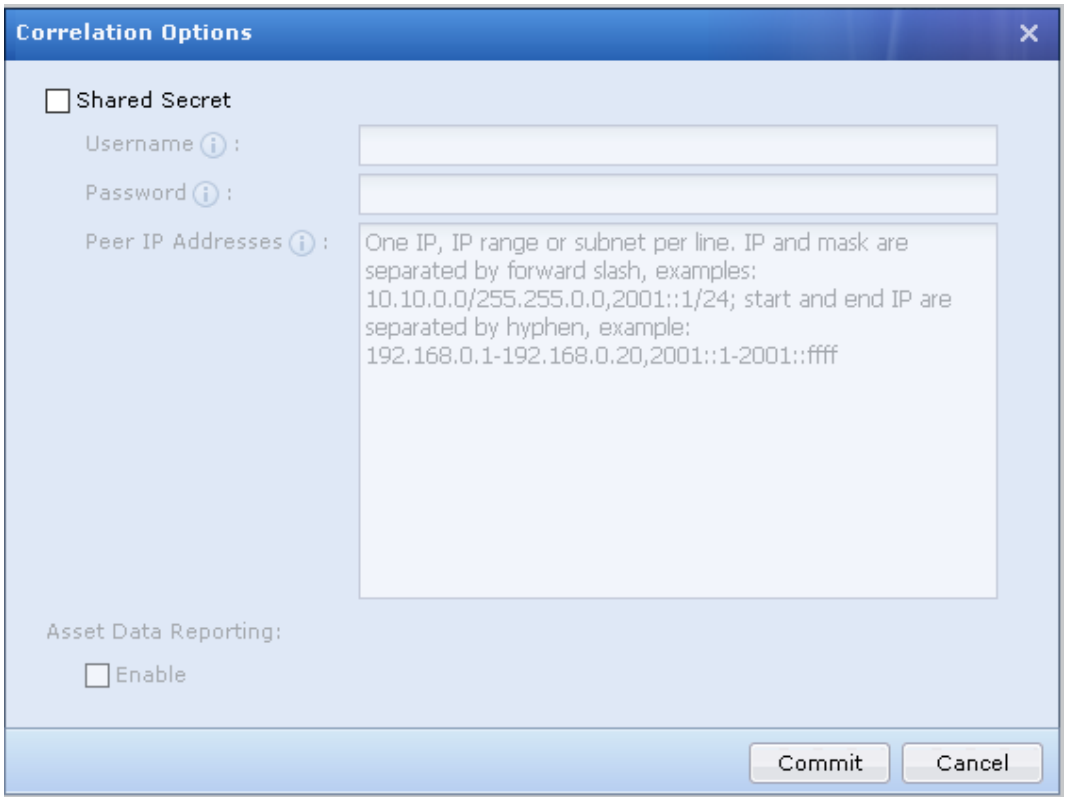
Table of Contents

1. Introduction

> This document describes how to use Restful APIs (Application Programing Interface) for external services and includes:
>
> - Currently supported APIs: status, user/group management, policy management, binding information and online user.
> - API input and output standard
>
> Public APIs are applicable to the following scenarios:
>
> - There is an unified management platform in your environment and administrators expect unified management and maintenance of Sangfor IAG, SG devices etc.
> - Systems supporting restful APIs can read or manage data from Sangfor IAG, SG etc through public APIs.

2. Use of RESTful APIs

> Suppose requesting parties are client (192.168.1.1) and IAG (192.168.1.2), of which client is a client and IAG is a device providing Restful service.
>
> - Enable Public API in Advanced Settings on the IAG device.
> - Configure shared key and add client's IP address (192.168.1.1) or IP segment to whitelist.
>
> 
>
>
> - random is a randomly generated string when the client requests. **and a random can be used only once within one hour. Otherwise, permission check will fail.**
> - md5 refers to md5 value calculated from concatenated string "shared key + random" . Please note that the string must be "shared key + random" and cannot be mixed in order.
>
> ```
> For example, the shared key is 1, and generated random is 2. The complete
> string is "12" and the md5 is c20ad4d76fe97759aa27a0c99bff6710.  Therefore,
> the requested parameter is:  random=2&md5=c20ad4d76fe97759aa27a0c99bff6710
> ```

- If request type of API is GET,random and md5 values will be appended to url, in format of http://acip:9999/interface name?random=xxxx&md5=xxxx,

For example, API parameters for online user: http://acip:9999/v1/status/online-user?random=xxxx&md5=xxxx

- If request type is POST, HTTP header must be specified Content-Type:application/json,random and md5 will be carried in JSON format when they are requested. For example, carried parameters for adding user API http://acip:9999/v1/user are:

```
{
    "random": "xxxx",
    "md5": "xxxx",
    "name": "hello"
}
```

- Notes:

1. **Global speed unit is bytes by default.**
2. The language of API information can be **Chinese and English**, and the default is English. The "Accept-Language" field in the request header is specified as "zh-CN" when being requested and language can be changed to Chinese.
3. User/group related APIs does not support **English/Chinese language change** . The language is the same as that in the IAG device providing Restful service.
4. Some APIs, such as, APIs for obtaining throughput, user traffic and application traffic rankings consume large amounts of resources. It is recommended not to use them too frequently.
5. To obtain global speed unit or radix, you can log in to the corresponding device: view "System - Advanced Settings- WEBUI" options or request for the following API:

http://acip:9999/v1/status/flux-units obtain (flow_unit:speed unit flux_prefix: data unit)

```
  ```json
  {
      "code": 0,
      "message": "Succeeded",
      "data": {
          "flow_unit": "B/s",
          "flux_prefix": 1024
      }
  }
  ```
```

```
client->AC:http://192.168.1.2:9999/interface name?random=xxx&md5=xxx (GET
example)
AC->>AC:   check permission and process requests.
AC->client:    If the check passes , it will return data, otherwise, return
specific error information.
```

3. API Description

1. Status API:

1.1 Obtain Version Information

- Request URL: http://acip:9999/v1/status/version
- Function: Obtain device version information.
- Method: `GET`
- Result parameters

Parameter	Required	Description
code	true	whether to obtain results successfully or not. 0 is yes and 1 is no.
message	true	When code is 0, message is always "successfully", and when code is 1, message is error.
data	false	When code is 0, it shows specific results, as the following JSON shows. When code is 1, no such field exists.

