IR600 and IR300 series router HTTP API

Updated April 28, 2021
Version 1.7
Author: Zeming Wang wangzm@inhand.com.cn

1. Function

With webUI, end user can view/set the parameters of IR600 & IR300 series routers. And with HTTP API, end user can have software to communicate with IR600 & IR300 routers to view/set the parameters. End user can control it remotely without login, but still username and password is required.

2. Instruction

HTTP API is the same as web access. HTTP POST and GET will work. With POST and GET, end user software sends the requests to IR600 & IR300 series router and the router will response.

The default port for this HTTP API is 4444, and the username and password are required in the request.

2.1 Get the parameters from IR600 & IR300 series router.

With this API, end user can get the parameters and status of IR600 & IR300 routes.

The request context is “getinfo.cgi”
The value for this request are in the Appendix 1
Format is：
“<param>”
If there are multiple param in one request, please use ‘&’ to separate them:
e.g. “<param1>&<param2>”
The router will response as below when it is succeed.
<param1>=<value1>\r\n<param2>=<value2>\r\n
The router will response with：”Command bad” when it is failed.

In IR600/IR300 router, there are two types of data can be read: parameters and status.

Parameters are the value user can configure which start with letter ‘a’ – ‘z’, such as the LAN IP address ‘lan0_ip’.

The status is read only, such as the IP address which is assigned by carrier to cellular interface. Status are start with ‘_’. For example, the IP address of cellular interface can’t be modified by end user, so it start with ‘_’, which is ‘_wan1_get_ip’;

Note:

To be compliant with old customer system, it supports some special parameters to support the legacy system as well. Here is the list:

- signal_info // get the signal
  - signal_status=xx\r\n //the signal range is 0 - 31
- sim_ip_info // get the cellular IP address. If the cellular is down, router will reply with 0
  - ip=xx\r\n
- iccid_info // get the ICCID
  - iccid=xx\r\n
- serials_number_info //get the Serials Number
  - serials_number=xx\r\n
- basic_info //get the basic info from router
  - hd_serials_number=xxx\r\n //serials number
  - hard_version=xxx\r\n //hardware version
  - software_version_tail=xxx\r\n //software version
- modem_info // geht the cellular module infor
  - online_modem_name=xx\r\n
  - tmp_time=xx\r\n
  - net_status=xx\r\n
  - nettype=xx\r\n
  - signal_status=xx\r\n
  - ip=%s\r\ndns=xx\r\n
  - sim_status=xx\r\n
  - cardflag=xx\r\n
  - iccid=xx\r\n
  - imsi=xx\r\n
  - lac_num=xx\r\n
  - cell_id_num=xx\r\n
  - rsrp=%s\r\nsinr=xx\r\n
- reboot //reboot the IR600/IR300 router. No response from IR600/IR300 router

2.2 Set parameters to IR600/IR300 routes

The request context is “apply.cgi”

Format is：

“<param>=<vaule>”

<param> is the name of the parameter

<value> is the value of the parameter should be set to IR600/IR300

If there are multiple param in one setting, please use ‘&’ to separate them:

e.g. “<param1>=<value1>&<param2>=<value2>&<param3>=<value3>”

Note: to apply the parameters after setting request, IR600/IR300 router needs to be reboot or restart the service.

When the <param> is "_service", which means control the services. The allowed value for a service are:

"_service=XXX-restart" the XXX service will be restarted

"_service=XXXX-stop" the XXX service will be stop

"_service=XXX-start" the XXX service will be start

IR600/IR300 supports multiple service operations at the same setting request. For example "_service=XXX-restart,YYY-restart"

"_service" can be set with other <param> for example:

"<param1>=<value1>&<param2>=<value2>&_service=XXX-restart,YYY-restart"

The router will return "SUCCESS" when it is succeed, or "FAIL" when it is failed.

2.3 Download the configure file from IR600/IR300 router

The request context is "download.cgi?type=config"

The router will return "SUCCESS" when it is succeed, or "FAIL" when it is failed.

