acos httpd JSON api format

Jeff Chiang

April 9, 2024

1 Introduction

In this document show the JSON API for acos httpd use

2 Usage

GET **/GetDevName.cgi**

get device name

Parameter

device_name	router name
pre_mode	current router mode

Response application/json

200 ok

```
{
  "device_name": "IndoorV2-test",
  "pre_mode": "wifi_router"
}
```

POST **/SetDevName.cgi**

device name setting

Parameter

device_name	device name

Body application/json

```
{
  "device_name": "IndoorV2-test"
}
```

Response application/json

200 ok

```
{
    "result:200
}
```

GET **/GetHotspot.cgi**

get hotspot info

Parameter

hotspot_3gpplist	hotspot list
hotspot_id	hotspot id code
hotspot_ip	hotspot ip address
hotspot_key	hotspot password
hotspot_ssid	wlan setting

Response application/json

200 ok

```
{
  "hotspot_ssid": "hostpot-test",
  "hotspot_ip": "12.34.56.78",
  "hotspot_id": "12345678",
  "hotspot_key": "password",
  "hotspot_3gpplist": "Unknown"
}
```

POST **/SetHotspot.cgi**

set hotspot config

Parameter

hotspot_3gpplist	hotspot key
hotspot_id	hotspot id
hotspot_ip	setting hotspot ip address
hotspot_key	hotspot key
hotspot_ssid	setting hotspot ssid

Body application/json

```
{
  "hotspot_3gpplist": "Unknown",
  "hotspot_id": "12345678",
  "hotspot_ip": "12.34.56.78",
  "hotspot_key": "password",
  "hotspot_ssid": "hostpot-test"
}
```

Response application/json

200 ok

```
{
    "result:200
}
```

GET **/GetHotSpotList.cgi**

null

Parameter

count	TBD
hspot_tbl	TBD

Response application/json

200 ok


```
{
  "count": 2,
  "hspot_tbl": [
    {
      "id": 1,
      "mcc": "192.168.1.2",
      "mnc": "dev1",
      "action": "activate"
    },
    {
      "id": 2,
      "mcc": "192.168.1.3",
      "mnc": "dev2",
      "action": "block"
    }
  ]
}
```

POST **/SetHotSpotList.cgi**

null

Parameter

count	TBD
hspot_tbl	TBD

Body application/json

```
{
  "count": 2,
  "hspot_tbl": [
    {
      "action": "activate",
      "id": 1,
      "mcc": "192.168.1.2",
      "mnc": "dev1"
    },
    {
      "action": "block",
      "id": 2,
      "mcc": "192.168.1.3",
      "mnc": "dev2"
    }
  ]
}
```

Response application/json

200 ok

```
{
    "result:200
}
```

GET **/GetIpv6Mode.cgi**

get hotspot info

Parameter

ipv6_dns	ipv6 mandory dns
ipv6_dns2	ipv6 secondary dns
ipv6_eanble	enable ipv6 or not
ipv6_id	ipv6 id
ipv6_lanAddr	ipv6 lan ip address
ipv6_lanMode	ipv6 lan ip address
ipv6_type	ipv6 type
ipv6_wanMode	ipv6 wan (dynamic or static)
ipv6_wanipaddr	ipv6 wan ip

Response application/json

200 ok


```
{
  "ipv6_dns": "unknown",
  "ipv6_dns2": "unknown",
  "ipv6_eanble": true,
  "ipv6_id": "unknown",
  "ipv6_lanAddr": "unknown",
  "ipv6_lanMode": "dynammic",
  "ipv6_type": "disable",
  "ipv6_wanMode": "dynammic",
  "ipv6_wanipaddr": "unknown"
}
```

POST **/SetIpv6Mode.cgi**

set hotspot config

Parameter

ipv6_dns	ipv6 mandory dns
ipv6_dns2	ipv6 secondary dns
ipv6_eanble	enable ipv6 or not
ipv6_id	ipv6 id
ipv6_lanAddr	ipv6 lan ip address
ipv6_lanMode	ipv6 lan (dynamic or staitc)
ipv6_type	ipv6 type
ipv6_wanMode	ipv6 wan (dynamic or static)
ipv6_wanipaddr	ipv6 wan ip