If the request succeeds, it will return:

```
{
  "code": 0,
  "message": "Succeeded",
  "data": "AC12.0.9.044  B Build20171216"
}
```

Data field description:

Field Name	Type	Description
data	string	version

If the request fails, it will return:

```
{
  "code": 1,
  "message": "Unknown error appeared. Failed to acquire data!"
}
```

1.2 Obtain the Number Of Online Users

- Request URL: http://acip:9999/v1/status/online-user
- Function: Obtain the number of online users on the current device.
- Method: `GET`
- Result parameters are same as above.

If the request succeeds, it will return:

```
{
    "code": 0,
    "message": "Succeeded",
    "data": 2
}
```

Data field description:

Field Name	Type	Description
data	number	number of online users

If the request fails, it will return:

```
{
  "code": 1,
  "message": "Unknown error appeared. Failed to acquire data!"
}
```

1.3 Obtain Number Of Sessions

- Request URL: http://acip:9999/v1/status/session-num
- Function: Obtain number of current device sessions.
- Method: `GET`
- Result parameters are same as above.

If the request succeeds, it will return:

```
{
    "code": 0,
    "message": "Succeeded",
    "data": 20
}
```

Data field description:

Field Name	Type	Description
data	number	number of sessions

If the request fails, it will return:

```
{
    "code": 1,
    "message": "Unknown error appeared. Failed to acquire data!"
}
```

1.4 Obtain Internal Database Information

- Request URL: http://acip:9999/v1/status/insidelib
- Function: Obtain information of internal database, including anti-virus database, URL database etc.
- Method: `GET`
- Result parameters are same as above.

If the request succeeds, it will return:

```
{
    "code": 0,
    "message": "Succeeded",
    "data": [
        {
            "current": "2017-12-15",
            "enable": true,
            "new": "2017-12-16",
            "name": "Anti-virus database",
            "expire": "2020-06-24",
            "type": "kav",
            "is_expired": 0
        },
        {
            "current": "2017-12-11  09:00:00",
            "enable": true,
            "new": "2017-12-11",
            "name": "URL database",
            "expire": "2020-06-24",
            "type": "url",
            "is_expired": 1
        }
    ]
}
```

Data field description:

Field Name	Type	Description
name	string	database name
current	string	current version
new	string	latest version
expire	string	expiration date of upgrade service license
enable	boolean	enable auto upgrade or not
type	string	Type of internal database (kav: anti-virus database url: URL database up: gateway patch database contchk: application signature database trace:audit rule database)
is_expired	number	Whether internal databases are expired or not? (0: valid 1: expired)

If the request fails, it will return:

```
{
    "code": 1,
    "message": "Unknown error appeared. Failed to acquire data!"
}
```

1.5 Obtain the Number of Activity Logs

- Request URL: http://acip:9999/v1/status/log
- Function: Obtain real-time number of activity logs, including rejected and logged.
- Method: `GET`
- Result parameters are same as above.

If the request succeeds, it will return:

```
{
    "code": 0,
    "message": "Succeeded",
    "data": {
        "block": 111,
        "record": 222
    }
}
```

Data field description:

Field Name	Type	Description
block	number	rejected
record	number	logged

If the request fails, it will return:

```
{
    "code": 1,
    "message": "Unknown error appeared. Failed to acquire data!"
}
```

1.6 Obtain CPU Usage

- Request URL: http://acip:9999/v1/status/cpu-usage
- Function: Obtain real-time CPU usage of device.
- Method: `GET`
- Result parameters are same as above.

If the request succeeds, it will return:

```
{
    "code": 0,
    "message": "Succeeded",
    "data": 4
}
```

Data field description:

Field Name	Type	Description
data	number	CPU usage (%)

If the request fails, it will return:

```
{
    "code": 1,
    "message": "Unknown error appeared. Failed to acquire data!"
}
```

1.7 Obtain Memory Usage

- Request URL: http://acip:9999/v1/status/mem-usage
- Function: Obtain real-time memory usage of devices.
- Method: `GET`
- Result parameters are same as above.

If the request succeeds, it will return:

```
{
    "code": 0,
    "message": "Succeeded",
    "data": 26
}
```

Data field description:

Field Name	Type	Description
data	number	memory usage (%)

If the request fails, it will return:

```
{
  "code": 1,
  "message": "Unknown error occurred. Failed to acquire data!"
}
```

1.8 Obtain Disk Usage

- Request URL: http://acip:9999/v1/status/disk-usage
- Function: Obtain disk usage of device.
- Method: `GET`
- Result parameters are same as above.

If the request succeeds, it will return:

```
{
    "code": 0,
    "message": "Succeeded",
    "data": 14
}
```

Data field description:

Field Name	Type	Description
data	number	disk usage (%)

If the request fails, it will return:

```
{
  "code": 1,
  "message": "Unknown error occurred. Failed to acquire data!"
}
```

1.9 Obtain System Time

- Request URL: http://acip:9999/v1/status/sys-time
- Function: Obtain current system time of device.
- Method: `GET`
- Result parameters are same as above.