2.4 Upload configure file to IR600/IR300 router

The request context is "upload.cgi?type=config"

The router will return "SUCCESS" when it is succeed, or "FAIL" when it is failed.

2.5 Upgrade IR600/IR300 router

The request context is "upgrade.cgi"

The router will return "SUCCESS" when it is succeed, or "FAIL" when it is failed.

2.6 IR600/IR300 router Actions

The router can execute some actions which takes some time, such as ping test. All actions need two calls: action_start, action_abort, action_result.

action_start can initiate an action

action_abort can abort an action

action_result can query the result of an action

Note: Only one action type can be executed at any time. If a second action_start is received,

the first one will abort.

A action_start is required before action_abort and action_result.

Context:

The action_start request context is “apply.cgi? action_start=*action_type*[*&paramters=x[…]*]”

The action_abort request context is “apply.cgi? action_abort=*action_type*”

The action_ result request context is “apply.cgi? action_ result =*action_type*”

Type:

action_type include **PING_Test**.

PING Test is the action to evaluate the network quality,

PING Test needs 3 parameters:

- dst: Server for PING action. IP address or domain name. Only ‘0’-‘9’, ‘a’-‘z’,‘A’-‘Z’ and ‘.’ Allowed. If no such parameter, 8.8.8.8 will be applied
- time: unit is second. Range: 4 - 3600 second. The for time PING Test. If no such parameter, 300 seconds will be applied
- size: unit is bytes. Range: 64 – 1024. The size of PING packet. If no such parameter, 64 bytes will be applied.

Response:

The router will return “STARTED”, or “FAILED” in response of action_start,

The router will return “ABORTED” in response of action_abort. “NOT STARTED” will be returned if there is no action_start.

The router will return “DONE” or “PENDING” in response of action_result. “NOT STARTED” will be returned if there is no action_start.

Additional context in response:

PING Test:

In action_start, if FAILED is in the response, a msg will show the reason, such as:

FAILED\r\n*msg=01 Internet is down*\r\n

Or *FAILED*\r\n*msg=02 Parameter xxx is out of range. Current range is xxxx*\r\n

Or *FAILED*\r\n*msg=03 System errorxxxx*\r\n

In action_result, if “DONE” is in the response, a msg will show the result. The format is

DONE\r\n

result-time=YYYY/MM/DD HH:MM:SS\r\n

ping-host=xxxx.com\r\n

ping-size=128\r\n

packets-transmitted=4\r\n

packets-received=4\r\n

packet-loss-rate=0.0%\r\n

round-trip-min=16.528\r\n

round-trip-avg=19.854\r\n

round-trip-max=29.122\r\n

Note: ping-size unit is byte. Round-trip-min, round-trip-avg, round-trip-max unit is mini second.

2.7 User Defined Parameters

In some cases, 3rd party software/devices want to save a simple string to router and then share to local devices, InHand implement the User Defined Parameters. With these parameters, 3rd party can save/read/erase up to 100*256 bytes characters to InHand router temperately (for one hour). All such parameter will be deleted/purged in 60 minutes.

The parameter name is: user_define_00 to user_define_99.

- To Save/add/change some value to router:

curl -u adm:123456 "*192.168.2.1:4444*/apply.cgi" -d "user_define_*00*=*uservalue*" -H "Content-Type:text/plain"

- To read such saved value:

curl -u adm:123456 "*192.168.2.1:4444*/getinfo.cgi" -d "user_define_00&userdefine_99" -H "Content-Type:text/plain"

it will reply:

user_define_00=*uservalue00*\r\n

user_define_99=*uservalue99*\r\n

Note:

1. The *uservalue* must be string, such as 'a'-'z', 'A'-'Z','0'-'9'. The length should 1 byte to 256 byte.
2. All these 100 parameters are save on NVM.
3. Every value will be deleted in 60 minutes after it is set.

2.8 Auto Ping function

Some customers require that InHand router needs to be able to ping automatically to test the network packet loss rate. To have this feature, InHand team developed the Auto Ping function.

