

GestióIP IPAM

v3.5

IP address management software

API Guide

v25

www.gestioip.net

Table of Contents

1 Introduction.....	4
2 API usage.....	4
2.1 Accessing the API.....	4
2.1.1 Passing parameters to the API.....	4
2.2 Output format.....	4
2.3 Getting help.....	5
2.4 API version.....	6
2.5 Working with Hosts.....	7
2.5.1 Create hosts.....	7
2.5.2 Read host information.....	8
2.5.3 Update hosts.....	9
2.5.4 Delete hosts.....	10
2.5.5 List hosts.....	11
2.6 Working with Networks.....	13
2.6.1 Create networks.....	13
2.6.2 Read network information.....	14
2.6.3 Update networks.....	15
2.6.4 Delete networks.....	16
2.6.5 List Networks.....	17
2.6.6 Get the first free network within a root network.....	19
2.6.7 List all free IP addresses of a network.....	20
2.6.8 Show the first free address of a network.....	21
2.6.9 Reserve the first free address of a network.....	21
2.6.10 List all used IP addresses of a network.....	22
2.7 Working with VLANs.....	24
2.7.1 Create VLANs.....	24
2.7.2 Read VLAN information.....	25
2.7.3 Update VLANs.....	26
2.7.4 Delete VLANs.....	27
2.7.5 Get the first free VLAN number.....	27
2.7.6 List VLANs.....	28
2.8 Working with VLAN Providers.....	29
2.8.1 Create VLAN Provider.....	29
2.8.2 Read VLAN Provider information.....	30
2.8.3 Update VLAN Providers.....	30
2.8.4 Delete VLAN Providers.....	31
2.8.5 List VLAN Providers.....	32
2.9 Working with Clients.....	33
2.9.1 Create a new Client.....	33
2.9.2 Read Client.....	34
2.9.3 Update a Client.....	36
2.9.4 Delete a Client.....	37
2.9.5 List Clients.....	38
2.10 Discovery.....	40
2.10.1 Network discovery via SNMP.....	40

API Guide GestióIP IPAM v3.5

2.10.2 Host discovery via DNS.....	41
2.10.3 Host discovery via SNMP.....	42
3 Usage examples.....	44
3.1 Example Powershell.....	44
3.2 Example Ansible Playbook.....	46

1 Introduction

GestióIP's Application Programming Interface (API) defines functions to allow other application to interact with the GestióIP database.

The API supports the following functions:

Hosts:

create, read, update, delete, list hosts (with filter functions)

Networks:

create, read, update, delete, query first free network within a root network, list networks (with filter functions), show first free IP, reserve first free IP, list all free IP addresses, list all used addresses

VLANs:

create, read, update, delete, get first free VLAN number, list VLANs

VLAN Provider:

create, read, update, delete

2 API usage

2.1 Accessing the API

The API is accessible via the following URL:

<http://server/gestioip/api/api.cgi>

2.1.1 Passing parameters to the API

Parameters are passed in in "URL" format (attribute=value) to the API.

Example:

```
"request_type=updateHost&client_name=DEFAULT&ip=192.168.0.1&new_hostname=myhost&new_comment=my new comment&myCustomColumn1=my value"
```

2.2 Output format

The API supports the output formats XML or JSON. You can determine the output format by adding the attribute "output_type" to the request. Valid vales for "output_type" are "xml" or "json". Default output_format is XML.

Example: show information about network 192.168.22.0 in JSON format.

```
curl --url "http://localhost/gestioip/api/api.cgi" -u gipadmin:xxxxxx -d
"request_type=readNetwork&client_name=DEFAULT&ip=192.168.22.0&output_type=json"
{"readNetworkResult":{"error":"","Network":
{"cat":"prod","rootnet":"","descr":"Test
Network","ip_version":"v4","IP":"192.168.22.0","comment":"","BM":"25","site":"NY
","customColumns":{"type":"","region":"","customer-
id":"","customer":"","circuit-id":"","city":""},"sync":"n"}}}
```

2.3 Getting help

The API disposes about an incorporated help system.