If the request succeeds, it will return:

```
{
    "code": 0,
    "message": "Succeeded",
    "data": "2017-12-13 17:52:11"
}
```

Data field description:

Field Name	Type	Description
data	string	system time (Format: %Y-%m-%d %H:%M:%S)

If the request fails, it will return:

```
{
  "code": 1,
  "message": "Unknown error appeared. Failed to acquire data!"
}
```

1.10 Obtain Throughput

- Request URL: http://acip:9999/v1/status/throughput?_method=GET
- Function: Obtain outbound and inbound speed of devices.
- Method: `POST`
- Result parameters are same as above.
- Carrying filtering data is supported. For example:

```
{
 "filter": {
     "unit": "bytes",
     "interface": "eth1"
 }
}
```

Filtering field description:

Field Name	Type	Required	Description
unit	string	false	speed unit (bits or bytes), and bytes is default.
interface	string	false	network interface ("eth0,eth1" etc.), and any WAN interface is default.

If the request succeeds, it will return:

```
{
    "code": 0,
    "message": "Succeeded",
    "data": {
        "send": 1048350,
        "recv": 211295,
        "unit": "bytes"
    }
}
```

Data field description:

Field Name	Type	Description
send	number	total outbound speed
recv	number	total inbound speed
unit	string	speed unit (bits or bytes)

If the request fails, it will return:

```
{
    "code": 1,
    "message": "Unknown error appeared. Failed to acquire data!"
}
```

1.11 Obtain User Traffic Ranking

- Request URL: http://acip:9999/v1/status/user-rank?_method=GET
- Function: Obtain outbound, inbound and total traffic rankings by users
- Method: `POST`
- Carrying filtering data is supported. For example:

```
{
 "filter": {
     "top": 60,
     "line": "0",
     "groups": [
         "/group1",
         "/group2"
     ],
     "users": [
         "user1",
         "user2"
     ],
     "ips": [
         "ip1",
         "ip2"
     ]
 }
}
```

- Result parameters are same as above.

Filtering field description:

Field Name	Type	Description
top	number	Top N rankings
groups	string array	user groups to be filtered (It starts with "/")
users	string array	users to be filtered
ips	string array	IP to be filtered (Only single IP address is supported, and not IP group.)
line	string	Line No. (0: all lines 1-N: specific line)

> Notes: only one of the above mentioned filtering fields "groups", "users" and " ips" can be used for filtering. If you enter more than one items, only one filtering item will take effect and the priority is "groups > users > ips".

If the request succeeds, it will return:


```
{
    "code": 0,
    "message": "Succeeded",
    "data": [
        {
            "id": 0,
            "name": "11.16.1.102",
            "group": "/web",
            "ip": "11.16.1.102",
            "up": 807,
            "down": 42242,
            "total": 43049,
            "session": 45,
            "status": true,
            "detail": {
                "data": [
                    {
                        "id": 0,
                        "app": "others",
                        "line": 1,
                        "percent": 0,
                        "up": 92,
                        "down": 52,
                        "total": 144
                    },
                    {
                        "id": 1,
                        "app": "website browsing",
                        "line": 1,
                        "percent": 100,
                        "up": 715,
                        "down": 42190,
                        "total": 42905
                    }
                ]
            }
        }
    ]
}
```

Data field description:

Field Name	Type	Description
id	number	No.
name	string	username
group	string	user group
ip	string	IP address
up	number	outbound traffic (unit: bytes)
down	number	inbound traffic (unit: bytes)
total	number	total traffic (unit: bytes)
session	number	sessions
status	boolean	status (False means locked.)
detail	object	Details

If the request fails, it will return:

```
{
    "code": 1,
    "message": "Unknown error appeared. Failed to acquire data!"
}
```

1.12 Obtain Application Traffic Ranking

- Request URL: http://acip:9999/v1/status/app-rank?_method=GET
- Function: Obtain outbound, inbound and total traffic rankings by application.
- Method: `POST`
- Carrying filtering data is supported. For example:

```
{
 "filter": {
     "top": 60,
     "groups": [
         "/group1",
         "/group2"
     ],
     "line": "0"
 }
}
```

- Result parameters are same as above.

Filtering field description:

Field Name	Type	Description
top	number	Top N ranking
groups	string array	user group (It starts with "/".)
line	string	Line No. (0: all lines 1-N: specific line)

If the request succeeds, it will return:


```
{
    "code": 0,
    "message": "Succeeded",
    "data": [
        {
            "app": "Other website about securities trends [website browsing]",
            "line": 0,
            "line_name": "all lines",
            "up": 5308,
            "down": 148043,
            "total": 153351,
            "rate": 100,
            "session": 0,
            "user_data": {
                "data": [
                    {
                        "user": "11.16.1.101",
                        "grp": "/tue",
                        "ip": "11.16.1.101",
                        "up": 5308,
                        "down": 148043,
                        "total": 153351
                    }
                ],
                "count": 1
            }
        }
    ]
}
```

Data field description:

Field Name	Type	Description
app	string	App name
line	number	line
line_name	string	line name
up	number	outbound traffic (unit: bytes)
down	number	inbound traffic (unit: bytes)
total	number	total traffic (unit: bytes)
rate	number	traffic percent (%)
session	number	sessions
user_data	object	user details

If the request fails, it will return:

```
{
    "code": 1,
    "message": "Unknown error appeared. Failed to acquire data!"
}
```

1.13 Obtain Bandwidth Usage

- Request URL: http://acip:9999/v1/status/bandwidth-usage
- Function: Obtain bandwidth usage
- Method: `GET`
- Result parameters are same as above.