Regarding this feature, there are 4 parameters involved:

- 'auto_ping_enable' should be set to 1. 0 means disable.
- auto_ping_dst: this is the destination for Auto Ping function. ~~The valid value must be '0' – '9', 'a' -'z','A'-'Z', and '.'.~~

  Note: The value can be any printable character. However, please be noted that, some special characters are meaningless, such as "*", "?", "~", etc. These special characters may casue system error. For you application, please don't try to set these special characters to InHand router.

  If there is no setting on parameter, the router will ping 8.8.8.8
- auto_ping_times: the time auto ping will do. The value must be 1-50. If not setting, the default is 20.

  Note: if it was configured, the previous value will be used.
- auto_ping_size: the packet size for Auto Ping. The range is 64-1024. If it is <64, the router will ping with 64 bytes.

  When configuring this parameter via HTTP API command, InHand software will verify its

value. If this parameter is out of range, the response will be FAIL.

When configuring this parameter via import configuration file, InHand software can't verify its value, so if it is greater than 1024, the router will ping with 1024 bytes. If not setting, the router will ping with 64 bytes.

To read the Auto ping result, 3rd party software

The action_ result request context is "apply.cgi? action_ result = **PING_Test**". And the result format is the same as 2.6:

DONE\r\n
result-time=YYYY/MM/DD HH:MM:SS\r\n
ping-host=xxxx.com\r\n
ping-size=128\r\n
packets-transmitted=4\r\n
packets-received=4\r\n
packet-loss-rate=0.0%\r\n
round-trip-min=16.528\r\n
round-trip-avg=19.854\r\n
round-trip-max=29.122\r\n

Note: ping-size unit is byte. Round-trip-min, round-trip-avg, round-trip-max unit is mini second.

Note: Changing any "Auto Ping" parameter will NOT restart the Auto Ping server immediately. If you want to check the new Auto Ping result after new parameter is configured, please reboot router.

2.9 Get the device list from IR600/IR300 series router.

With this API, end user can get the device list of IR600/IR300 routes.
The request context is "update.cgi"
Format is：

- curl -u adm:123456 192.168.2.1:4444/update.cgi -d "exec=devlist" -H "Content-Type:text/plain"

The router will response as below when it is succeed.
Arplist = [['ipaddress','mac','network port'],['…','…','…'], ……];\r\n

2.10 Digital I/O functions **(only applicable to IR300 series with I/O)**

With this API, end user can use digital I/O functions, including: configure mode, set level, get status.
Configure I/O 1 as input mode:

- curl -u adm:password 192.168.2.1:4444/iodigital.cgi -d "type=config&io=1&mode=in"
- Successful execution returns "SUCCESS"

Configure I/O 1 as output mode:

- curl -u adm:password 192.168.2.1:4444/iodigital.cgi -d "type=config&io=1&mode=out"

- Successful execution returns “SUCCESS”

Get the current status of I/O 1:

- curl -u adm:password 192.168.2.1:4444/iodigital.cgi -d "type=get&io=1"
- Successful execution returns “io_level=high\r\n” or “io_level=low\r\n”

Configure the output level of I/O 1 to be low:

- curl -u adm:password 192.168.2.1:4444/iodigital.cgi -d "type=set&io=1&setlevel=low"
- Successful execution returns “SUCCESS”

Configure the output level of I/O 1 to be high:

- curl -u adm:password 192.168.2.1:4444/iodigital.cgi -d "type=set&io=1&setlevel=high"
- Successful execution returns “SUCCESS”

Note: The operation method of I/O 2 is the same as above, just change io=1 to io=2.

“Bad Command” will pop up if the parameters are wrong.