Calling the API without parameter or with the parameter `request_type=help` returns a list of the supported attributes:

```
$ curl --url "http://localhost/gestioip/api/api.cgi" -u gipadmin:xxxxxx

<?xml version='1.0' encoding='UTF-8'?>
<helpResult>
  <supported_attributes>
    <client_name>CLIENT_NAME</client_name>
    <request_type>help|readHost|readHostHelp|updateHost|updateHostHelp|
createHost|createHostHelp|deleteHost|deleteHostHelp|listHosts|listHostsHelp|
readNetwork|readNetworkHelp|updateNetwork|updateNetworkHelp|createNetwork|
createNetworkHelp|deleteNetwork|deleteNetworkHelp|listNetworks|listNetworksHelp|
firstFreeNetwork|firstFreeNetworkHelp|freeNetworkAddresses|
freeNetworkAddressesHelp|firstFreeNetworkAddress|firstFreeNetworkAddressHelp|
reserveFirstFreeNetworkAddress|reserveFirstFreeNetworkAddressHelp|
usedNetworkAddresses|usedNetworkAddressesHelp|readVlan|readVlanHelp|updateVlan|
updateVlanHelp|createVlan|createVlanHelp|deleteVlan|deleteVlanHelp|
firstFreeVlan|firstFreeVlanHelp|listVlans|listVlansHelp|readVlanProvider|
readVlanProviderHelp|updateVlanProvider|updateVlanProviderHelp|
createVlanProvider|createVlanProviderHelp|deleteVlanProvider|
deleteVlanProviderHelp</request_type>
  </supported_attributes>
</helpResult>
```

Every function of the API disposes about a specific help attribute. One can call the help function by adding the keyword “Help” to the `request_type` parameter.

Example: getting help for the function “createVlan” by using the `request_type` “createVlanHelp”.

```
$ curl --url "http://localhost/gestioip/api/api.cgi" -u gipadmin:xxxxxx -d
"request_type=createVlanHelp"

<?xml version='1.0' encoding='UTF-8'?>
<createVlanHelpResult>
  <supported_attributes>
    <request_type>createVlan</request_type>
```

```
<client_name>CLIENT_NAME</client_name>
  <new_bg_color>LightCyan|LightBlue|dodgerblue|LimeGreen|SeaGreen|
pink|white|black|brown|red|DarkOrange</new_bg_color>
  <new_font_color>LightCyan|LightBlue|dodgerblue|LimeGreen|SeaGreen|
pink|white|black|brown|red|DarkOrange</new_font_color>
  <new_comment>NEW VLAN COMMENT</new_comment>
  <new_name>NEW VLAN NAME</new_name>
  <new_number>NEW_VLAN_NUMBER</new_number>
  <new_provider>NEW VLAN PROVIDER NAME</new_provider>
</supported_attributes>
</createVlanHelpResult>
```

2.4 API version

To consult the version of the API use request_type “version”.

Example:

```
$ curl --url "https://localhost/gestioip/api/api.cgi" -u gipadmin:XXXXXX
-d"request_type=version"
<?xml version='1.0' encoding='UTF-8'?>
<versionResult>
  <version>1.2.27</version>
</versionResult>
```

2.5 Working with Hosts

The API allows to create, read, update and delete hosts including the support for host's custom columns.

2.5.1 Create hosts

Use request_type “createHost” to create hosts.

Required attributes:

```
request_type=createHost
client_name=CLIENT_NAME
ip=IP_ADDRESS_HOST
new_hostname=HOSTNAME
```

Optional attributes:

```
new_descr=DESCR
new_site=SITE
new_cat=CATEGROY
new_int_admin=yn
new_comment=COMMENT
new_custom_column=VALUE
```

Example:

```
$ curl --url "http://localhost/gestioip/api/api.cgi" -u gipadmin:xxxxxx -d
"request_type=createHost&client_name=DEFAULT&ip=192.168.20.15&new_hostname=switc
hA77&new_comment=new switch&new_vendor=cisco&new_COL_A=new value"
```

```
<?xml version='1.0' encoding='UTF-8'?>
<insertHostResult>
  <error>
  </error>
  <Host>
    <new_id>1574</new_id>
    <IP>192.168.20.15</IP>
    <new_hostname>switchA77</new_hostname>
    <new_descr></new_descr>
    <new_site></new_site>
    <new_cat></new_cat>
    <new_int_admin>n</new_int_admin>
    <new_comment>new switch</new_comment>
    <new_update_type></new_update_type>
```

```

<alive>never checked</alive>
<last_response></last_response>
<ip_version>v4</ip_version>
<customColumns>
  <new_COL_A>new value</new_COL_A>
  <new_COL_B></new_COL_B>
  <new_OS></new_OS>
  <new_URL></new_URL>
  <new_linked_IP></new_linked_IP>
  <new_model></new_model>
  <new_vendor>cisco</new_vendor>
</customColumns>
</Host>
</insertHostResult>