If the request succeeds, it will return:

```
{
    "code": 0,
    "message": "Successfully",
    "data": 30
}
```

Data field description:

Field Name	Type	Description
data	number	Bandwidth usage (%)

If the request fails, it will return:

```
{
    "code": 1,
    "message": "Unknow error, fail to acquire data!"
}
```

2. User/Group Related APIs

2.1 Add User

- Request URL: http://acip:9999/v1/user
- Function: Add new user and configure multiple user attributes
- Method: `POST`

Carried data in request:

```
{
    "name": "hello",
    "father_path": "/",
    "desc": "",
    "show_name": "",
    "expire_time": "2018-01-10 00:00:00"
}
```

Operation field description:

Field Name	Type	Required	Description
name	string	true	username
father_path	string	false	Parent group, to which users are added. (The group starts with"/", and adding users to domain user group is not supported.
desc	string	false	user description
show_name	string	false	user display name
expire_time	string	false	user account expiration date, and the format is "YY-MM-dd hh:mm:ss". If this field is empty or does not exist, it means the account will not expire.

If the request succeeds, it will return:

```
{
    "code": 0,
    "message": "Succeeded",
    "data": "Add user successfully!"
}
```

Extended user attributes

```
{
    "enable": true,
    "self_pass": {
        "enable": true,
        "pwdword": "xxx",
        "modify_once": false
    },
    "bind_cfg": [{
        "ip": "192.168.1.2",
        "mac": "ac-ed-ee-ee-ee-ee",
        "out_time": "2019-10-31",
        "bindgoal": "noauth_and_loginlimit",
        "desc":"example"
    }],
    "custom_cfg": {
        "attr1": "value1",
        "attr2": "value2"
    },
    "logout": true,
    "common_user": {
        "allow_change": true,
        "enable": true
    },
    "limit_ipmac": [
        "192.168.1.1-192.168.1.3",
        "192.168.1.2"
    ]
}
```

> Notes: Fields of basic attributes and extended attributes are relatively independent. Those fields are in the same level.

Extended fields description:

Field Name	Type	Required	Description
enable	boolean	false	Whether this user is enabled or not (true means enabled.)
self_pass	object	false	local password (When self_pass field is required, it means local password needs to be configured for the account and password field is required. password: local password modify_once: whether to change password upon first login.
bind_cfg	string array	false	Users can bind more than one. IP, MAC, out_time are supported: binding validity period (Remove this if you don't need it), bindgoal: binding method (noauth: authentication-free, loginlimit: restricted login, noauth_and_loginlimit: authentication-free & restricted login), desc: binding description.
custom_cfg	object	false	attribute name-attribute value of customized attributes
logout	boolean	false	whether to show logout page upon successful login.
common_user	object	false	enable: whether to allow concurrent login on multiple terminals. Allow_change: whether to allow modify local password.
limit_ipmac	object	false	blocked login address (IP address or MAC address and single IP address or IP segment are supported. The format of IP address: (192.168.1.1-192.168.1.2), and the format of MAC address: ee-ee-ee-ee-ee-ee)

2.2 Delete User Account

- Request URL: http://acip:9999/v1/user?_method=DELETE
- Function: Delete existing user account.
- Method: `POST`

Carried data in request:

```
{
    "name": "hello"
}
```

Operation field description:

Field Name	Type	Required	Description
name	string	true	Username

If the request succeeds, it will return:

```
{
    "code": 0,
    "message": "Succeeded",
    "data": "Delete user successfully!"
}
```

2.3 Search for User

- Request URL: http://acip:9999/v1/user?_method=GET
- Function: Search for user and return user account details.
- Method: `POST`

Carried data in request:

```
{
    "search_type": "user",
    "search_value": "Zhang San",
    "extend": {
        "custom_cfg": {
            "attr1": "value1"
        },
        "expire": {
            "start": "2018-05-01",
            "end": "2018-05-16"
        },
        "user_status": "all",
        "public": true,
        "father_path": "/"
    }
}
```

Operation field description:

Field Name	Type	Required	Description
search_type	string	true	search type (Username, IP and MAC are supported.)
search_value	string or object	true	search value (Search type and search value are listed in the table below.)
extend	object	false	search by extended attributes of users, optional

Search type and its corresponding search value:

search_type	search_value type	example	description
user	string	Zhang San	username (Fuzzy match is supported.)
ip	object	{"start":"1.1.1.1", "end": "2.2.2.2"}	IP segment bound to user (start: start IP address end: end IP address)
mac	string	ee-ee-ee-ee-ee-ee	mac address bound to user

If the request succeeds, it will return:

```
{
    "code": 0,
    "message": "Succeeded",
    "data": "details of more than one user accounts"
}
```

> Notes: The format of returned results is consistent with that of single user account. Please view user account details.

Search by extended user attributes

```
{
    "custom_cfg": {
      "attr1": "value1"
    },
    "expire": {
      "start": "2018-05-01",
      "end": "2018-05-16"
    },
    "user_status": "all",
    "public": true,
    "father_path":"/"
}
```

Extended fields description:

Field Name	Type	Required	Description
custom_cfg	object	false	attribute name-attribute value of customized attributes (Searching by more than one customized attributes is not supported.)
expire	object	false	Expiration date of account(start: start time end: end time; Start time and end time appear in pairs to form a time interval.
user_status	string	false	account status (There are 3 status values. all, enabled and disabled. The default is "all"(both enabled and disabled.)
public	boolean	false	true: Search for and filter out accounts that allow concurrent login on multiple terminals. Default is false.
father_path	boolean	false	Search for user accounts in the specified father_path group. Default is "/".

