we will receive a null value.Set with plaintext values and reboot to take effect.Or set it on devices website.

## api relay get

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

This request does not require parameters.

This request is designed to obtain a part of configurations about relay, it could be replaced by [api/config/get](#).

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",
```

## 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.  
This request does not require parameters.

Return Params	Description
RelayA	1: OPEN 0: CLOSE
RelayB	
RelayC	

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 trig**

The parameters required for this request are as follow

Request Params	Description
mode	0:Auto Close 1: Manual
num	1 Relay A 2 Relay B 3 Relay C 11 Security relay A 12 Security relay B
level	0 NO - COM 1 NC - COM
delay	0~65535(Second)

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 ftp server. 0 - Disable 1 - Enable
Config.DoorSetting.INPUT.FtpEnableA	
Config.DoorSetting.INPUT.FtpEnableA	
Config.DoorSetting.INPUT.HttpEnableA	when input trig, send a http get request with url 0 - Disable 1 - Enable
Config.DoorSetting.INPUT.HttpEnableB	
Config.DoorSetting.INPUT.HttpEnableC	
Config.DoorSetting.INPUT.SmtpEnableA	when input trig, send capture files with smtp 0 - Disable 1 - Enable
Config.DoorSetting.INPUT.SmtpEnableB	
Config.DoorSetting.INPUT.SmtpEnableC	
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.  
This could be replaced with `api/config/get`

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

Return 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 call

### api call status

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

This request does not require parameters and returns information as follow

Retuen Param	Description
Status	Device status NONE: not ready for make call Ready: ready for make call Talking: device is in a call ATUOP_CONFIG_UPGRADING: updating configuration over autop AUTOP_UPGRADING: upgrading firmware over autop WEB_UPGRADING: upgrading over web site UPGRADE_FAILED: upgrade failed now

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.

Request Param	Description
---------------	-------------

Phone	Make call to this number
DisplayName	Display name for this number

Example Request:

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

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

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
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. It returns for security reason.
Config.Account1.GENERAL.DisplayName	Remote Display Name
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",
    ...
  }
}
```

## api sip set

Only POST method can be used for this function.

Example Request:

```
POST:
{
  "target": "sip",
  "action": "set",
  "data" : {
    "Config.Account1.GENERAL.UserName": "130",
  }
}
```

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

Set prefix to 123 and set replace1 to 192.168.1.2, then it will make call to 192.168.1.2 when we make call to 123.

## api dialreplace get

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

Request Param	Description
page	Obtain the value corresponding to the specified number of pages, with 10 data per page. If it is not supported, the request would return all dialreplaces in device. 0 and 1 means first 10 dialreplaces. <u>This parameter is only supported in X915, S539</u>

Return Param	Description
curPageNum	Number of this page
num	Total count
item	data

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"
      }
    ]
  }
}

```

In S539, we got return message like this

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

- Method:** GET
- URL:** http://192.168.1.124/api/dialreplace/get?page=2
- Params:** none
- Body:** This request does not have a body
- Status:** 200 OK, 15 ms, 276 B
- Response Body (Pretty):**

```

1  {
2    "retcode": 0,
3    "action": "get",
4    "message": "OK",
5    "data": {
6      "curPageNum": 0,
7      "num": 1,
8      "item": []
9    }
10 }

```

POST http://192.168.1.124/api/dialreplace/get?page=1

Params Authorization Headers (8) **Body** Pre-request Script

● none ● form-data ● x-www-form-urlencoded ● **raw** ● binary ● C

```
1
2   "target": "dialreplace",
3   "action": "get",
4   "data" : {
5     "page": "1"
6   }
7
```

**Body** Cookies Headers (5) Test Results

Pretty Raw Preview Visualize JSON

```
1
2   "retcode": 0,
3   "action": "get",
4   "message": "OK",
5   "data": {
6     "curPageNum": 1,
7     "num": 1,
8     "item": [
9       {
10        "ID": "1",
11        "Line": "0",
12        "Prefix": "110",
13        "Replace1": "192.168.2.1",
14        "Replace2": "192.168.2.4",
15        "Replace3": "192.168.2.2",
16        "Replace4": "192.168.2.5",
17        "Replace5": "192.168.2.3"
18      }
19    ]
20  }
21
```

## api dialreplace add

Only POST method can be used for this function.  
ID is not necessary, it means nothing event set.  
We can add several replaces at once like this

```
POST:
{
  "target": "dialreplace",
  "action": "add",
  "data" : {
    "item": [
      {
        "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"
      },{
        "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 Request:

```

{
  "retcode": 0,
  "action": "add",
  "message": "OK",
  "data": {
    "num": 2,
    "item": [
      {
        "Ret": 0,
        "ID": "-1",
        "Prefix": "110"
      },
      {
        "Ret": 7,
        "ID": "-1",
        "Prefix": "110"
      }
    ]
  }
}

```

It will return the results, and Ret means

Return code	RET	Description
-1 or 27		Failed to operate file or database. Or data adding is empty.
10		Reach the max limit, no more dial plan could be added.
7		The same Prefix exists
0		OK

## api dialreplace set

Only POST method can be used for this function.

We can modify several replaces at once like this

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"
      }, {
        "ID": "3",
        "Line": "1",
        "Prefix" : "111",
        "Replace1": "192.168.2.7",
        "Replace2": "192.168.2.8",
        "Replace3": "192.168.2.9",
        "Replace4": "192.168.2.10",
        "Replace5": "192.168.2.11"
      }
    ]
  }
}
```

Example Return:

```

{
  "retcode": 0,
  "action": "set",
  "message": "OK",
  "data": {
    "num": 2,
    "item": [
      {
        "Ret": 0,
        "ID": "1",
        "Prefix": "110"
      },
      {
        "Ret": 4,
        "ID": "3",
        "Prefix": "111"
      }
    ]
  }
}

```

It will return the results, and Ret means

Return code	RET	Description
0		OK
-1		Failed to operate file or database. Or parameter required is empty.
7		The same Prefix exists
4		Failed to found dialplace which id is given in the request.



## api dialreplace del

Only POST method can be used for this function.

The ID can be found based on the requested return value(api/dialreplace/get)

Example Request:

```
POST:
{
  "target": "dialreplace",
  "action": "del",
  "data": {
    "item": [
      {
        "ID": "1"
      },{
        "ID": "4"
      }
    ]
  }
}
```

Example Return:

```
{
  "retcode": 0,
  "action": "del",
  "message": "OK",
  "data": {
    "num": 2,
    "item": [
      {
        "Ret": 0,
        "ID": "1"
      },
      {
        "Ret": 0,
        "ID": "5"
      }
    ]
  }
}
```

## api dialreplace clear

Example Request:

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

POST:

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

Example Return:

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

## api dialreplace import

Only POST method can be used for this function.Example Request&Return::

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

- Method: POST
- URL: http://192.168.35.29/api/dialreplace/import
- Body type: form-data
- Body content: A table with one row: 

KEY	VALUE
<input checked="" type="checkbox"/> config	Choose Files DialReplace.xml
- Response body (JSON):

```
1 {
2   "retcode": 0,
3   "action": "import",
4   "message": "OK"
5 }
```

## api dialreplace export

Example Request:

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

POST:

```
{
  "target": "dialreplace",
  "action": "export",
}
```

Example Return: Different models have different return values

GET http://192.168.1.113/api/dialreplace/export

Params Authorization Headers (6) **Body** Pre-request Script Tests Settings

none  form-data  x-www-form-urlencoded  raw  binary  GraphQL

Body Cookies Headers (6) Test Results

Pretty Raw Preview Visualize Text

```
1 <?xml version="1.0" encoding="UTF-8" ?>
2 <Dial>
3   <DialReplace/>
4 </Dial>
5 |
```

GET http://192.168.1.124/api/dialreplace/export

Params Authorization Headers (6) **Body** Pre-request Script Tests Settings

none  form-data  x-www-form-urlencoded  raw  binary  GraphQL

Body Cookies Headers (5) Test Results

Pretty Raw Preview Visualize **JSON**

```
1 {
2   "retcode": 0,
3   "action": "export",
4   "message": "OK",
5   "data": {
6     "url": "http://192.168.1.124/download/DialReplace.tgz"
7   }
8 }
```

## 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

Request 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/export?type=log
```

POST:

```
{
  "target": "log",
  "action": "export",
  "data" : {
    "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 callog

### api callog get

Request Param	Description
Type	Optional (Default: All) All Dialed Received Missed Forwarded
TimeStart	Start log time <u>This parameter is only supported in X915, S539</u>
TimeEnd	End log time <u>This parameter is only supported in X915, S539</u>
NameOrNumber	Filter by name or number <u>This parameter is only supported in X915, S539</u>
page	Obtain the value corresponding to the specified number of pages, with 10 data per page. If it is not supported, the request would return all dialreplaces in device. 0 and 1 means first 10 dialreplaces. <u>This parameter is only supported in X915, S539</u>

Return Param	Description
ID	Call log index in db. When using request set and del, it is needed.
Type	"Dialed" "Received" "Missed" "Forwarded" "Unknow"
Date	
Time	
Local Identity	Sip Server
Name	Remote DisplayName

Number	Call Num
curPageNum	Number of this page
num	Total count
item	data

Example Request:

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

```
GET: /api/calllog/get?Type=Dialed
```

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

```
POST:
{
  "target": "calllog",
  "action": "get",
  "data": {
    "type": "Dialed"
  }
}
```

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"
      }
    ]
  }
}
```

In S539 and X915, we could search call history by additional parameters, like this

Call History Received 2023-05-10 18:07:31 - 2023-06-08 18:07:31 933 Search Export

<input type="checkbox"/>	Index	Type	Date	Time	Local Identity	Name	Number
<input type="checkbox"/>	1	Received	2023-05-15	00:35:27	192.168.1.124@192.168.1.124	933	192.168.1.52@192.168.1.52
<input type="checkbox"/>	2	Received	2023-05-14	23:56:42	192.168.1.124@192.168.1.124	933	192.168.1.52@192.168.1.52
<input type="checkbox"/>	3	Received	2023-05-14	23:32:32	192.168.1.124@192.168.1.124	933	192.168.1.52@192.168.1.52
<input type="checkbox"/>	4	Received	2023-05-14	23:30:38	192.168.1.124@192.168.1.124	933	192.168.1.52@192.168.1.52
<input type="checkbox"/>	5	Received	2023-05-14	23:07:57	192.168.1.124@192.168.1.124	933	192.168.1.52@192.168.1.52
<input type="checkbox"/>	6	Received	2023-05-14	23:07:04	192.168.1.124@192.168.1.124	933	192.168.1.52@192.168.1.52

Selected: 0/6 Delete Delete All Total: 6 Prev 1/1 Next Go To Page  Go

```
POST:
{
  "target": "callog",
  "action": "get",
  "data": {
    "page": "1",
    "TimeStart": "2023-05-10 18:07:31",
    "TimeEnd": "2023-06-08 18:07:31",
    "NameOrNumber": "933",
    "Type": "Received"
  }
}
```

## api callog del

Param	Description
ID	string, index

Only POST method can be used for this function.

Only one log can be deleted at a time.