Body application/json

```
{
  "ipv6_dns": "unknown",
  "ipv6_dns2": "unknown",
  "ipv6_eanble": true,
  "ipv6_id": "unknown",
  "ipv6_lanAddr": "unknown",
  "ipv6_lanMode": "dynammic",
  "ipv6_type": "disable",
  "ipv6_wanMode": "dynammic",
  "ipv6_wanipaddr": "unknown"
}
```

Response application/json

200 ok

```
{
    "result:200
}
```

GET **/GetLanReserv.cgi**

get hotspot info

Parameter

count	wlan setting
lan_Reservtbl	wlan setting

Response application/json

200 ok


```
{
  "count": 3,
  "lan_Reservtbl": [
    {
      "id": 1,
      "ipaddr": "192.168.1.2",
      "devname": "dev1",
      "mac": "A1:B2:C3:D4:E5:F1",
      "action": "block"
    },
    {
      "id": 2,
      "ipaddr": "192.168.1.3",
      "devname": "dev3",
      "mac": "A1:B2:C3:D4:E5:F3",
      "action": "allow"
    },
    {
      "id": 3,
      "ipaddr": "192.168.1.4",
      "devname": "dev4",
      "mac": "A1:B2:C3:D4:E5:F4",
      "action": "TBD"
    }
  ]
}
```

POST **/SetLanReserv.cgi**

set hotspot config

Parameter

count	null
lan_Reservtbl	null

Body application/json


```
{
  "count": 3,
  "lan_Reservtbl": [
    {
      "action": "block",
      "devname": "dev1",
      "id": 1,
      "ipaddr": "192.168.1.2",
      "mac": "A1:B2:C3:D4:E5:F1"
    },
    {
      "action": "allow",
      "devname": "dev3",
      "id": 2,
      "ipaddr": "192.168.1.3",
      "mac": "A1:B2:C3:D4:E5:F3"
    },
    {
      "action": "TBD",
      "devname": "dev4",
      "id": 3,
      "ipaddr": "192.168.1.4",
      "mac": "A1:B2:C3:D4:E5:F4"
    }
  ]
}
```


Response application/json

200 ok


```
{
    "result:200
}
```

GET **/GetPortForwarding.cgi**

show port fowarding / triggering rule setting and config

Parameter

lan_ip	port forwarding translation LAN IP
pf_enable	port forwarding function on/off
pf_table	port forwarding rule table (id,Name,exPort,inPort,inAddr)
pt_enable	port triggering function on / off
pt_table	port triggering rule table (id,Name,exPort,inPort,inAddr)

Response application/json

200 ok


```
{
  "lan_ip": "192.168.1.1",
  "pf_enable": true,
  "pf_table": [
    {
      "id": 1,
      "Name": "tes1",
      "exPort": 11,
      "inPort": 111,
      "ipAddr": "192.168.1.1"
    },
    {
      "id": 2,
      "Name": "tes2",
      "exPort": 12,
      "inPort": 112,
      "ipAddr": "192.168.1.2"
    },
    {
      "id": 3,
      "Name": "tes3",
      "exPort": 13,
      "inPort": 113,
      "ipAddr": "192.168.1.3"
    },
    {
      "id": 4,
      "Name": "tes4",
      "exPort": 14,
      "inPort": 114,
      "ipAddr": "192.168.1.4"
    },
    {
      "id": 5,
      "Name": "tes5",
      "exPort": 15,
      "inPort": 115,
      "ipAddr": "192.168.1.5"
    }
  ]
```

POST **/SetPortForwarding.cgi**

setting each port forwardding / triggering rule and setting

Parameter

pf_enable	null
pf_table	null
pt_enable	null
pt_table	null

Body application/json


```
{
  "pf_enable": true,
  "pf_table": [
    {
      "Name": "tes1",
      "exPort": 11,
      "id": 1,
      "inPort": 111,
      "ipAddr": "192.168.1.1"
    },
    {
      "Name": "tes2",
      "exPort": 12,
      "id": 2,
      "inPort": 112,
      "ipAddr": "192.168.1.2"
    },
    {
      "Name": "tes3",
      "exPort": 13,
      "id": 3,
      "inPort": 113,
      "ipAddr": "192.168.1.3"
    },
    {
      "Name": "tes4",
      "exPort": 14,
      "id": 4,
      "inPort": 114,
      "ipAddr": "192.168.1.4"
    },
    {
      "Name": "tes5",
      "exPort": 15,
      "id": 5,
      "inPort": 115,
      "ipAddr": "192.168.1.5"
    }
  ],
  "pt_enable": true,
```