2.4 Edit User Profile

- Request URL: http://acip:9999/v1/user?_method=PUT
- Function: Edit existing user profiles and modify attributes of multiple users at the same time.
- Method: `POST`

Carried data in request:

```
{
    "name": "hello",
    "data": {
        "desc": "",
        "expire_time": "2018-01-10 00:00:00",
        "extend": {
            "father_path": "/hello/world",
            ...
        }
    }
}
```

Operation field description:

Field Name	Type	Required	Description
name	string	true	username to be modified
data	object	false	user profile needs to be modified

User attributes supported for modification

> Notes: This API also supports 2.1 Extended User Attributes

Field Name	Type	Required	Description
desc	string	false	user description
expire_time	string	false	expiration date of account (Never expire: Do not define expire_time field or expire_time field is null character string. Expiration date:expire_time field is a specific time. ("YY-MM-DD hh:mm:ss")

If the request succeeds, it will return:

```
{
    "code": 0,
    "message": "Succeeded",
    "data": "Modify user successfully!"
}
```

Extended user attributes

```
{
    "father_path": "/hello/world",
    "show_name": "show_name",
    "bind_cfg": [{
        "orig_mac": "ac-ed-ee-ee-ee-ee",
        "ip": "201.168.1.12",
        "mac": "ac-ed-ee-ee-ee-ee",
        "out_time": "2019-10-31",
        "bindgoal": "noauth_and_loginlimit",
        "desc":"example"
    }]
}
```

Extended fields description:

Field Name	Type	Mandatory	Description
father_path	string	false	parent group
show_name	string	false	display name
bind_cfg	object	false	The orig_ip or orig_mac field needs to be added to indicate the user binding relationship to be modified. (Which one is used to associate with the binding object used in the previous user binding relationship depends on ip). The remaining fields are the same as the corresponding extended attributes when adding users in 2.1.

2.5 Obtain User Details

- Request URL: http://acip:9999/v1/user?name=NAME
- Function: Obtain details of a certain user.
- Method: `GET`

Operation field description:

Field Name	Type	Required	Description
name	string	true	Username

If the request succeeds, it will return:


```
{
    "code": 0,
    "message": "Succeeded",
    "data": [
        {
            "enable": true,
            "bind_cfg": [
                "8.8.8.8",
                "200.200.0.1"
            ],
            "policy": [
                {
                    "founder": "admin",
                    "type": "quota control",
                    "status": true,
                    "name": "policy1",
                    "expire": "never expires"
                }
            ],
            "self_pass": {
                "enable": true,
                "modify_once": false
            },
```

```
            "custom_cfg": [
                {
                    "name": "attr1",
                    "value": "value1"
                }
            ],
            "create": "account is created by admin",
            "create_flag": false,
            "desc": "",
            "logout": false,
            "limit_ipmac": {
                "enable": true,
                "ipmac": [
                    "3.3.3.3",
                    "9.9.9.9"
                ]
            },
            "common_user": {
                "enable": true,
                "allow_change": true
            },
            "father_path": "/",
            "expire_time": {
                "enable": false
            },
            "name": "22",
            "show_name": ""
        }
    ]
}
```

Return field description:

Field Name	Type	Required	Description
enable	boolean	true	Whether the account is enabled or not.
bind_cfg	string array	true	IP and MAC address binding information of user
policy	object array	true	user-associated policy (to be specific,name:policy name status:whether the policy is enabled type: policy type founder:policy creator expire:expiration date of policy)
self_pass	object	true	enable: whether password is enabled or not. modify_once: whether password must be changed upon first login
custom_cfg	object array	true	customized attributes (to be specific,name:attribute name value: attribute value)
create	string	true	account creator
create_flag	boolean	true	whether user is added after successful authentication or by automatically synchronized.
desc	string	true	user description
name	string	true	username
show_name	string	true	user display name
logout	boolean	true	whether to show logout page upon login success or not.
limit_ipmac	object	true	blocked login information (enable: whether login block is enabled or not. ipmac: specific blocked IP and MAC addresses)
common_user	object	true	enable: whether to allow concurrent login on multiple terminals or not. allow_change:whether to allow change local password or not.
father_path	string	true	group where users in
expire_time	object	true	enable: whether to configure expiration time of user account or not. date: expiration date of user account. Only if enable is true, it has value.

2.6 Associate Policy to User

- Request URL: http://acip:9999/v1/user/netpolicy
- Function: Modify, add or delete policy information based on specified users. (If there is no DN field, search local users first and then domain users.)
- Method: `POST`

Carried data in request:

```
{
    "opr": "add",
    "user": "hello",
    "policy": [
        "policy1",
        "policy2"
    ]
}
```

Operation field description:

Field Name	Type	Required	Description
opr	string	true	Operation field (add: Add new policy. del: Delete policy. modify: Change the current policy to that one specified in policy field and the original one will be cleared.)
user	string	true	user who needs policy to be modified
dn	string	false	domain name
policy	string array	true	policy name (obtained via interfaces described in section 3.1)

If the request succeeds, it will return:

```
{
    "code": 0,
    "message": "Succeeded",
    "data": "Modify user policy successfully!"
}
```

2.7 Obtain User's Policy

- Request URL: http://acip:9999/v1/user/netpolicy?user=NAME

Field Name	Type	Required	Description
user	string	true	username
dn	string	false	domain name

- Function: Obtain policy associated to a certain user.
- Method: `GET`

If the request succeeds, it will return:

```
{
    "code": 0,
    "message": "Succeeded",
    "data": [
        "policy1",
        "policy2"
    ]
}
```

Return field description:

Field Name	Type	Description
data	string array	policy name array

2.8 Flow Control Policy Associated with User

- Request URL: http://acip:9999/v1/user/fluxpolicy
- Function: Modify, add or delete policy information based on specified users. (If there is no DN field, search local users first and then domain users.)
- Method: `POST`