Example Request:

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

Example Return:

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

## api callog 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:

Clear all

```
GET: /api/callog/clear
```

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

Clear missed

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

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

Example Return:

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

## api callog export

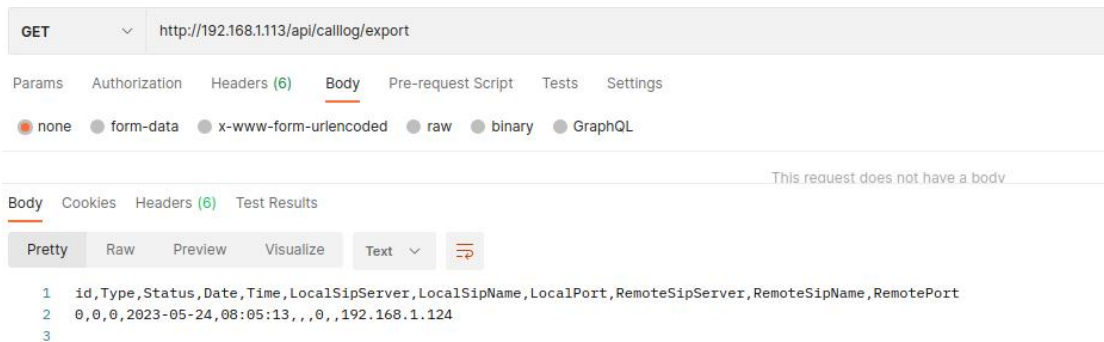
Example Request:

```
GET: /api/callog/export
```

```
POST:
{
  "target": "callog",
  "action": "export",
}
```

Example Return: file or URL for download

```
History.csv
```




GET ▼ http://192.168.1.124/api/calllog/export

Params Authorization Headers (6) **Body** Pre-request Script Tests Settings

none  form-data  x-www-form-urlencoded  raw  binary  GraphQL

---

Body **Cookies** Headers (5) Test Results

Pretty Raw Preview Visualize JSON ▼ 

```
1  |
2  |   "retcode": 0,
3  |   "action": "export",
4  |   "message": "OK",
5  |   "data": {
6  |     |   "url": "http://192.168.1.124/download/History.csv"
7  |     |
8  |     | }
```

## api doorlog

### api doorlog get

Request Param	Description
Type	Optional (Default: All) All Dialed Received Missed Forwarded
TimeStart	Start log time <u>This parameter is only supported in X915, S539</u>
TimeEnd	End log time <u>This parameter is only supported in X915, S539</u>
NameOrNumber	Filter by name or number <u>This parameter is only supported in X915, S539</u>
page	Obtain the value corresponding to the specified number of pages, with 10 data per page. If it is not supported, the request would return all dialreplaces in device. 0 and 1 means first 10 dialreplaces. <u>This parameter is only supported in X915, S539</u>

Return Param	Description	
ID	Door log index in db. When using request set and del, it is needed.	
Type	Card	Open door by card/NFC/BLE
	Password	Open door by pin
	Public	Open door by public code
	Face	Open door by face

	FacePIN	Two-factor authentication: face + pin
	FaceCard	Two-factor authentication: face + card
	CardPIN	Two-factor authentication: card + pin
	QRCode	Temp PIN
	Server	PM open door from cloud server website.
Name	User name	
Status	Failed    open door failed Succ    open door successfully	
Code	Card code、 pin code、 Public PIN...	
Date		
Time		

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

Example Request:

```
GET:/api/doorlog/get
```

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

Example Return:

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

## api doorlog del

Request Param	Description
ID	string, index    get this value by request /api/doorlog/get

Example Request:

```
GET:/api/doorlog/del?ID=1
```



```
POST:
{
  "target": "doorlog",
  "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 or POST method can be used for this function.

Request Param	Description
status	This parameter is only supported in <u>X915, S539</u>

	Success Failed Empty or not set means all
TimeStart	Start log time <u>This parameter is only supported in X915, S539</u>
TimeEnd	End log time <u>This parameter is only supported in X915, S539</u>
NameOrCode	Filter by name or code <u>This parameter is only supported in X915, S539</u>
filetype	xml    get log in xml file csv    get log in csv file

Example Request:

Export with no parameter

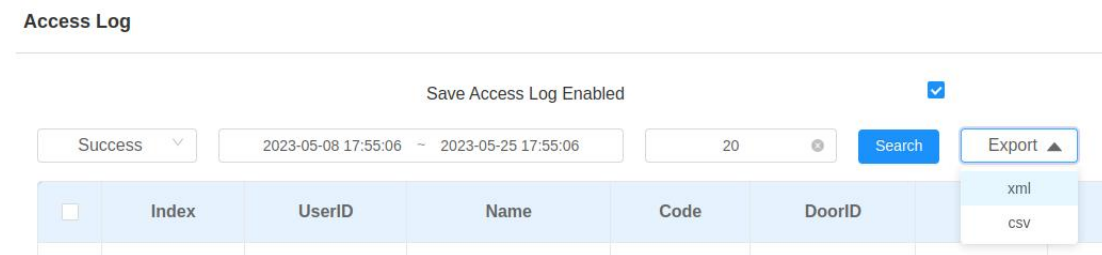
```
GET: /api/doorlog/export
```

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

Example Return:

```
DoorLog.xml
```

Export with parameters:

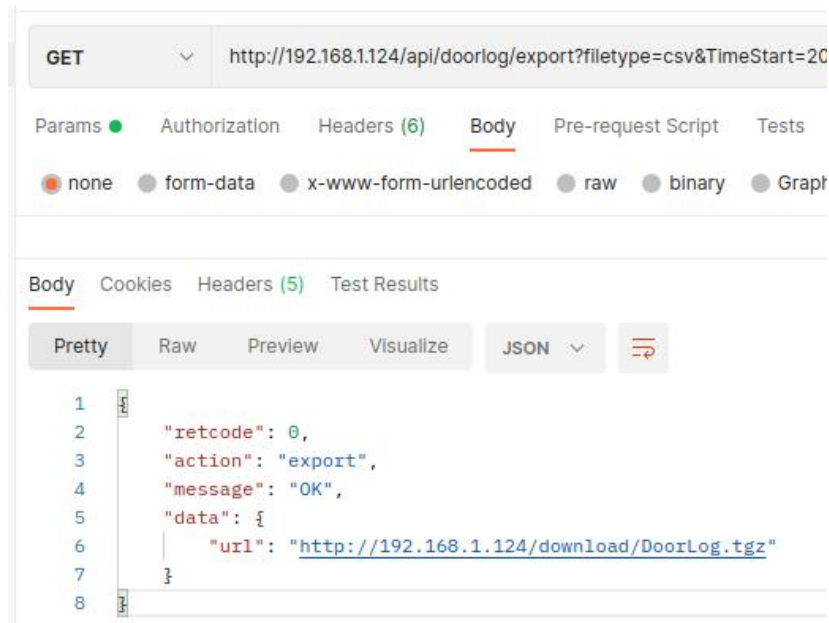


```
GET:
/api/doorlog/export?filetype=csv&TimeStart=2023-05-08
17:55:06&TimeEnd=2023-05-25
17:55:06&NameOrCode=20&Status=Success
```

POST:

```
{
  "target": "doorlog",
  "action": "export",
  "data": {
    "filetype": "csv",
    "TimeStart": "2023-05-08 17:55:06",
    "TimeEnd": "2023-05-25 17:55:06",
    "NameOrCode": "20",
    "Status": "Success"
  }
}
```

Return:



The screenshot shows a REST client interface. At the top, a GET request is shown for the URL `http://192.168.1.124/api/doorlog/export?filetype=csv&TimeStart=20`. Below the URL bar, there are tabs for Params, Authorization, Headers (6), Body, Pre-request Script, and Tests. The 'Body' tab is selected, and the content type is set to 'none'. Below this, there are tabs for Body, Cookies, Headers (5), and Test Results. The 'Body' tab is selected, and the response is displayed in 'Pretty' JSON format. The response is a JSON object with the following structure:

```
1 {
2   "retcode": 0,
3   "action": "export",
4   "message": "OK",
5   "data": {
6     "url": "http://192.168.1.124/download/DoorLog.tgz"
7   }
8 }
```

## api pcap

## api pcap status

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

Return Param	Description
Status	start stop
FileSize	File size now captured.

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

Request Param	Description
type	capture type, optional auto auto stop (capture 60 second) manual do not stop until /api/pcap/stop called
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 29 916 default: 1048576 539 915 default: 1024*1024*50
interface	Network card name, optional If is empty or not set will use default value: eth0
!!!!	To filter by port. Set config <a href="#">Config.Settings.PCAP.PortFilter</a> to the port value first.And then api/pcap/start {

```
"target": "config",
"action": "set",
"data": {
  "Config.Settings.PCAP.PortFilter": 80
}
```

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

Example Request:

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

```
GET: /api//api/pcap/start?type=auto&filter=ip&time=180&size=1048576
```

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

POST:

```
{  
  "target": "pcap",  
  "action": "export"  
}
```

Example Return:

```
phone.pcap
```

GET http://192.168.1.124/api/pcap/export Send

Params Auth Headers (6) **Body** Pre-req. Tests Settings Cookies

none

This request does not have a body

Body Cookies Headers (6) Test Results 200 OK 16 ms 24.82 KB Save Response

Pretty Raw Preview Visualize Text Copy Search

```
1  HTTP/1.1 200 OK
2
3  Content-type: application/json;charset=utf8
4  Content-Length: 153
5  Date: Fri, 26 May 2023 01:12:43 GMT
6  Server: EasyHttpServer
7  X-Frame-Options: SAMEORIGIN
8
9  { "retcode": 0, "action": "start", "message": "OK", "data": { "filter": "ip", "interface":
```

## api contact

**This section is abandoned in S539.**

## api contact get

Request Param	Description
page	Obtain the value corresponding to the specified number of pages, with 10 data per page. If it is not supported, the request would return all dialreplaces in device. 1 means 1st to 10th. 2 means get 11th to 20th. <u>This parameter is only supported in X915</u>
group	Filter by group name if page not set <u>This parameter is only supported in X915</u>
keyword	Filter by contact name if page and group not set <u>This parameter is only supported in X915</u>

Return Param	Description	
num	Total number of local contacts	
curPageNum	Total number in this return. <u>Only supported in 915</u>	
item	DialType	Account to dial 0 auto 1 account1 2 account2
	Group	Group name
	ID	Contact data index in db. When using request set and del, it is needed.
	Name	Contact name
	Phone	Phone number
	Email	Email
	LiftFloorNum	Floor bound to contact. <u>Only supported in 915</u>
	CallPriority	Call Priority in sequential call. <u>Only returned in 915</u>

Example Request:

Get all local contacts

```
GET: /api/contact/get
```



POST:

```
{  
  "target": "contact",  
  "action": "get"  
}
```

Return

```
{  
  "retcode": 0,  
  "action": "get",  
  "message": "OK",  
  "data": {  
    "num": 3,  
    "item": [  
      {  
        "DialType": "2",  
        "Group": "group1",  
        "ID": "2",  
        "Name": "contact2",  
        "Phone": "192.168.3.103"  
      },  
      {  
        "DialType": "0",  
        "Group": "group1",  
        "ID": "3",  
        "Name": "contact3",  
        "Phone": "4444"  
      },  
      {  
        "DialType": "1",  
        "Group": "",  
        "ID": "1",  
        "Name": "joker",  
        "Phone": "192.168.1.144"  
      }  
    ]  
  }  
}
```

<input type="checkbox"/>	Index	Name	Phone	Group	Dial Type	Email	Priority Of Call
<input type="checkbox"/>	1	contact2	<a href="#">192.168.3.103</a>	group1	Account2	rr@qq.com	Secondary
<input type="checkbox"/>	2	contact3	<a href="#">4444</a>	group1	Default		Firstly
<input type="checkbox"/>	3	joker	<a href="#">192.168.1.144</a>	Default	Account1	333@test.com	NULL
<input type="checkbox"/>	4						

## Search contacts with parameters in X915

Local Tenants List

All Tenants

<input type="checkbox"/>	Index	Name	Group	Dial Account	Email	Floor Num	Priority Of Call	Edit
<input type="checkbox"/>	1	3454	Default	Auto		NULL	NULL	<input type="checkbox"/>
<input type="checkbox"/>	2	contact1	group1	Auto	123@qq.com	1:3	NULL	<input type="checkbox"/>
<input type="checkbox"/>	3	contact2	Default	Account1	222@qq.com	NULL	NULL	<input type="checkbox"/>

1/1

POST:

```
{
  "target": "contact",
  "action": "get",
  "data": {
    "page": "1",
    "group": "group1",
    "keyword": ""
  }
}
```

Return

```

{
  "retcode": 0,
  "action": "get",
  "message": "OK",
  "data":
  {
    "curPageNum": 1,
    "num": 1,
    "item":
    [
      {
        "CallPriority": "0",
        "DialType": "0",
        "Email": "123@qq.com",
        "Group": "group1",
        "ID": "1",
        "LiftFloorNum": "1;3",
        "Name": "contact1",
        "Phone": "123"
      }
    ]
  }
}

```

**api contact add**

Request Param	Description
DialType	Account to dial 0 auto 1 account1 2 account2
Group	Group name
ID	The interface will automatically generate new values to overwrite it. It is ok to be empty or to be not provided..
Name	Contact name
Phone	Phone number
Email	Email <u>Only supported in 915</u>
LiftFloorNum	Floor bound to contact. <u>Only supported in 915</u>
CallPriority	Call Priority in sequential call. <u>Only returned in 915</u>

Return Param	Description
num	Number of contacts modified this time

item	Ret	Result of this contact modification
		17 reach the max
		7 the same contact name existed
		0 OK

Add nodes with the same structure into the item if you want to add several contacts in one request.

```

POST:
{
  "target":"contact",
  "action":"add",
  "data":
  {
    "item":
    [
      {
        "Group":"group1",
        "Name":"contactnew",
        "Phone":"123",
        "Email":"123@qq.com",
        "DialType":"0",
        "LiftFloorNum":"1;3",
        "CallPriority":"0"
      }
    ]
  }
}

```

Return

```

{
  "retcode": 0, "action": "add", "message": "OK",
  "data": {
    "num": 1,
    "item": [
      {
        "Ret": 0,
        "CallPriority": "0",
        "DialType": "0",
        "Email": "123@qq.com",
        "Group": "group1",
        "ID": "1",
        "LiftFloorNum": "1;3",
        "Name": "contactnew",
        "Phone": "123"
      }
    ]
  }
}

```

## api contact set

Request Param	Description
DialType	Account to dial 0 auto 1 account1 2 account2
Group	Group name
ID	Contact data index in db. Which one to modify
Name	Contact name
Phone	Phone number
Email	Email <u>Only supported in 915</u>
LiftFloorNum	Floor bound to contact. <u>Only supported in 915</u>
CallPriority	Call Priority in sequential call. <u>Only returned in 915</u>

Return Param	Description	
num	Number of contacts modified this time	
item	Ret	Result of this contact modification -7 the same contact name existed 0 OK
	...	New value data

Add nodes with the same structure into the item if you want to modify several contacts in one request.

```
POST:
{
  "target":"contact",
  "action":"set",
  "data":
  {
    "item":
    [
      {
        "ID":"1",
        "Group":"group1",
        "Name":"contactnew",
        "Phone":"123",
        "Email":"123@qq.com",
        "DialType":"0",
        "LiftFloorNum":"1;3",
        "CallPriority":"0"
      }
    ]
  }
}
```

Return

```
{
  "retcode": 0, "action": "set", "message": "OK",
  "data": {
    "num": 1,
    "item": [
      {
        "Ret": 0,
        "CallPriority": "0",
        "DialType": "0",
        "Email": "123@qq.com",
        "Group": "group1",
        "ID": "1",
        "LiftFloorNum": "1;3",
        "Name": "contactnew",
        "Phone": "123"
      }
    ]
  }
}
```

## api contact del

Request with ids got from api/contact/get

```
POST:
{
  "target": "contact", "action": "del",
  "data": {
    "item": [
      {"ID": "1"}, {"ID": "2"}
    ]
  }
}
```

```

{
  "retcode": 0, "action": "del", "message": "OK",
  "data": {
    "num": 2,
    "item": [
      {
        "Ret": 0,
        "CallPriority": "-1",
        "DialType": "",
        "Email": "",
        "Group": "",
        "ID": "3",
        "LiftFloorNum": "",
        "Name": "3454",
        "Phone": "456464"
      },{
        "Ret": 0,
        "CallPriority": "-1",
        "DialType": "",
        "Email": "",
        "Group": "",
        "ID": "6",
        "LiftFloorNum": "",
        "Name": "343e",
        "Phone": "ertetw"
      }
    ]
  }
}

```

## api contact clear

Example Request:

```
GET: /api/contact/clear
```

POST:

```
{
  "target": "contact",
  "action": "clear",
}
```

Example Return:



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

## api contact export

Request Param	Description
filetype	export file type,X915 dose not support vcard. xml csv (default) vcf vcard

Example Request:

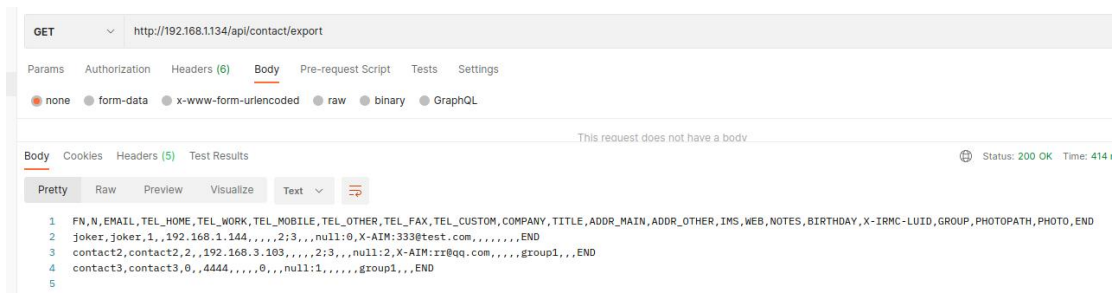
```
GET: /api/log/export
```

```
GET: /api/log/export?filetype=csv
```

```
POST:
{
  "target": "contact",
  "action": "export",
  "data": {
    "filetype": "csv"
  }
}
```

Return

**R29\X916** blocking api, it will not return until the file is prepared.And it return the file data.



**X915** no blocking api, it will return OK if the file preparation task successfully started.

POST http://192.168.1.72/api/contact/add

Params Authorization Headers (8) **Body** Pre-request Script Tests Settings


● none ● form-data ● x-www-form-urlencoded ● **raw** ● binary ● GraphQL **Text** ▾

```

1 {
2   "target": "contact",
3   "action": "export",
4   "data": {
5     "filetype": "csv"
6   }
7 }

```

Body Cookies Headers (5) Test Results

Pretty Raw Preview Visualize JSON ▾ 

```

1 {"retcode": 0,
2  "action": "export",
3  "message": "OK",
4  "data": {}}

```

**Next step is send request to obtain file preparation status**, until got return like this. Download file in the url.

```

1 {"retcode": 0,
2  "action": "status",
3  "message": "OK",
4  "data": {
5    "url": "http://192.168.1.72/download/PhoneBook.tgz"
6  }
7 }

```

If retcode equals to 1 means phone is still preparing file.  
 If retcode equals to -1 means export file failed.

## api contact import

Import .tgz file will failed.it only supports xml csv vcf vcard


POST ▼ http://192.168.1.72/api/contact/import

Params Authorization Headers (8) **Body** ● Pre-request Script Tests Settings

● none ● form-data ● x-www-form-urlencoded ● raw ● binary ● GraphQL

KEY	VALUE
<input checked="" type="checkbox"/>	PhoneBook (1).tgz ×
Key	Value

Body Cookies Headers (5) Test Results

Pretty Raw Preview Visualize JSON ▼ 

```

1  {
2    "retcode": -1,
3    "action": "import",
4    "message": "unspport file type..
5  }
```


POST ▼ http://192.168.1.72/api/contact/import

Params Authorization Headers (8) **Body** ● Pre-request Script Tests Settings

● none ● form-data ● x-www-form-urlencoded ● raw ● binary ● GraphQL

KEY	VALUE
<input checked="" type="checkbox"/>	Contacts.csv ×
Key	Text <span>▼</span> Value

Body Cookies Headers (5) Test Results

Pretty Raw Preview Visualize JSON ▼ 

```

1  {
2    "retcode": 0,
3    "action": "import",
4    "message": "OK"
5  }
```

## api group

### api group get

Request Param	Description
page	<b>String</b> . Obtain the value corresponding to the specified number of pages, with 10 data per page. If it is not supported, the request would return all dialreplaces in device. 1 means 1st to 10th. 2 means get 11th to 20th. <u>This parameter is only supported in X915、S539</u>

Return Param	Description	
num	Total number of groups	
curPageNum	Number of group in this return	
item	ID	Index in db
	Name	Group name

#### Get all local groups

```
GET: /api/group/get
```

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

Return

```
{
  "retcode": 0,
  "action": "get",
  "message": "OK",
  "data": {
    "curPageNum": 2,
    "num": 2,
    "item": [
      {
        "ID": "1",
        "Name": "group1"
      },
      {
        "ID": "2",
        "Name": "group2"
      }
    ]
  }
}
```

Get local group with page in X915 or S539

```
GET: /api/group/get?page=2
```

POST:

```
{
  "target": "group",
  "action": "get",
  "data": {
    "page": "2"
  }
}
```

POST:

```
{
  "target": "group",
  "action": "get",
  "data": {
    "page": "2"
  }
}
```

Return

```
{
  "retcode": 0,
  "action": "get",
  "message": "OK",
  "data": {
    "curPageNum": 0,
    "num": 2,
    "item": []
  }
}
```

```
{
  "retcode": 0,
  "action": "get",
  "message": "OK",
  "data": {
    "curPageNum": 2,
    "num": 2,
    "item": [
      {
        "ID": "1",
        "Name": "group1"
      },
      {
        "ID": "2",
        "Name": "group2"
      }
    ]
  }
}
```

## api group add

Request Param	Description
Name	String .

Return Param	Description
num	Total number of groups
curPageNum	Number of group in this return
item	Ret
	Result: 0 OK 16 Reach max group number 14、27 data is empty 7、28 the same name existed 29 failed to open db
	ID
	Index in db
	Name
	Group name

Add nodes with the same structure into the item if you want to add several groups in one request.like

**"item":[ {"Name":"group1"}, {"Name":"group2"}]**

```
POST:
{
  "target":"group","action":"add",
  "data":{
    "item":[
      {"Name":"group1"}
    ]
  }
}
```

Return