GET **/GetRouterMode.cgi**

get current router mode and wan ip addres info

Parameter

device_name	device name len<64
enable_ap_mode	router mode setting (AP/router)
ifconfig	router WAN network info

Response application/json

200 ok


```
{
  "device_name": "IndoorV2-test",
  "enable_ap_mode": "0",
  "ifconfig": {
    "ipaddr": "192.168.7.128",
    "netmask": "255.255.255.0",
    "geteway": "192.168.7.151",
    "dns1_pri": ""
  }
}
```


POST **/SetRouterMode.cgi**

set router mode

Parameter

router_mode	setting router mode setting (AP/router)

Body application/json


```
{
  "router_mode": "router"
}
```


Response application/json

200 ok


```
{
    "result:200
}
```

GET **/GetUpnp.cgi**

get upnp related info and upnp portforwarding table

Parameter

PortMapTable	upnp port mapping table
hiddenAdverTime	upnp advertise time
hiddenTimeToLive	upnp time to live
hiddenTurnUPnPOn	enable/disable upnp

Response application/json

200 ok

```
{
  "hiddenAdverTime": 111,
  "hiddenTimeToLive": 123,
  "hiddenTurnUPnPOn": true
}
```

POST **/SetUpnp.cgi**

set upnp related config

Parameter

hiddenAdverTime	setting upnp advertise time
hiddenTimeToLive	setting upnp time to live
hiddenTurnUPnPOn	setting upnp on / off time

Body application/json

```
{
  "hiddenAdverTime": 111,
  "hiddenTimeToLive": 123,
  "hiddenTurnUPnPOn": true
}
```

Response application/json

200 ok

```
{
    "result:200
}
```

GET **/GetVer.cgi**

Get firmeare version

Parameter

SOFTWARE_VERSION	Main version
UI_VERSION	sub version

Response application/json

200 ok

```
{
  "SOFTWARE_VERSION": "V1.0.0.1",
  "UI_VERSION": "1.0.1"
}
```

GET **/GetVlanBridge.cgi**

Get vlan info

Parameter

bridgePort	enable/disable vlan
lan_portcount	enable/disable vlan
vlan_enable	enable/disable vlan
vlan_group	enable/disable vlan
vlan_mode	enable/disable vlan
wirebandCount	enable/disable vlan

Response application/json

200 ok


```
{
  "bridgePort": [
    {
      "id": 0,
      "type": "",
      "Enable": false
    },
    {
      "id": 0,
      "type": "",
      "Enable": false
    },
    {
      "id": 0,
      "type": "",
      "Enable": false
    },
    {
      "id": 0,
      "type": "",
      "Enable": false
    },
    {
      "id": 0,
      "type": "",
      "Enable": false
    }
  ],
  "lan_portcount": 4,
  "vlan_enable": true,
  "vlan_group": [
    {
      "Enable": true,
      "Name": "vlan_10",
      "vlan_id": 0,
      "Priority": 0,
      "wieredPort": "1,2",
      "wierelessBnad": "",
```

POST **/SetVlanBridge.cgi**

Set vlan config

Parameter

bridgePort	enable/disable vlan
lan_portcount	enable/disable vlan
vlan_enable	enable/disable vlan
vlan_group	enable/disable vlan
vlan_mode	enable/disable vlan
wirebandCount	enable/disable vlan

Body application/json


```
{
  "bridgePort": [
    {
      "Enable": true,
      "id": 1,
      "type": "wired"
    },
    {
      "Enable": false,
      "id": 2,
      "type": "wired"
    },
    {
      "Enable": true,
      "id": 3,
      "type": "wired"
    },
    {
      "Enable": true,
      "id": 1,
      "type": "wireless"
    },
    {
      "Enable": true,
      "id": 2,
      "type": "wireless"
    }
  ],
  "lan_portcount": 4,
  "vlan_enable": true,
  "vlan_group": [
    {
      "Enable": true,
      "Name": "vlan_10",
      "Priority": 8,
      "action": "active",
      "vlan_id": 10,
      "wieredPort": "1,2",
```