Carried data in request:

```
{
    "opr": "add",
    "user": "hello",
    "dn": "xxx.08r2.com",
    "policy": [
        "policy1",
        "policy2"
    ]
}
```

Operation field description:

Field Name	Type	Required	Description
opr	string	true	Operation field (add: Add new policy. del: Delete policy. modify: Change the current policy to that one specified in policy field and the original one will be cleared.)
user	string	true	user who needs policy to be modified
dn	string	false	domain name
policy	string array	true	policy name (obtained via interfaces described in section 5.1)

If the request succeeds, it will return:

```
{
    "code": 0,
    "message": "Successfully",
    "data": "Modify user fluxpolicy success!"
}
```

2.9 Obtain User's Flow Control Policy

- Request URL: http://acip:9999/v1/user/fluxpolicy?user=NAME

Field Name	Type	Required	Description
user	string	true	username
dn	string	false	domain name

- Function: Obtain policy associated with a certain user.
- Method: `GET`

If the request succeeds, it will return:

```
{
    "code": 0,
    "message": "Successfully",
    "data": [
        "policy1",
        "policy2"
    ]
}
```

Return field description:

Field Name	Type	Description
data	string array	policy name array

2.10 Verify Username and Password

- Request URL: http://acip:9999/v1/user?_method=verify&name=NAME&password=PASSWORD

Parameter	Required	Description
_method	true	Fixed value: verify
name	true	username
password	true	password

- Function: Verify a local username and password.
- Method: `GET`

If the request succeeds, it will return:

```
{
  "code": 0,
  "message": "Success",
  "data": []
}
```

2.11 Add Group

- Request URL: http://acip:9999/v1/group
- Function: Add group and configure group description.
- Method: `POST`
  Carried data in request:

```
{
    "path": "/hello/world",
    "desc": "description"
}
```

Operation field description:

Field Name	Type	Required	Description
path	string	true	Group path to which group is added and it supports up to 15-level directory. (The path starts with "/ " and do not allow adding group to domain user group.)
desc	string	false	group description

If the request succeeds, it will return:

```
{
    "code": 0,
    "message": "Succeeded",
    "data": "Add group successfully!"
}
```

2.12 Delete Group

- Request URL: http://acip:9999/v1/group?_method=DELETE
- Function: Delete an existing group
- Method: `POST`

Carried data in request:

```
{
    "path": "/hello/world"
}
```

Operation field description:

Field Name	Type	Required	Description
path	string	true	group name to be deleted (The name starts with "/")

If the request succeeds, it will return:

```
{
    "code": 0,
    "message": "Succeeded",
    "data": "Delete group successfully!"
}
```

2.13 Modify Group Information

- Request URL: http://acip:9999/v1/group?_method=PUT
- Function: Modify information of a certain group
- Method: `POST`

Carried data in request:

```
{
    "path": "/hello",
    "desc": "description"
}
```

Operation field description:

Field Name	Type	Required	Description
path	string	true	group name
desc	string	true	group description

If the request succeeds, it will return:

```
{
    "code": 0,
    "data": "Modify group successfully!",
    "message": "Succeeded""
}
```

2.14 Policy Associated to Group

- Request URL: http://acip:9999/v1/group/netpolicy
- Function: Modify, add or delete policy information based on specified groups. (If there is no DN field, search local groups first and then domain groups.)
- Method: `POST`

Carried data in request:

```
{
    "opr": "add",
    "group": "/hello",
    "policy": [
        "policy1",
        "policy2"
    ]
}
```

Operation field description:

Field Name	Type	Required	Description
opr	string	true	Operation field (add: Add new policy. del: Delete policy. modify: Change the current policy to that one specified in policy field and the original one will be cleared.)
group	string	true	group that needs policy to be modified
dn	string	false	domain name
policy	string array	true	policy name (obtained via interfaces described in section 3.1)

If the request succeeds, it will return:

```
{
    "code": 0,
    "message": "Succeeded",
    "data": "Modify group policy successfully!"
}
```

2.15 Obtain Policy Associated to Group

- Request URL: http://acip:9999/v1/group/netpolicy?path=PATH

Field Name	Type	Required	Description
path	string	true	group name (It starts with "/")
dn	string	false	domain name

- Function: Obtain policy associated to a certain group
- Method: `GET`

If the request succeeds, it will return:


```
{
    "code": 0,
    "message": "Succeeded",
    "data": [
        "policy1",
        "policy2"
    ]
}
```


3. Policy Related APIs

3.1 Obtain all policy information

- Request URL: http://acip:9999/v1/policy/netpolicy
- Function: Obtain device's policy information
- Method: `GET`

If the request succeeds, it will return:


```
{
    "code": 0,
    "message": "Succeeded",
    "data": [
        {
            "policy_info": {
                "name": "hello",
                "type": "access control policy",
                "founder": "admin",
                "expire": "never expire",
                "status": true,
                "depict": ""
            },
            "user_info": {
                "ou": [],
                "aduser": [],
                "adgroup": [],
                "exc_aduser": [],
                "attribute": [],
                "user_attr_grp": [],
                "sourceip": [],
                "location": [
                    "All"
                ],
                "terminal": [
                    "All"
                ],
                "target_area": [
                    "All"
                ],
                "local": ""
            }
        }
    ]
}
```

Return field description:

Field Name	Type	Required	Description
name	string	true	policy name
type	string	true	policy type (access control, audit, ingress etc.)
founder	string	true	role who creates policy (admin etc.)
expire	string	true	expiration date
status	boolean	true	enabled or disabled
depict	string	true	policy description
ou	string array	true	online user information
aduser	string array	true	domain user information
adgroup	string array	true	domain security group information
exc_aduser	string array	true	information excluded domain user
attribute	string array	true	domain attribute information
user_attr_grp	string array	true	user/group attribute information
sourceip	string array	true	source IP
location	string array	true	location list
terminal	string array	true	terminal list
target_area	string array	true	target area
local	string	true	associated users (if applicable)