```
{
  "retcode": 0, "action": "add",
  "message": "OK",
  "data": {
    "num": 1,
    "item": [
      {"Ret": 0, "ID": "2", "Name": "group2" }
    ]
  }
}
```

## api group set

Request Param	Description
Name	<b>String</b> .
ID	Group data index in db. Which one to modify

Return Param	Description	
num	Number of groups modified this time	
item	Ret	Result of this group modification
		7 the same group name existed
		0 OK
	...	New value data

```
POST:
{
  "target": "group", "action": "set",
  "data": {
    "item": [
      { "ID": "2", "Name": "group2new" }
    ]
  }
}
```

### Return

```
{
  "retcode": 0, "action": "set",
  "message": "OK",
  "data": {
    "num": 1,
    "item": [
      {
        "Ret": 0,
        "ID": "2",
        "Name": "group2new"
      }
    ]
  }
}
```

## api group clear

Example Request:



```
GET: /api/group/clear
```

```
POST:
```

```
{  
  "target": "group",  
  "action": "clear",  
}
```

Example Return:

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

## api time

### api time get

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

<b>R29/X916 Return Param</b>	<b>Description</b>
local_time	Device time
auto_time	1 auto time 0 manual time
utc_time	UTC time of device
timezone_id	Timezone id in list
auto_timezone	1 auto timezone 0 manual timezone
<b>X915/S539 Return Param</b>	<b>Description</b>
local_time	Number of groups modified this time
auto_time	1 auto time 0 manual time
timezone_display_list	Timezone display name list
timezone_id	Timezone id in list
ntp_server1	Ntp server name

Example Request:

```
GET: http://192.168.1.127/api/time/get
```

POST:

```
{  
  "target": "time",  
  "action": "get"  
}
```

Example Return:

```
{
  "retcode": 0, "action": "get",
  "message": "",
  "data": {
    "local_time": "20230529 10:33:14",
    "utc_time": "20230529 02:33:14",
    "auto_time": 0,
    "auto_timezone": 0,
    "timezone_id": 376
  }
}
```

```
{
  "retcode": 0, "action": "get", "message": "OK",
  "data": {
    "local_time": "20230528 22:28:39",
    "auto_time": "1",
    "timezone_display_list":
    "GMT-11:00\tMidway&GMT-11:00\tNiue&.....&GMT+12:45\tChatham&
    GMT+13:00\tApia&GMT+13:00\tEnderbury&GMT+13:00\tFakaofu&GMT
    +13:00\tTongatapu&",
    "timezone_id": 86,
    "ntp_server1": "pool.ntp.org"
  }
}
```

**api time set**

**R29/X916**

Request Param	Description
auto_time	1 auto time 0 manual time -1 Only modify timezone by timezone name
timezone_id	Timezone id in list

**X915/S539**

Request Param	Description
time_format	0 24h -> 13:00 1 12h -> 1:00 pm
date_format	"MM-dd-yyyy" "dd-MM-yyyy" "yyyy-MM-dd"
auto_time	1 auto time

	0 manual time
timezone_id	Timezone id in list timezone_display_list
ntp_server1	Ntp server name
local_time	Auto_time=0 manual time "2019,9,11,22,13" format: YYYY,mm,dd,hh,mm

1. Only modify timezone by timezone name

```
POST:
{
  "target":"time","action":"set",
  "data":{
    "auto_time":"-1",
    "timezone_id":"Antigua"
  }
}
```

2. Set time to manual time:

**Time**

---

Automatic Date&Time

Date

Time

Time Zone

Date Format

Time Format

NTP Server

```
POST:
{
  "target":"time","action":"set",
  "data":{
    "local_time":"2023,5,31,19,54",
    "auto_time":"0",
    "timezone_id":"93",
    "ntp_server1":"pool.ntp.org",
    "date_format":"0",
    "time_format":"1"
  }
}
```

## api rfkey

### NOTE.

For R29/X916, the version which supports user, interfaces for this section is currently abandoned.  
To get info about card use new interface /api/user

For X915 and S539, the version which supports user, api/rfkey/get api/rfkey/add and api/rfkey/set only works on super card. The required parameters vary from the model.

## api rfkey get

Return Param		Description
num		Total number of super cards
curPageNum		Number of group in this return
item	Code	Card code
	ID	Index in db
	DoorNum	1 relay A 2 relay B 3 relay C Eg. 123 relayABC
	Mon Tue Wed Thur Fri Sat Sun	0 do not open relay when swiping card on mon.... 1 open relay when swiping card on mon....
	Tags Frequency	Abandoned parameters
	TimeStart TimeEnd	Open relay when time is between TimeStart and TimeEnd

for x915, get all super cards data.

Only /api/rfkry/get?type=7 is correct in X915.

```
GET: /api/rfkry/get?type=7
```

```
POST:
{
  "target": "rfkry",
  "action": "get",
  "data":
  {
    "type": "7"
  }
}
```

For S539

```
GET: /api/rfkry/get
```

```
POST:
```

```
{  
  "target": "rfkry",  
  "action": "get"  
}
```

```
{
  "retcode": 0,
  "action": "get",
  "message": "OK",
  "data": {
    "curPageNum": 2,
    "num": 2,
    "item": [
      {
        "Code": "111111111",
        "DoorNum": "123",
        "Frequency": "0",
        "Fri": "1",
        "ID": "2",
        "Mon": "1",
        "Name": "",
        "Sat": "1",
        "Sun": "1",
        "Tags": "0",
        "Thur": "1",
        "TimeEnd": "00:00",
        "TimeStart": "00:00",
        "Tue": "1",
        "Wed": "1"
      },
      {
        "Code": "211111111",
        "DoorNum": "123",
        "Frequency": "0",
        "Fri": "1",
        "ID": "3",
        "Mon": "1",
        "Name": "",
        "Sat": "1",
        "Sun": "1",
        "Tags": "0",
        "Thur": "1",
        "TimeEnd": "00:00",
        "TimeStart": "00:00",
        "Tue": "1",
        "Wed": "1"
      }
    ]
  }
}
```

## api rfkey add

Request Param		Description
item	Code	Card code
	ID	Index in db
	DoorNum	1 relay A 2 relay B 3 relay C Eg. 123 relayABC
	Mon Tue Wed Thur Fri Sat Sun	0 do not open relay when swiping card on mon.... 1 open relay when swiping card on mon....
	Tags Frequency	Abandoned parameters
	TimeStart TimeEnd	Open relay when time is between TimeStart and TimeEnd

Return Param		Description
num		Number of super cards modified this time
item	Ret	Result of this group modification 4 or 7 the same code existed 5 or 10 The sum of cards has reached the maximum limit
	...	New value data

The screenshot shows a dialog box titled "AddSuperCard" with a close button (X) in the top right corner. The dialog contains the following elements:

- A "Code" field with the value "22222222" and an "Obtain" button to its right.
- A "Relay" section with four checked checkboxes: "RelayA", "RelayB", "RelayC", and "Check All".
- At the bottom, there are "Cancel" and "Submit" buttons.



POST:

```
{
  "target": "rfkey", "action": "add",
  "data": {
    "item": [
      {
        "ID": "-1",
        "Code": "22222222",
        "DoorNum": "123",
        "Frequency": "0",
        "Fri": "1", "Mon": "1",
        "Sat": "1", "Sun": "1",
        "Tags": "0",
        "Thur": "1",
        "Tue": "1",
        "Wed": "1",
        "Type": "7",
        "TimeEnd": "00:00",
        "TimeStart": "00:00"
      }
    ]
  }
}
```

Example Return:

```
{
  "retcode": 0,
  "action": "add",
  "message": "OK",
  "data": {
    "num": 1,
    "item": [
      {
        "Ret": 0,
        "Code": "22222222",
        "ID": "4",
        "Name": ""
      }
    ]
  }
}
```

## api rfkey set

Request Param		Description
item	Code	Card code
	<b>Index</b>	Serial number in card list ( X915 /api/rfkey/get?type=7 ) ( S539 /api/rfkey/get ) Count from 0
	ID	Index in db
	DoorNum	1 relay A 2 relay B 3 relay C Eg. 123 relayABC
	Mon Tue Wed Thur Fri Sat Sun	0 do not open relay when swiping card on mon.... 1 open relay when swiping card on mon....
	Tags Frequency	Abandoned parameters
	TimeStart TimeEnd	Open relay when time is between TimeStart and TimeEnd

**EditSuperCard** ✕

---

Code

Relay  RelayA  RelayB  RelayC  Check All

POST:

```
{
  "target": "rfkey", "action": "set",
  "data": {
    "item": [
      {
        "ID": "4",
        "Index": "2",
        "Code": "22222222",
        "DoorNum": "13",
        "Frequency": "0",
        "Fri": "1", "Mon": "1",
        "Sat": "1", "Sun": "1",
        "Tags": "0", "Thur": "1",
        "Type": "7",
        "TimeEnd": "00:00",
        "TimeStart": "00:00",
        "Tue": "1",
        "Wed": "1"
      }
    ]
  }
}
```

## api rfkey del

Only POST method can be used for this function.

Example Request:

```
{
  "target": "rfkey",
  "action": "del",
  "data": {
    "item": [
      {"ID": "2"},
      {"ID": "3"}
    ]
  }
}
```

Example Return:

```
{
  "retcode": 0, "action": "del",
  "message": "OK",
  "data": {
    "num": 1,
    "item": [
      {
        "Ret": 0,
        "Name": "",
        "ID": "2",
        "Code": "11111111"
      },
      {
        "Ret": 0,
        "Name": "",
        "ID": "3",
        "Code": "22222222"
      }
    ]
  }
}
```

## api rfkey clear

GET or POST method can be used for this function.

clear super card

Example Request:

In S539

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

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

In X915

```
GET: /api/rfkey/clear?type=7
```

```
{
  "target": "rfkey",
  "action": "clear",
  "data": {
    "type": "7"
  }
}
```

#### Example Return:

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

## api schedule

Param	Description
ID	e.g. "ID"."1" The id of data in database
DisplayID	ID used to bind with user
DateStart	YYYYMMDD 20220111
DateEnd	YYYYMMDD 20220215
TimeStart	HH:MM:SS 00:00:00
TimeEnd	HH:MM:SS 23:59:59
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
Name	Schedule name
Type	0: Normal 1: Weekly 2: Daily
SourceType	-1:Local 0:Cloud 1:ACMS 2:SDMC

## api schedule get

Only POST method can be used for this function

Request Param	Description
page	Optional filling -1 or without this param - Get all schedule 1 - Get first page schedule items ( 10 items every page)
type	Optional filling To get all the information about the schedule or a simple list. <u>Only supported in X915 and S539</u>
name	Optional filling search schedule by keyword of name <u>Only supported in X915 and S539</u>
SourceType	Optional filling Search by source without this param - all Local - get local schedule Cloud - get schedule from cloud ACMS - get schedule from ACMS SDMC - get schedule from SDMC

Return Param	Description
curPageNum	current number of page
num	Total number of schedule meets filtering criteria

### 1. Get first page data

Example Request:

```
{
  "target": "schedule",
  "action": "get",
  "data" : {
    "page": "1"
  }
}
```

Example Return:

```
{
  "retcode": 0,
  "action": "get",
  "message": "OK",
  "data": {
    "curPageNum": 4,
    "num": 4,
    "item": [
      {
        "DateEnd": "20220215",
        "DateStart": "20220111",
        "DisplayID": "1",
        "Fri": "0",
        "ID": "4",
        "Mon": "1",
        "Name": "123",
        "Sat": "1",
        "SourceType": "-1",
        "Sun": "1",
        "Thur": "1",
        "TimeEnd": "20:57",
        "TimeStart": "02:04",
        "Tue": "1",
        "Type": "0",
        "Wed": "1"
      },
      {
        "DateEnd": "",
        "DateStart": "",
        "DisplayID": "1002",
        "Fri": "1",
        "ID": "2",
        "Mon": "1",
        "Name": "Never",
        "Sat": "1",
        "SourceType": "-1",
        "Sun": "1",
        "Thur": "1",
        "TimeEnd": "00:00:00",
        "TimeStart": "00:00:00",
        "Tue": "1",
        "Type": "2",
      }
    ]
  }
}
```



## 2. Get all local schedules

Example Request:

```
{
  "target": "schedule",
  "action": "get",
  "data": {
    "SourceType": "Local"
  }
}
```

Example Return:

```
{
  "retcode": 0,
  "action": "get",
  "message": "OK",
  "data": {
    "curPageNum": 5,
    "num": 5,
    "item": [
      {
        "DateEnd": "00000000",
        "DateStart": "00000000",
        "DisplayID": "3",
        "Fri": "1",
        "ID": "10",
        "Mon": "1",
        "Name": "schedule_daily",
        "Sat": "1",
        "SourceType": "-1",
        "Sun": "1",

```

### 3. Get all local schedules basic info.

Example Request:

```
{
  "target": "schedule",
  "action": "get",
  "data":
  {
    "SourceType": "Local",
    "type": "list"
  }
}
```

Example Return:

```
{
  "retcode": 0,
  "action": "get",
  "message": "OK",
  "data": {
    "curPageNum": 5,
    "num": 5,
    "item": [
      { "DisplayID": "3", "ID": "10", "Name": "schedule_daily" },
      { "DisplayID": "2", "ID": "9", "Name": "schedule_weekly"},
      { "DisplayID": "1", "ID": "8", "Name": "schedule_normal"},
      { "DisplayID": "1002", "ID": "2", "Name": "Never" },
      { "DisplayID": "1001", "ID": "1", "Name": "Always"}
    ]
  }
}
```

## api schedule add

Only POST method can be used for this function.

Example Request:

Return Param		Description
num		Number of schedules added this time
item	Ret	Result of this schedule addition 0 OK -1 failed to open db -2 or 10 Maximum amount is reached -4 or 7 Schedule name existed
...		New value data

POST:

```
{
  "target": "schedule",
  "action": "add",
  "data":
  {
    "item":
    [
      {
        "Type": "0",
        "Name": "schedule_normal",
        "DisplayID": "0",
        "DateStart": "20220101",
        "DateEnd": "20220304",
        "Mon": "0",
        "Tue": "0",
        "Wed": "1",
        "Thur": "1",
        "Fri": "1",
        "Sat": "1",
        "Sun": "1",
        "TimeStart": "01:03",
        "TimeEnd": "20:57"
      },
      {
        "Type": "1",
        "Name": "schedule_weekly",
        "DisplayID": "0",
        "DateStart": "00000000",
        "DateEnd": "00000000",
        "Mon": "1",
        "Tue": "1",
        "Wed": "1",
        "Thur": "1",
        "Fri": "1",
        "Sat": "1",
        "Sun": "1",
        "TimeStart": "02:02",
        "TimeEnd": "20:57"
      },
      {
        "Type": "2",
        "Name": "schedule_daily",
        "DisplayID": "0",
```

### Example Return:

```
{
  "retcode": 0,
  "action": "add",
  "message": "OK",
  "data": {
    "num": 3,
    "item": [
      {
        "Ret": 0,
        "Type": 0,
        "ID": "-1",
        "Name": "schedule_normal"
      },
      {
        "Ret": 0,
        "Type": 1,
        "ID": "-1",
        "Name": "schedule_weekly"
      },
      {
        "Ret": 0,
        "Type": 2,
        "ID": "-1",
        "Name": "schedule_daily"
      }
    ]
  }
}
```

## api schedule set

Only POST method can be used for this function.

Add nodes with the same structure into the item if you want to set several schedules in one request.

Return Param		Description
num		Number of schedules modified this time
item	Ret	Result of this schedule addition 0 OK -1 failed to open db -4 or 7 Schedule name existed
	...	New value data

Example Request:

POST:

```
{
  "target": "schedule", "action": "set",
  "data": {
    "item": [
      {
        "ID": "8",
        "Type": "0",
        "Name": "schedule_normal2",
        "DisplayID": "1",
        "DateStart": "20220119",
        "DateEnd": "20220331",
        "Mon": "1",
        "Tue": "0",
        "Wed": "1",
        "Thur": "0",
        "Fri": "1",
        "Sat": "1",
        "Sun": "0",
        "TimeStart": "02:03",
        "TimeEnd": "20:59"
      }
    ]
  }
}
```

Example Return:

```
{
  "retcode": 0, "action": "set",
  "message": "OK",
  "data":
  {
    "num": 1,
    "item": [
      {
        "Ret": 0,
        "Type": 0,
        "ID": "8",
        "Name": "schedule_normal2"
      }
    ]
  }
}
```

## api schedule del

Only POST method can be used for this function.

Example Request:

```
{
  "target": "schedule",
  "action": "del",
  "data": {
    "item": [
      {"ID": "2"},
      {"ID": "3"}
    ]
  }
}
```

Example Return:

```
{
  "retcode": 0, "action": "del",
  "message": "OK",
  "data": {
    "num": 1,
    "item": [
      {
        "Ret": 0,
        "Type": -1,
        "ID": "2",
        "Name": "name1"
      },
      {
        "Ret": 0,
        "Type": -1,
        "ID": "3",
        "Name": "name2"
      }
    ]
  }
}
```



## api schedule clear

GET or POST method can be used for this function.

clear local schedule

Example Request:

```
GET: /api/schedule/clear
```

```
{  
  "target": "schedule",  
  "action": "clear"  
}
```

Example Return:

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

## api schedule export

GET or POST method can be used for this function.

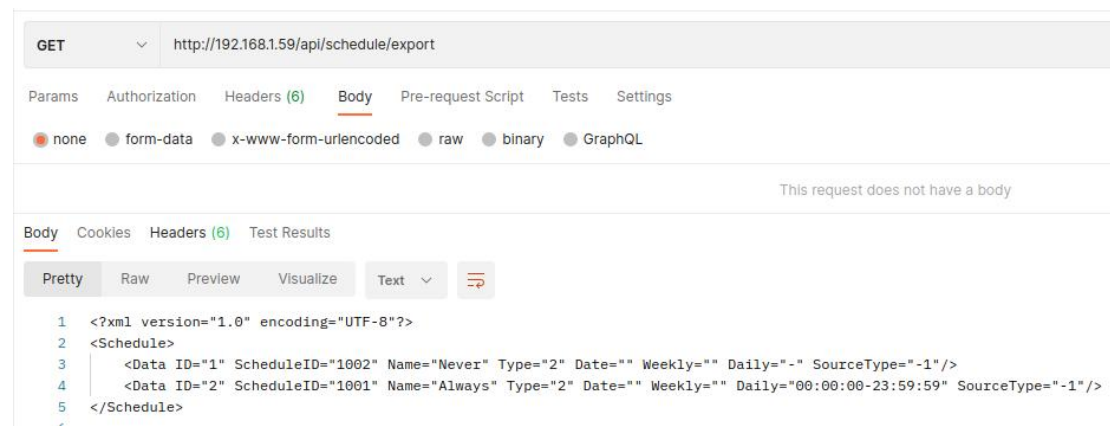
Example Request:

```
GET: /api/schedule/export
```

```
{
  "target": "schedule",
  "action": "export"
}
```

Example Return:

**R29 and X916** return the xml file directly



**X915 and S539** return the url to download .tgz file

```
{
  "retcode": 0,
  "action": "export",
  "message": "OK",
  "data": {
    "url": "http://192.168.1.26/download/DoorSchedule.tgz"
  }
}
```

## api schedule import

In X915 and S539, this request is replaced with `/api/user/import?type=Schedule`

Example :

POST:http://192.168.1.26//api/user/import?type=Schedule

POST http://192.168.1.26//api/user/import?type=Schedule

Params Authorization Headers (1) **Body** Pre-request Script Tests Cookies Cod

none form-data x-www-form-urlencoded raw binary

KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/> config	Schedule.xml X	
Key	Value	Description

Body Cookies Headers (4) Test Results Status: 200 OK Time: 94 ms Size: 201 B

Pretty Raw Preview JSON

```
1 {
2   "retcode": 0,
3   "action": "import",
4   "message": "OK"
5 }
```

In R29 and X916

POST http://192.168.1.59/api/schedule/import

Params Authorization Headers (8) **Body** Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary GraphQL

KEY	VALUE
<input checked="" type="checkbox"/>	Schedule.xml X

Body Cookies Headers (5) Test Results

Pretty Raw Preview Visualize JSON

```
1 {
2   "retcode": 0,
3   "action": "import",
4   "message": "OK"
5 }
```

**api user**

**api user get**

Request Param	Description
page	Optional filling

	-1 or without this param - Get all schedule 1 - Get first page schedule items ( 10 items every page)
NameOrPerID (R29\X916\X915) NameOrCode (S539)	Optional filling search userby keyword of name or perid <u>Only supported in X915 and S539</u>
SourceType	Optional filling Search by source without this param - all Local - get local schedule Cloud - get schedule from cloud ACMS - get schedule from ACMS SDMC - get schedule from SDMC

Return parameters vary from models.In **R29/x916**

Return Param	Description
num	Total number of data that meets the filtering criteria
curPageNum	Number of data in this return
item	Building
	Building name
	CardCode
	e.g."CardCode": "11111111;22222222" Card code
	FaceUrl
	Url of face img
	FingerUrl
	Url of finger file <u>Only supported in R29</u>
	FingerId
	finger id for user
	ID
	e.g. "ID": "1" The id of data in database
	LiftFloorNum
	e.g. "LiftFloorNum": "1;2;4" lift floor
	Name
	User name
	PrivatePIN
	e.g. "PrivatePIN": "1234;12345" private key
	Room
	Room name
	ScheduleRelay
	"ScheduleRelay": "1001-13;1002-13" scheduleID-relay
	Type
	-1 - get local schedule 0 - get schedule from cloud 1 - get schedule from ACMS 2 - get schedule from SDMC
	UserID
	User perID
	WebRelay
	Webrelay id

**In X915/S539**

Return Param	Description
num	Total number of data that meets the filtering criteria

curPageNum	Number of data in this return
item	Building
	Building name
	CardCode
	Card code 11111111;22222222
	ID
	e.g. "ID": "1" The id of data in database
	LiftFloorNum
	e.g. "LiftFloorNum": "1;2;4" lift floor
	Name
	User name
	PrivatePIN
	e.g. "PrivatePIN": "1234;12345" private key
	Room
	Room name
	ScheduleRelay
	"ScheduleRelay": "1001-13;1002-13" scheduleID-relay
	Type
	-1 - get local schedule 0 - get schedule from cloud 1 - get schedule from ACMS 2 - get schedule from SDMC
	UserID
	User perID
	WebRelay
	Webrelay id
	FaceStatus
	1 Face has been registered 0 Face not registered
	AuthMode
	Reserved fields
	ContactID
	Contact id of user <u>Only supported in S539</u>
	DialAccount
	Dial account <u>Only supported in S539</u>
	Group
	Group name <u>Only supported in S539</u>
	PhoneNum
	Phone number <u>Only supported in S539</u>
	Priority
	Call priority <u>Only supported in S539</u>

Get all user data