GET **/GetWanAdv.cgi**

get WAN / firewall advance setting

Parameter

NatInboundFiltering	setting dmz enable /disable
disable_igmp	setting dmz enable /disable
disable_sip	setting dmz enable /disable
dmz_enable	setting dmz enable /disable
dmz_ipaddr	setting dmz enable /disable
response_ping	setting dmz enable /disable
wan_mtu	setting dmz enable /disable

Response application/json

200 ok


```
{
  "dmz_enable": "Enable",
  "dmz_ipaddr": "192.168.1.0",
  "response_ping": "Disable",
  "disable_sip": "fw_sip_enab",
  "NatInboundFiltering": "Disable",
  "disable_igmp": "Disable",
  "wan_mtu": "1450"
}
```

POST **/SetWanAdv.cgi**

set WAN / firewall advance setting

Parameter

NatInboundFiltering	setting dmz enable /disable
disable_igmp	setting dmz enable /disable
disable_sip	setting dmz enable /disable
dmz_enable	setting dmz enable /disable
dmz_ipaddr	setting dmz enable /disable
response_ping	setting dmz enable /disable
wan_mtu	setting dmz enable /disable

Body application/json

```
{
  "NatInboundFiltering": "Disable",
  "disable_igmp": "Disable",
  "disable_sip": "fw_sip_enab",
  "dmz_enable": "Enable",
  "dmz_ipaddr": "192.168.1.0",
  "response_ping": "Disable",
  "wan_mtu": "1450"
}
```

Response application/json

200 ok

```
{
    "result:200
}
```

GET **/GetWifiManagement.cgi**

Get clould wifi management info

Parameter

enable_payment	host IP address or resolve name
mqtt_host	host IP address or resolve name
mqtt_password	host IP address or resolve name
mqtt_status	host IP address or resolve name
mqtt_status_msg	host IP address or resolve name
mqtt_username	host IP address or resolve name

Response application/json

200 ok


```
{
  "mqtt_host": "12.34.56.78",
  "mqtt_username": "mqttHost",
  "mqtt_password": "mqttPassword",
  "mqtt_status": 0,
  "mqtt_status_msg": "Disconnected",
  "enable_payment": ""
}
```

POST **/SetWifiManagement.cgi**

Set clould wifi management info

Parameter

mqtt_host	host IP address or resolve name
mqtt_password	host IP address or resolve name
mqtt_username	host IP address or resolve name

Body application/json

```
{
  "mqtt_host": "12.34.56.78",
  "mqtt_password": "mqttPassword",
  "mqtt_username": "mqttHost"
}
```

Response application/json

200 ok

```
{
    "result:200
}
```

GET **/GetWirelessAdv.cgi**

getting wireless advanced setting

Parameter

enable_atf	country area code
enable_beamforming	country area code
fronthaul_2g	country area code
fronthaul_5g	country area code
init_preamble_2g	country area code
init_preamble_5g	country area code
performance_boost	country area code
rts_2g	country area code
rts_5g	country area code
sku_name	country area code

Response application/json

200 ok


```
{
  "enable_atf": false,
  "enable_beamforming": true,
  "fronthaul_2g": true,
  "fronthaul_5g": true,
  "init_preamble_2g": "short",
  "init_preamble_5g": "long",
  "performance_boost": false,
  "rts_2g": 1234,
  "rts_5g": 567,
  "sku_name": ""
}
```

POST **/SetWirelessAdv.cgi**

setting wireless advanced setting

Parameter

enable_atf	country area code
enable_beamforming	country area code
fronthaul_2g	country area code
fronthaul_5g	country area code
init_preamble_2g	country area code
init_preamble_5g	country area code
performance_boost	country area code
rts_2g	country area code
rts_5g	country area code
sku_name	country area code

Body application/json

```
{
  "enable_atf": false,
  "enable_beamforming": true,
  "fronthaul_2g": true,
  "fronthaul_5g": true,
  "init_preamble_2g": "short",
  "init_preamble_5g": "long",
  "performance_boost": false,
  "rts_2g": 1234,
  "rts_5g": 567,
  "sku_name": "WW"
}
```

Response application/json

200 ok

```
{
    "result:200
}
```