3.2 Obtain All Flow Control Information

- Request URL: http://acip:9999/v1/policy/fluxpolicy
- Function: Obtain device's existing flow control information
- Method: `GET`

If the request succeeds, it will return:


```
{
    "code": 0,
    "message": "",
    "data": [{
            "name": "Guarantee latency-sensitive applications",
            "target_applications": "DNS, game, financial market trends,
financial transactions, video conference, ICMP, remote access and online
banking",
            "target_users": "All users",
            "ip_group": "All",
            "active_time": "All day",
            "line": "Line 1",
            "assured_bandwidth": ["20000000", "20000000"],
            "max_bandwidth": ["100000000", "100000000"],
            "single_user_limit": ["-1", "-1"],
            "priority": "High",
            "policy_enable": true,
            "is_default_child": false,
            "is_low_speed": false,
            "children": [
                ...
            ]
        },
        ...
    ]
}
```

Data filed is array of channel and channel filed description:

Field Name	Type	Required	Description
name	string	true	channel name
target_applications	string	true	applicable applications
target_users	string	true	applicable user
targetip_group	string	true	destination IP group
active_time	boolean	true	schedule
assured_bandwidth	number array	true	assured bandwidth; array includes outbound and inbound bandwidth. -1 indicates no limitation.
max_bandwidth	number array	true	maximum bandwidth; array includes outbound and inbound bandwidth. -1 indicates no limitation.
single_user_limit	number array	true	maximum bandwidth for per user; array includes outbound and inbound bandwidth. -1 indicates no limitation.
policy_enable	boolean	true	Whether policy is enabled or not? true: enabled, false: disabled
is_default_child	boolean	true	Whether the channel is default? true: It is default, false: It is not default.
is_low_speed	string array	true	domain attribute information
childrens	object array	true	array of child channel objects, and the objects are same as channel objects

4. BindInfo API

4.1 Query User-IP/MAC Bindings

- Request URL: http://acip:9999/v1/bindinfo/user-bindinfo?search=VALUE

Parameter	Required	Description
search	true	Filtering by username, IP or MAC address is supported.

- Method: `GET`

If the request succeeds, it will return:

```
{
    "code": 0,
    "message": "Success",
    "data": [{
            "enable": true,
```

```
            "desc": "",
            "name": "a",
            "addr": "1.1.1.2",
            "addr_type": "ip",
            "time": "2018-05-14",
            "limitlogon": true,
            "noauth": {
                "enable": true,
                "indate": 0
            }
        }, {
            "enable": true,
            "desc": "",
            "name": "a",
            "addr": "11-11-11-11-11-11",
            "addr_type": "mac",
            "time": "2018-05-14",
            "limitlogon": true,
            "noauth": {
                "enable": false,
                "indate": 0
            }
        }, {
            "enable": true,
            "desc": "",
            "name": "a",
            "addr": "1.1.1.1+11-11-11-11-11-11",
            "addr_type": "ipmac",
            "time": "2018-05-14",
            "limitlogon": true,
            "noauth": {
                "enable": false,
                "indate": 0
            }
        }
    ]
}
```

Field description of data array objects

Field Name	Type	Description
enable	boolean	User and IP/MAC binding status
desc	string	Description
name	string	username
addr_type	string	Specifies the type of binding object, optionally with values of "ip", "mac", or "ipmac".
addr	string	Binding object, when of type "ipmac", in the format of "1.1.1.1+11-11-11-11"
time	string	Date is a string which indicates binding time.
limitlogon	boolean	false: limit login disabled, true: limit login enabled
noauth.enable	boolean	false: disable by open authentication true: enable by open authentication
noauth.indate	number	Unix timestamp. 0 indicates never expires and >0 indicates expiry timestamp.

4.2 Query IP&MAC Binding

- Request URL: http://acip:9999/v1/ipmac-bindinfo?search=VALUE

Parameter	Required	Description
search	true	Searching by IP or MAC address is supported.

- Method: `GET`

If the request succeeds, it will return:

```
{
    "code": 0,
    "message": "Success",
    "data": {
            "desc": "",
            "mac": "11-11-11-11-11-11",
            "ip": "1.1.1.2"}
}
```

Filed description of data array objects

Field Name	Type	Description
ip	string	IP address
mac	string	MAC address
desc	string	Description

4.3 Unbind User Account from IP or MAC address

- Request URL: http://acip:9999/v1/bindinfo/user-bindinfo?_method=DELETE
- Method: `POST`

Carried data in request:

```
{
  "addr": "1.1.1.1"
}
```

Operation field description:

Field Name	Type	Required	Description
addr	string	true	Object to bind, IP or MAC address.