```
GET: /api/user/get
```

Get local user in page 1

```
GET: /api/user/get?page=1&SourceType=Local
```

Get local user and name or perid contains "123".

R29/X916/X915

```
GET: /api/user/get?page=1&SourceType=Local&NameOrPerID=123
```

```
POST:
{
  "target": "user",
  "action": "get",
  "data":
  {
    "SourceType": "Local",
    "NameOrPerID": "123",
    "page": "1"
  }
}
```

S539

```
GET: /api/user/get?page=1&SourceType=Local&NameOrCode=123
```

```
POST:
{
  "target": "user",
  "action": "get",
  "data":
  {
    "SourceType": "Local",
    "NameOrCode": "123",
    "page": "1"
  }
}
```

X916

```
GET http://192.168.1.22/api/user/get

Params Authorization Headers (6) Body Pre-request Script

Body Cookies Headers (4) Test Results

Pretty Raw Preview Visualize JSON

1
2   "retcode": 0,
3   "action": "get",
4   "message": "OK",
5   "data": {
6     "curPageNum": 1,
7     "num": 1,
8     "item": [
9       {
10        "Building": "buildingname",
11        "CardCode": "11111111;22222222",
12        "FaceUrl": "",
13        "ID": "1",
14        "LiftFloorNum": "1;2;4",
15        "Name": "username1",
16        "PrivatePIN": "1234;12345",
17        "Room": "",
18        "ScheduleRelay": "1001-13;1002-13",
19        "Type": "-1",
20        "UserID": "1",
21        "WebRelay": "2"
22      }
23    ]
24  }
25
```

R29

```
GET http://192.168.1.59/api/user/get

Params Authorization Headers (6) Body Pre-request Script

Body Cookies Headers (5) Test Results

Pretty Raw Preview Visualize JSON

1  {
2    "retcode": 0,
3    "action": "get",
4    "message": "OK",
5    "data": {
6      "curPageNum": 1,
7      "num": 1,
8      "item": [
9        {
10       "Building": "buildingname",
11       "CardCode": "11111111;22222222",
12       "FaceUrl": "",
13       "FingerId": "0",
14       "FingerUrl": "",
15       "ID": "1",
16       "LiftFloorNum": "9;10;11;",
17       "Name": "username",
18       "PrivatePIN": "1234;12345",
19       "Room": "",
20       "ScheduleRelay": "1001-1;1-1",
21       "Type": "-1",
22       "UserID": "1",
23       "WebRelay": "5"
24     }
25   ]
26 }
27 }
```

S539



```
GET http://192.168.1.21/api/user/get

Params Authorization Headers (6) Body Pre-request Script Tests Settings

Body Cookies Headers (5) Test Results

Pretty Raw Preview Visualize JSON

1 {
2   "retcode": 0,
3   "action": "get",
4   "message": "OK",
5   "data": {
6     "curPageNum": 16,
7     "num": 16,
8     "item": [
9       {
10        "Type": 0,
11        "AuthMode": "0",
12        "Building": "",
13        "CardCode": "B7E395D40C216AB8;F038F26BD410ACE5",
14        "ContactID": "-1",
15        "DialAccount": "0",
16        "FaceStatus": "1",
17        "Group": "Default",
18        "ID": "16",
19        "LiftFloorNum": "",
20        "Name": "s345 dd",
21        "PhoneNum": "",
22        "Priority": "0",
23        "PrivatePIN": "1003+445566",
24        "Room": "",
25        "ScheduleRelay": "4569-1",
26        "UserID": "5115102135",
27        "WebRelay": "0"
28      },
29    ]
30   "Type": 0,
```

X915

GET ⌵ http://192.168.1.72/api/user/get

Params Authorization Headers (6) Body Pre-request Script

Body Cookies Headers (5) Test Results

Pretty Raw Preview Visualize JSON ⌵ 

```
1  {
2    "retcode": 0,
3    "action": "get",
4    "message": "OK",
5    "data": {
6      "curPageNum": 1,
7      "num": 1,
8      "item": [
9        {
10         "Building": "",
11         "CardCode": "7CE2CE70",
12         "DoorNum": "12",
13         "EffectiveType": "2",
14         "FaceStatus": "1",
15         "Frequency": "1",
16         "ID": "1",
17         "LiftFloorNum": "3",
18         "Name": "1231",
19         "PrivatePIN": "1111",
20         "Room": "",
21         "Schedule": "1001",
22         "Type": "-1",
23         "UserID": "1",
24         "WebRelay": "0"
25       }
26     ]
27   }
28 }
```

## api user info

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

Return Param	Description
AuthorizationMode	the authorization mode of pin (pin or apt+pin) 1 pin e.g. 1234;1212 0 apt+pin e.g. 123+123;2334+2123
RecommendId	the recommend id for user's UserID

Example Request:

GET: /api/user/info

Example Return:

```
{
  "retcode": 0,
  "action": "info",
  "message": "OK",
  "data": {
    "AuthorizationMode": 1,
    "RecommendId": 2
  }
}
```

## api user add

In R29\X916

Request Param		Description
item	Building	Building name
	CardCode	e.g."CardCode": "11111111;22222222" Card code
	FaceUrl	Optional. Url of face img. Empty or not given no face data to register URL device will download and register background

FingerUrl	Url of finger file <u>Only supported in R29</u> Empty or not given no finger data to register URL device will download and register background
ID	e.g. "ID": "1" The id of data in database
LiftFloorNum	e.g. "LiftFloorNum": "1;2;4" lift floor
Name	User name
PrivatePIN	AuthorizationMode is 1 e.g. "PrivatePIN": "1234;12345" AuthorizationMode is 0 e.g. "PrivatePIN": "123+1234;123+3345"
Room	Room name
ScheduleRelay	"ScheduleRelay": "1001-13;1002-13"
UserID	User perID
WebRelay	Webrelay id

Return Param	Description
retcode	USER_MODMOV_PER_ID_EMPTY = -7, USER_MODMOV_PRIVATE_KEY_SAME = -6, USER_MODMOV_EDIT_FAILED = -5, USER_MODMOV_REACH_MAX = -4, USER_MODMOV_PER_ID_SAME = -3, USER_MODMOV_IC_KEY_SAME = -2, USER_MODMOV_FAILED = -1, USER_MODMOV_FINISHED = 0, USER_MODMOV_SUCCESS = 1, USER_MODMOV_EDIT_SUCCESS = 2,
message	details

Example request:

1. add user without face url and finger url in R29

```

{
  "target": "user",
  "action": "add",
  "data":
  {
    "item":
    [
      {
        "UserID": "per3",
        "PrivatePIN": "234557;1233",
        "LiftFloorNum": "2;3;4",
        "Name": "user3",
        "ScheduleRelay": "1001-12",
        "CardCode": "2342315;33221122",
        "Building": "",
        "Room": "",
        "WebRelay": "0"
      }
    ]
  }
}



```

Example Return:

```

{
  "retcode": 1,
  "action": "add",
  "message": "OK"
}

```

<input type="checkbox"/> Index	Source	User ID	Name	Private PIN	RF Card	Face	Floor No.	Web Relay	Schedule-Relay	Edit
<input type="checkbox"/> 1	Local	per3	user3	1233;234557	33221122; 23423 15		2;3;4	0	1001-12	

## 2.add user with face url in R29 or X916

```
{
  "target": "user",
  "action": "add",
  "data":
  {
    "item":
    [
      {
        "UserID": "per4",
        "PrivatePIN": "2345",
        "LiftFloorNum": "2;4",
        "Name": "userface",
        "ScheduleRelay": "1001-12",
        "CardCode": "23424343",
        "Building": "",
        "Room": "",
        "WebRelay": "0",
        "FaceUrl": "http://192.168.1.48:8088/3.png"
      }
    ]
  }
}
```

### Example Return:

```
{
  "retcode": 1,
  "action": "add",
  "message": "OK"
}
```

<input type="checkbox"/> Index	Source	User ID	Name	Private PIN	RF Card	Face	Floor No.	Web Relay	Schedule-Relay	Edit
<input type="checkbox"/> 1	Local	per4	userface	2345	23424343		2;4	0	1001-12	

Request Param(X915)		Description
item	Building	Building name
	CardCode	Card code e.g."CardCode": "11111111:22222222"
	ID	e.g. "ID": "1" The id of data in database

LiftFloorNum	e.g. "LiftFloorNum": "1;2;4" lift floor
Name	User name
FaceStatus	Is there face data registered 0 no 1 yes
PrivatePIN	AuthorizationMode is 1 e.g. "PrivatePIN": "1234;12345" AuthorizationMode is 0 e.g. "PrivatePIN": "123+1234;123+3345"
Room	Room name
Schedule	"Schedule": "1:1001"
DoorNum	"DoorNum": "13" 1 relay a 2 relay b 3 relay c SA security relay a SB security relay b
UserID	User perID
WebRelay	Webrelay id

Return Param(X915)	Description
retcode	OK = 0, PRIVATE_KEY_EXISTED = 1, PRIVATE_KEY_REACH_MAX = 2, PRIVATE_KEY_FILE_OPEN_FAILED = 3, RFCARD_EXISTED = 4, RFCARD_REACH_MAX = 5, RFCARD_FILE_OPEN_FAILED = 6, USER_NAME_EXISTED = 7, USER_REACH_MAX = 8, USER_FILE_OPEN_FAILED = 9, USER_KEY_REACH_MAX = 10, USER_PERID_EXISTED = 11, SCHEDULE_REACH_MAX = 12
message	details

Request Param(S539)	Description
item	Building
	Building name
	CardCode
	Card code e.g. "CardCode": "11111111:22222222"
	ID
	e.g. "ID": "1" The id of data in database
	LiftFloorNum
	e.g. "LiftFloorNum": "1;2;4" lift floor

	Name	User name
	FaceStatus	Is there face data registered 0 no      1 yes
	PrivatePIN	AuthorizationMode is 1 e.g. "PrivatePIN": "1234;12345" AuthorizationMode is 0 e.g. "PrivatePIN": "123+1234;123+3345"
	Room	Room name
	ScheduleRelay	"ScheduleRelay": "1"
	UserID	User perID
	WebRelay	Webrelay id
	PhoneNum	Phone number
	Group	group for phone number
	Email	Email for phone number
	PriorityCall	Call priority in sequence call 1 Primary 2 Secondary 3 Tertiary
	DialAccount	Which account to call 0 auto 1 account1 2 account2

Return Param(S539)	Description
retcode	0 CFG_ERR_SUCCESS 24 CFG_ERR_PRIVATE_KEY_ALREADY_EXIST 23 CFG_ERR_IC_KEY_ALREADY_EXIST 26 CFG_ERR_USER_ID_ALREADY_EXIST 10 CFG_ERR_COUNT_REACH_MAX 30 CFG_ERR_CONTACT_NUMBER_EXIST -1 CFG_ERR_FAILED
message	details

Example request

**Step1** check if security relay A and security relay B is enabled?