3. Examples

I tested with:

MacOS

IR611 username and password: adm/123456

IR611 IP address: 192.168.2.1

http tools: curl

1. **Get the parameter and status from IR600/IR300 router**

   Method 1: Get the IP and cellular signal strength by standard parameters

   curl -u adm:123456 192.168.1.14:4444/getinfo.cgi -d “_wan1_get_ip&_wan1_siglevel” -H “Content-Type:text/plain”

   Reply from IR611:

   _wan1_get_ip=10.131.151.210

   _wan1_siglevel=24

   Method 2: Get the IP and cellular signal strength by lagecy parameters

   curl -u adm:123456 192.168.2.1:4444/getinfo.cgi -d “modem_info”

   online_modem_name=usb0

   tmp_time=4420 seconds

   net_status=connected

   nettype=4G

   signal_status=24

   ip=10.131.151.210

   dns=198.224.191.135

   sim_status=sim1

   cardflag=ok

   imsi=311480148857204

   iccid=89148000001470640300

   lac_num=29953

   cell_id_num=29989377

rsrp=

sinr=

2. **Set the parameters of IR600/IR300 router**

e.g. enable Wi-Fi interface

curl -u adm:123456 “192.168.2.1:4444/apply.cgi” -d “wl0_enable=1” -H “Content-Type:text/plain”

e.g. restart Wi-Fi service to apply the setting

curl -u adm:123456 “192.168.2.1:4444/apply.cgi” -d “wl0_enable=1&_service=wl0-restart,an0-restart,dhcpd-restart” -H “Content-Type:text/plain”

e.g. restart the Status Update service to have router upload status immediately

curl -u adm:123456 “192.168.2.1:4444/apply.cgi” -d “rmonthird-restart” -H “Content-Type:text/plain”

3. **Download the configuration file from IR600/IR300 router**

curl -u adm:123456 -o config.dat “192.168.2.1:4444/download.cgi?type=config”

Note: -o is the output of the **configuration** file. Curl will save the **configuration** file to your local computer.

4. **Upload configuration file to IR600/IR300 router**

curl -u adm:123456 -F “filename=@config.dat” “192.168.2.1:4444/upload.cgi?type=config”

Note:

1. -F the new configure file, name is “config.dat”. It must be on the folder which curl is going to use.

2. Please reboot router after upload **configuration** file to apply it.

5. **Upgrade IR600/IR300 router**

curl -u adm:123456 -s -F upload=@router.bin http://192.168.2.1:4444/upgrade.cgi

Note:

1. -F the firmware file, name is “router.bin”. It must be on the folder which curl is going to use.

2. Please reboot router after upload firmware file to apply it.

6. Reboot the router

curl -u adm:123456 192.168.2.1:4444/getinfo.cgi -d “reboot”

Note: after rebooting command, the response is empty.

7. PING Test

Start a PING Test

curl -u adm:123456 192.168.2.1:4444/ apply.cgi -d “action_start= PING_Test*&dst=8.8.8.8&time=600&size=128*”

STARTED

Abort the PING Test
curl -u adm:123456 192.168.2.1:4444/ apply.cgi -d “action_abort= PING_Test”
ABORTED

Check the result of PING Test when it is not done yet
curl -u adm:123456 192.168.2.1:4444/ apply.cgi -d “action_result= PING_Test”
PENDING

Check the result of PING Test when it is done
curl -u adm:123456 192.168.2.1:4444/ apply.cgi -d “action_result= PING_Test”
DONE\r\n
packets-transmitted=600\r\n
packets-received=590\r\n
packet-loss-rate=1.67%\r\n
round-trip-min=160.528\r\n
round-trip-avg=190.854\r\n
round-trip-max=290.122\r\n

History

Revision	Date	Author	Note
1.0	May 9th, 2019	Zeming Wang	Initial revision
1.1	May 15, 2019	Zeming Wang	Added PING test API
1.2	May 16, 2019	Zeiming Wang	Add a note in section 2.6 Action
1.3	May 28, 2019	Zeming Wang	Fixed some bug in PING TEST Add "result-time", "ping-host", "ping-size" in PING TEST
1.4	August 16, 2019	Zeming Wang	1. Add user_define_0 to user_define_99 usage 2. Add auto PING function
1.5	December 30,2020	Niko Huang	Add device list API
1.6	January 13,2021	Niko Huang	Add IR300 digital I/O API
1.7	April 8, 2022	Cailong Wu	Proofread