If the request succeeds, it will return:

```
{
  "code": 0,
  "message": "Success"
}
```

4.4 Delete IP&MAC Binding

- Request URL: http://acip:9999/v1/bindinfo/ipmac-bindinfo?_method=DELETE

- Method: `POST`

Carried data in request:

```
{
  "ip": "1.1.1.1"
}
```

Operation field description:

Field Name	Type	Required	Description
ip	string	true	IP address

If the request succeeds, it will return:

```
{
  "code": 0,
  "message": "Success"
}
```

4.5 Bind User Account to IP or MAC address

- Request URL: http://acip:9999/v1/bindinfo/user-bindinfo
- Method: `POST`

Carried parameter in request:

```
{
  "enable": true,
  "name": "a",
  "desc": "",
  "addr_type": "ip",
  "addr": "1.1.1.1",
  "limitlogon": true,
  "noauth": {
    "enable": false,
    "expire_time": 0
  }
}
```

```
{
  "enable": true,
  "name": "a",
  "desc": "",
  "addr_type": "ipmac",
  "addr": "1.1.1.1+11-11-11-11-11-11",
  "limitlogon": true,
  "noauth": {
    "enable": false,
    "expire_time": 0
  }
}
```

Operation field description:

Field Name	Type	Required	Description
enable	boolean	true	User and IP/MAC binding status
desc	string	false	Description
name	string	true	Username
addr_type	string	true	Specifies the type of binding object, optionally with values of "ip", "mac", or "ipmac".
addr	string	true	Binding object, when of type "ipmac", in the format of "1.1.1.1+11-11-11-11"
limitlogon	boolean	true	false: limit login disabled, true: limit login enabled
noauth.enable	boolean	true	false: disable by open authentication, true: enable by open
noauth.expire_time	number	true	Unix timestamp. 0 indicates never expires and >0 indicates expiry timestamp.

If the request succeeds, it will return:

```
{
    "code": 0,
    "message": "Success"
}
```

4.6 Add IP&MAC Binding

- Request URL: http://acip:9999/v1/bindinfo/ipmac-bindinfo
- Method: `POST`

Carried parameter in request (Field description is same as 4.2 Query IP&MAC Binding)

```
{
  "desc": "",
  "mac": "11-11-11-11-11-11",
  "ip": "1.1.1.2"
}
```

Operation field description:

Field Name	Type	Required	Description
ip	string	true	IP address
mac	string	true	MAC address
desc	string	false	Description

If the request succeeds, it will return:

```
{
    "code": 0,
    "message": "Success"
}
```

5. OnlineUsers API

5.1 Obtain online information of a certain user

- Request URL: http://acip:9999/v1/online-users?_method=GET
- Function: Obtain the number of online users on the current device.
- Method: `POST`

Carried parameter in request:

```
{
  "status": "all",
  "terminal": "all",
  "filter": {
    "type": "user",
    "value": ["a"]
}
```

Operation field description:

Field Name	Type	Required	Description
status	string	false	user status, "all": all; "frozen": locked; "active": active
terminal	string	false	Endpoint type, "all": all; "pc": PC user; "mobile": mobile endpoint user; "multi": users from different types of endpoint devices; "iot": dumb endpoint user; "armarium": medical equipment user; "custom": custom device user
filter	object	false	search criteria, and if it is empty, it indicates search all.

Field Description of filter type object

Field Name	Type	Required	Description
type	string	true	Search type. "user": The value is username array. "ip": The value is IP address array. "mac": The value is MAC address array.
value	string array	true	Search object. When the type is "ip", the value is array object and the object is IP address or IP segment.

If the request succeeds, it will return:


```
{
    "code": 0,
    "message": "Success",
    "data": {
        "count": 1,
        "users": [{
                "name": "a",
                "show_name": "a",
                "father_path": "\/default",
                "ip": "1.1.1.1",
                "mac": "11-11-11-11-11-11",
                "terminal": 2,
                "authway": 1,
                "login_time": 1526352600,
                "online_time": 8
            }
        ]
    }
}
```


Field description of data object:

Field Name	Type	Description
count	number	online users that meet the criteria
users	object array	array of online users and a maximum of 100 users can be returned

Field description of users array

Field Name	Type	Description
name	string	username
show_name	string	display name
father_path	string	group path
ip	string	IP address
mac	string	MAC address
terminal	number	endpoint type, 0 indicates unrecognized type, 1 indicates mobile endpoint, 2 indicates PC, 3 indicates hybrid endpoint types, 4 indicates dumb endpoint, 5 indicates medical equipment, 7 indicates custom device
authway	number	authentication method, 0 indicates no authentication, 1 indicates password-based authentication, 2 indicates SMS authentication, 3 indicates Single Sign-on and 4 indicates open authentication.
login_time	number	Login timestamp (Unix timestamp)
online_time	number	online duration and the unit is seconds

5.2 Log out online users forceably

- Request URL: http://acip:9999/v1/online-users?_method=DELETE

Parameter	Required	Description
ip	true	User IP address

- Function: Log out online users forceably
- Method: `POST`

Carried data in request:

```
{
  "ip": "1.1.1.1"
}
```

Operation field description

Parameter	Type	Required	Description
ip	string	true	user IP address

If the request succeeds, it will return:

```
{
  "code": 0,
  "message": "Success"
}
```

5.3 Get Users Online via SSO

- Request URL: http://acip:9999/v1/online-users
- Function: Get users online via SSO
- Method: `POST`

Carried parameter in request:

```
{
  "ip": "1.1.1.1",
  "name": "a",
  "show_name": "a",
  "group": "\/default",
  "mac": "11-11-11-11-11-11",
}
```

Operation field description:

Field Name	Type	Required	Description
ip	string	true	user IP address
show_name	string	false	display name
group	string	false	group path
name	string	true	username
mac	string	false	MAC address

If the request succeeds, it will return:

```
{
    "code": 0,
    "message": "Success"
}
```

4. Error Details

Chinese	English
只支持本地请求!	Need request from local!
Restful配置更新成功!	Restful config updated!
Restful配置更新失败!	Fail to update restful config!
Restful服务未启用!	Restful service isn't enabled!
请求的IP不在白名单!	IP of request isn't in white list!
权限校验失败!	Permission check failed!
未知错误,获取数据失败!	Unknow error, fail to acquire data!

5. Notes

Only following APIs are supported by BBC

> - user/group related APIs
> - policy related APIs