```
{
  "target":"config",
  "action":"get",
  "data":
  {
    "filter":[
      "Config.DoorSetting.RELAY.SecurityRelayAEnabled",
      "Config.DoorSetting.RELAY.SecurityRelayBEnabled",
      "Config.DoorSetting.GENERAL.Output12vPower"
    ]
  }
}
```

If SecurityRelayAEnabled = 1 and Output12vPower = 3 relay A will be disabled, other wise security relay a will be disabled.

**Security Relay**

Relay ID	Security Re...▼	Security Re...▼
Connect Type	Relay A Po...▼	RS485 ▼
Trigger Delay(Sec)	0 ▼	0 ▼
Hold Delay(Sec)	5 ▼	5 ▼
1 Digit DTMF	2 ▼	3 ▼
2~4 Digits DTMF	013	014
Relay Name	Security Relay A	Security Relay B
	Enabled <input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
	<input type="button" value="Test"/>	<input type="button" value="Test"/>

**12V Power Output**

Relay ID	RelayA
12V Power Output	Security Relay A ▼

**Step 2** get local schedule list info

**GET /api/web/schedule/get?type=list&SourceType=Local**

```
{
  "retcode": 0, "action": "get",
  "message": "OK",
  "data": {
    "curPageNum": 3,
    "num": 3,
    "item": [
      { "DisplayID": "1", "ID": "3", "Name": "schedule1" },
      { "DisplayID": "1002", "ID": "2", "Name": "Never" },
      { "DisplayID": "1001", "ID": "1", "Name": "Always" }
    ]
  }
}
```

**Step 3** get pin authorization mode and recommend id

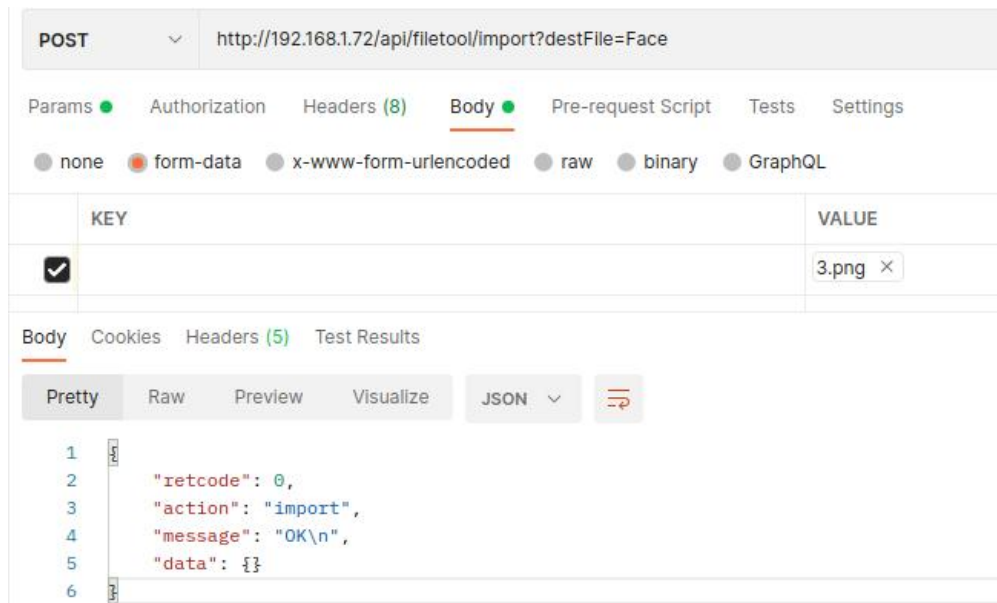
**GET** [/api/web/user/info](#)

```
{
  "retcode": 0,
  "action": "info",
  "message": "OK",
  "data": {
    "AuthorizationMode": 1,
    "RecommendId": 4
  }
}
```

**Step 4**

If this user has face data to register

**/api/web/filetool/import?destFile=Face**



After file upload successfully, Query image detection results at regular intervals until detectFacePicRet value is not 1.

[GET /api/web/filetool/status?type=detectFacePic](#)

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

Return Param	Description
detectFacePicRet	1 detecting 0 ok -9 The photo should be higher than 250x250 pixels. -8 Please select another photo without mask. -7 Please use the front view of the face. -6 No face were detected in this photo. -5 The face was no clear enough. Please make sure your face is not covered, stretched and no backlight. -4 The face in the photo is too small. -3 The face in the photo is too large. -2 More than one face were detected in this photo.

### Step 5

Set FaceFileName to the file name uploaded in Step 4. If there is not face data to register skip step 4 and set FaceFileName empty.

**IN x915**

```

{
  "target":"user",
  "action":"add",
  "data":
  {
    "item":[
      {
        "UserID":"3",
        "DoorNum":"13",
        "PrivatePIN":"3423;2342",
        "LiftFloorNum":"4;6",
        "Name":"user3",
        "Schedule":"1:1001",
        "CardCode":"43455555:54543234",
        "Building":"",
        "EffectiveType":"2",
        "Frequency":"0",
        "Room":"",
        "FaceFileName":"3.png"
        "WebRelay":"3"
      }
    ]
  }
}


```

**Example Return:**

```

{
  "retcode": 0,
  "action": "add",
  "message": "OK"
}

```

Index	Source	UserID	Name	Private PIN	RF Card	Face	Floor No.	Web Relay	Schedule-Relay	Edit
1	Local	5	userface	4334,3434	56575,4564674,676...	✓	NULL	2	1-23SASB:1001-23...	

Source	UserID	Name	Private PIN	RF Card	Face	Floor No.	Web Relay	Schedule-Relay
Local	3	user3	3423,2342	43455555,54543234	✗	4;6	3	1-13;1001-13

**IN S539**

```
{
  "target":"user",
  "action":"add",
  "data":
  {
    "item":[
      {
        "UserID":"1",
        "Name":"539user",
        "PrivatePIN":"2333+333;434+444",
        "LiftFloorNum":"2;4",
        "ScheduleRelay":"1001-13;1002-13",
        "CardCode":"74574744;67767676",
        "Building":"","Room":"","",
        "AuthMode":"3",
        "FaceFileName":"3.jpg",
        "WebRelay":"3",
        "PhoneNum":"232",
        "PriorityCall":"2",
        "DialAccount":"1",
        "Group":"Default"
      }
    ]
  }
}
```

Source	UserID	Name	Private PIN	RF Card	Face	Floor No.	Web Relay	Schedule-Relay
Local	1	539user	2333+333;434+444	74574744;67767676	✔	2;4	3	1001-13;1002-13

**Facial registration takes a certain amount of time and will only be added to the database after successful registration. So after successfully adding, immediately check the facial registration status of user may get "FaceStatus":"0".**

## api user set

The format of modifying the user's command is the same as when adding it, except that it needs to add parameter ID.

For example To delete face data registered already

In R29/X916 request with FaceUrl to empty, e.g. "FaceUrl":""

```
{
  "target": "user",
  "action": "add",
  "data":
  {
    "item":
    [
      {
        "ID": "4",
        "UserID": "per4",
        "PrivatePIN": "2345",
        "LiftFloorNum": "2;4",
        "Name": "userface",
        "ScheduleRelay": "1001-12",
        "CardCode": "23424343",
        "Building": "",
        "Room": "",
        "WebRelay": "0",
        "FaceUrl":""
      }
    ]
  }
}
```

In X915/S539, request as follow is designed to delete user face data.

```
{ "target":"user", "action":"del",
  "data":{
    "item":[
      {"ID":"17","DelType":"Face"}
    ]
  }
}
```

## api user del

Only POST method can be used for this function.

Example Request:

Delete user data including private pin, card, face...

```
{
  "target": "user",
  "action": "del",
  "data": {
    "item": [
      {"ID": "2"},
      {"ID": "3"}
    ]
  }
}
```

In X915 and S539

Return Param	Description
ID	Id in db, get it from /api/user/get
DelType	E.g. {"ID":"17","DelType":"Face"} means delete user`s face which id is 17.

## api user clear

GET or POST method can be used for this function.

clear local user

Return Param	Description
type	UserData      dalete local user data DoorSchedule    delete local schedule data

Example Request:

```
GET: /api/user/clear?type=UserData
```

```
POST
{
  "target": "user",
  "action": "clear",
  "data": {
    "type": "UserData"
  }
}
```

Example Return:

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

## api user export

**Only S539 and X915 support export.**

GET or POST method can be used for this function.

Request Param	Description
type	UserData      export local user data DoorSchedule    export local schedule data

Example Request:



```
GET: /api/user/export?type=UserData
```

```
{
  "target": "user",
  "action": "export",
  "data": {
    "type": "UserData"
  }
}
```

If type is DoorSchedule, it will return the url to download .tgz file after file prepared successfully.

If type is UserData in **X915** it is a **blocking api**. return the url to download .tgz file after file prepared successfully.

```
{
  "retcode": 0,
  "action": "export",
  "message": "OK",
  "data": {
    "url": "http://192.168.1.26/download/DoorSchedule.tgz"
  }
}
```

In **s539** **no blocking api**, it will return OK if the file preparation task successfully started.

```
{
  "retcode": 0,
  "action": "export",
  "message": "OK",
  "data": {
    "url": ""
  }
}
```

It returns an empty url. We need access

**GET /api/user/status?type=exportImportUser**

to monitor document preparation status. More details about

/api/user/status?type=exportImportUser see [api user status](#)

**GET /api/user/status?type=exportImportUser**

```
{ "retcode": 0, "action": "status", "message": "OK", "data": { "cur": 0, "errCode": 0, "step": 2, "total": 0, "action": "export" } }
```

**GET /api/user/status?type=exportImportUser**

```
{ "retcode": 0, "action": "status", "message": "OK", "data": { "cur": 0, "errCode": 0, "step": 6, "total": 0, "action": "export", "url": "/download/UserData.tgz" } }
```

If action is export and step is 6, URL to download file will be there. Download it.

**GET /download/UserData.tgz**

Before download, access /api/commontool/set to notify device that file exporting OK.

```
{
  "target": "commontool",
  "action": "set",
  "data": {
    "action": "ResetUserExportStatus"
  }
}
```

## api user import

**Only S539 and X915 support import.**

Request Param	Description
type	UserData import local user data DoorSchedule import local schedule data

POST http://192.168.1.72/api/user/import?type=DoorSchedule

Params Authorization Headers (8) **Body** Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary GraphQL

KEY	VALUE
<input checked="" type="checkbox"/>	Schedule.xml ×

body Cookies Headers (5) Test Results

Pretty Raw Preview Visualize JSON

```
1
2   "retcode": 0,
3   "action": "import",
4   "message": "OK"
5
```

If type is DoorSchedule, it will return after importing finished.

POST ▼ http://192.168.1.72/api/user/import?type=UserData

Params ● Authorization Headers (8) **Body** ● Pre-request Script Tests Settings

none  form-data  x-www-form-urlencoded  raw  binary  GraphQL

KEY	VALUE
<input checked="" type="checkbox"/>	UserData.tgz <span>×</span>

Body Cookies Headers (5) Test Results

Pretty Raw Preview Visualize JSON ▼ ⌵

```

1  {
2    "retcode": 0,
3    "action": "import",
4    "message": "OK"
5  }

```


In X915, if type is UserData, it will return after importing finished.  
 In S539, it will return immediately after file uploaded. And then we need access [GET /api/user/status?type=exportImportUser](#) to monitor importing status. More details about [/api/user/status?type=exportImportUser](#) see [api user status](#)  
 Finally, access request [/api/commontool/set](#) to notify device that The front-end has already processed this import result.

```

{
  "target": "commontool",
  "action": "set",
  "data": {
    "action": "ResetUserImportStatus"
  }
}

```

## api user status

Request Param	Description
type	<p>Empty or request without this parameter get card code swiping.</p>  <p><b>importFace</b> (Only in X915) is importing face data background.  <b>exportImportUser</b> (Only in S539)            Is importing or exporting user data background</p>

Return Param	Description
CardCode	Card code swiping
importFaceRet (type = importFace)	1 import finished 0 importing -1 not importing face data
action (Only in S539, type=exportImportUser)	export preparing file for exporting background import importing user data background
Step (Only in S539, type=exportImportUser)	action=export 0 not exporting, means normal 1 export failed 2 prepare user data from db 3 writing user data to file 4 copy face data 5 compress file 6 prepared OK.
	action=import 0 unzip file 1 save schedule to db 2 compare user data with db 3 save user data to db 4 save face data to db 5 not importing 6 import failed 7 import successfully 8 start importing
total (Only in S539, type=exportImportUser)	Total progress in step current
cur (Only in S539, type=exportImportUser)	Cur progress in step current

errCode (Only in S539, type=exportImport User)	action=export	useless
	action=import	0 normal, device is not importing or exporting data 1 import successfully 2 file not found in .tgz file 3 schedule reaches max 4 user reaches max 5 failed to open db
url (Only in S539, type=exportImportUser)		Url to download user .tgz file

For example, in S539

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

- Method:** GET
- URL:** http://192.168.1.21/api/user/status?type=exportImportUser
- Body Type:** none
- Response Body (JSON):**

```

1  {
2    "retcode": 0,
3    "action": "status",
4    "message": "OK",
5    "data": {
6      "cur": 0,
7      "errCode": 0,
8      "step": 5,
9      "total": 0,
10     "action": "import"
11   }
12 }
```

Action is import and step is 5 means device is not importing or exporting data.

GET http://192.168.1.72/api/user/status

Params Authorization Headers (6) **Body** Pre-request Script

none  form-data  x-www-form-urlencoded  raw  binary

---

Body Cookies Headers (5) Test Results

Pretty Raw Preview Visualize JSON

```
1 {
2   "retcode": 0,
3   "action": "status",
4   "message": "OK",
5   "data": {
6     "CardCode": "7CE2CE70"
7   }
8 }
```

GET http://192.168.1.21/api/user/status

Params Authorization Headers (6) **Body** Pre-request Script

none  form-data  x-www-form-urlencoded  raw  binary

---

Body Cookies Headers (5) Test Results

Pretty Raw Preview Visualize JSON

```
1 {
2   "retcode": -1,
3   "action": "status",
4   "message": "Failed",
5   "data": {
6     "status": -1,
7     "CardCode": ""
8   }
9 }
```

# api commontool

Only S539 and X915 support import.

## api commontool get

Request Param	Description
config_name	DirectIPPort      Direct IP Port DoorOnvifServer   onvif server PowerType          0 =>POE OR 1=>12V

### 1.DirectIPPort

The screenshot shows a web interface for configuring 'Direct IP'. The 'Enabled' checkbox is checked. The 'Port' field is set to 5060. Below this is a REST client interface showing a GET request to `http://192.168.1.72/api/commontool/get?config_name=DirectIPPort`. The response body is shown in JSON format:

```
1  {
2    "retcode": 0,
3    "action": "get",
4    "message": "OK",
5    "data": {
6      "DirectIPPort": 5060
7    }
8 }
```

### 2. DoorOnvifServer

Surveillance» ONVIF

### Basic Setting

Discoverable	<input checked="" type="checkbox"/>
User Name	<input type="text" value="admin"/>
Password	<input type="password" value="*****"/>

### Advanced Setting

Milestone	<input type="checkbox"/>
-----------	--------------------------

GET ▼ http://192.168.1.72/api/commontool/get?config\_name=DoorOnvifServer

Params ● Authorization Headers (6) **Body** Pre-request Script Tests Settings

● none ● form-data ● x-www-form-urlencoded ● raw ● binary ● GraphQL

Body Cookies Headers (5) Test Results

Pretty Raw Preview Visualize JSON ▼

```
1  {
2    "retcode": 0,
3    "action": "get",
4    "message": "OK",
5    "data": {
6      "DeviceMode": 0,
7      "DeviceUser": "admin",
8      "MilestoneEnabled": 0
9    }
10 }
```

### 3. PowerType



GET ▼ http://192.168.1.72/api/commontool/get?config\_name=PowerType

Params ● Authorization Headers (6) **Body** Pre-request Script Tests Settings

none  form-data  x-www-form-urlencoded  raw  binary  GraphQL

---

Body Cookies Headers (5) Test Results

Pretty Raw Preview Visualize JSON ▼

```

1  {
2    "retcode": 0,
3    "action": "get",
4    "message": "OK",
5    "data": {
6      "PowerType": 0
7    }
8  }

```

## api commontool set

Request Param	Description	
config_name	DoorOnvifServer	DeviceMode discoverable MilestoneEnabled milestone enable
	MicMode	Switch mic mode 1 left mic 0 right mic
action	config_name not given	ResetUserImportStatus ResetUserExportStatus

Set milestone enable

```

{
  "target": "commontool",
  "action": "set",
  "data": {
    "MilestoneEnabled": "1",
    "config_name": "DoorOnvifServer"
  }
}

```

After set config Config.Settings.HANDFREE.MicMode by api/config/set, we need to access request as follow to set mic mode to right mic immediately.

```
{
  "target":"commontool",
  "action":"set",
  "data":{
    "value":"1",
    "config_name":"MicMode"
  }
}
```

# api filetool







Only S539 and X915 support filetool.

## api filetool import

Request Param	Description																											
destFile	Screensaver	<p>Screen saver img. index value between 1 to 5.</p> <p>Upload Screensaver</p> <table border="1"> <thead> <tr> <th>Screensaver ID</th> <th>File Status</th> <th>Interval(Sec)</th> <th>Submit</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>File Exists</td> <td><input type="text" value="5"/></td> <td><input type="button" value="Submit"/></td> </tr> <tr> <td>2</td> <td>File Exists</td> <td><input type="text" value="5"/></td> <td><input type="button" value="Submit"/></td> </tr> <tr> <td>3</td> <td>File Exists</td> <td><input type="text" value="5"/></td> <td><input type="button" value="Submit"/></td> </tr> <tr> <td>4</td> <td>File Exists</td> <td><input type="text" value="5"/></td> <td><input type="button" value="Submit"/></td> </tr> <tr> <td>5</td> <td>File Exists</td> <td><input type="text" value="5"/></td> <td><input type="button" value="Submit"/></td> </tr> </tbody> </table>	Screensaver ID	File Status	Interval(Sec)	Submit	1	File Exists	<input type="text" value="5"/>	<input type="button" value="Submit"/>	2	File Exists	<input type="text" value="5"/>	<input type="button" value="Submit"/>	3	File Exists	<input type="text" value="5"/>	<input type="button" value="Submit"/>	4	File Exists	<input type="text" value="5"/>	<input type="button" value="Submit"/>	5	File Exists	<input type="text" value="5"/>	<input type="button" value="Submit"/>		
	Screensaver ID	File Status	Interval(Sec)	Submit																								
	1	File Exists	<input type="text" value="5"/>	<input type="button" value="Submit"/>																								
	2	File Exists	<input type="text" value="5"/>	<input type="button" value="Submit"/>																								
	3	File Exists	<input type="text" value="5"/>	<input type="button" value="Submit"/>																								
	4	File Exists	<input type="text" value="5"/>	<input type="button" value="Submit"/>																								
	5	File Exists	<input type="text" value="5"/>	<input type="button" value="Submit"/>																								
VoicePrompt	<p>Prompt voice file. index value between 0 to 7 0-greeting 1-access granted ...</p> <p>Voice Prompt Setting</p> <table border="1"> <thead> <tr> <th>ID</th> <th>Tone</th> <th>Import</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>Greetings</td> <td><input type="button" value="Import"/></td> </tr> <tr> <td>2</td> <td>Access Granted</td> <td><input type="button" value="Import"/></td> </tr> <tr> <td>3</td> <td>Access Denied</td> <td><input type="button" value="Import"/></td> </tr> <tr> <td>4</td> <td>PIN Page</td> <td><input type="button" value="Import"/></td> </tr> <tr> <td>5</td> <td>APT+PIN</td> <td><input type="button" value="Import"/></td> </tr> <tr> <td>6</td> <td>Call Page</td> <td><input type="button" value="Import"/></td> </tr> <tr> <td>7</td> <td>Calling</td> <td><input type="button" value="Import"/></td> </tr> <tr> <td>8</td> <td>Directory</td> <td><input type="button" value="Import"/></td> </tr> </tbody> </table>	ID	Tone	Import	1	Greetings	<input type="button" value="Import"/>	2	Access Granted	<input type="button" value="Import"/>	3	Access Denied	<input type="button" value="Import"/>	4	PIN Page	<input type="button" value="Import"/>	5	APT+PIN	<input type="button" value="Import"/>	6	Call Page	<input type="button" value="Import"/>	7	Calling	<input type="button" value="Import"/>	8	Directory	<input type="button" value="Import"/>
ID	Tone	Import																										
1	Greetings	<input type="button" value="Import"/>																										
2	Access Granted	<input type="button" value="Import"/>																										
3	Access Denied	<input type="button" value="Import"/>																										
4	PIN Page	<input type="button" value="Import"/>																										
5	APT+PIN	<input type="button" value="Import"/>																										
6	Call Page	<input type="button" value="Import"/>																										
7	Calling	<input type="button" value="Import"/>																										
8	Directory	<input type="button" value="Import"/>																										
BackgroundOfTimeView	<p>Background of timeView</p> <p>Background of Dial Tips(.png) <input type="button" value="Import"/> <input type="button" value="Reset"/></p>																											
Face	<p>Upload file to scan face.details in <b><u>api user add</u></b></p>																											
BootAnimation	<p>Boot animation</p> <p>Boot Animation (.png / .zip) <input type="button" value="Import"/> <input type="button" value="Reset"/></p>																											
BackgroundOfContact	<p>Background of Contact</p> <p>Background of Directory List(.png) <input type="button" value="Import"/> <input type="button" value="Reset"/></p>																											
index	destFile=Screensave																											

	r or destFile=VoicePrompt	
--	------------------------------	--

## api filetool del

Request Param	Description	
destFile	Screensaver	Delete screen saver img. index value between 1 to 5.
	VoicePrompt	Delete uploaded prompt voice file. index value between 0 to 7 0-greeting 1-access granted ...
	BackgroundOfTimeView	Reset background of timeView  Background of Dial Tips(.png)  
	BootAnimation	Reset boot animation  Boot Animation (.png / .zip)  
	BackgroundOfContact	Reset background of Contact  Background of Directory List(.png)  
index	destFile=Screensaver or destFile=VoicePrompt	