```

2.5.2 Read host information

Use request_type “readHost” to obtain all host attributes. Use the IP address or the hostname as identifier. IP addresses are unique. Hostnames may not be unique. If there are more than one host with the same hostname in the database, **the first found host will be returned.**

Required attributes:

request_type=readHost

client_name=*CLIENT_NAME*

ip=*IP_ADDRESS_HOST* or

hostname=HOSTNAME

Example:

```
$ curl --url "http://localhost/gestioip/api/api.cgi" -u gipadmin:xxxxxx -d
"request_type=readHost&client_name=DEFAULT&ip=192.168.20.15"
```

```

<?xml version='1.0' encoding='UTF-8'?>
<readHostResult>
  <error>
  </error>
  <Host>
    <id>1574</id>
    <IP>192.168.20.15</IP>
    <hostname>switchA77</hostname>
    <descr></descr>
    <site></site>
    <cat></cat>
    <int_admin>n</int_admin>
    <comment>new switch</comment>
    <update_type></update_type>
    <alive>never checked</alive>
    <last_response></last_response>
    <ip_version>v4</ip_version>

```



```

    <customColumns>
      <COL_A>new value</COL_A>
      <COL_B></COL_B>
      <OS></OS>
      <URL></URL>
      <linked IP></linked IP>
      <vendor>cisco</vendor>
    </customColumns>
  </Host>
</readHostResult>

```

2.5.3 Update hosts

Use request_type “updateHost” to update host attributes.

Required attributes:

request_type=updateHost

client_name=*CLIENT_NAME*

ip=*IP_ADDRESS_HOST*

Optional attributes:

new_hostname=*HOSTNAME*

new_descr=*DESCR*

new_site=*SITE*

new_cat=*CATEGROY*

new_int_admin=y|n

new_comment=*COMMENT*

new_custom_column=*VALUE*

Example:

```

$ curl --url "http://localhost/gestioip/api/api.cgi" -u gipadmin:xxxxxx -d
"request_type=updateHost&client_name=DEFAULT&ip=192.168.20.15&new_hostname=switc
hA77_1&new_comment=NEW COMMENT&new_vendor=cisco&new_COL_A=changed value"

```

```

<?xml version='1.0' encoding='UTF-8'?>
<UpdateHostResult>
  <error>
  </error>
  <Host>
    <id>1574</id>
    <IP>192.168.20.15</IP>
    <new_hostname>switchA77_1</new_hostname>

```

```

<new_descr></new_descr>
<new_site></new_site>
<new_cat></new_cat>
<new_int_admin>n</new_int_admin>
<new_comment>NEW COMMENT</new_comment>
<new_update_type></new_update_type>
<alive>never checked</alive>
<last_response></last_response>
<ip_version>v4</ip_version>
<customColumns>
  <new_COL_A>changed value</new_COL_A>
  <new_COL_B></new_COL_B>
  <new_OS></new_OS>
  <new_URL></new_URL>
  <new_linked IP></new_linked IP>
  <new_Updateel></new_Updateel>
  <new_vendor>cisco</new_vendor>
</customColumns>
</Host>
</UpdateHostResult>

```

2.5.4 Delete hosts

Use request_type “deleteHost” to delete hosts. Use the IP address or the hostname as identifier. IP addresses are unique. Hostnames may not be unique. If there are more than one host with the same hostname found in the database, **the first found host will be deleted.**

Required attributes:

```

request_type=deleteHost
client_name=CLIENT_NAME
ip=IP_ADDRESS_HOST or
hostname=HOSTNAME

```

Example:

```

$ curl --url "http://localhost/gestioip/api/api.cgi" -u gipadmin:xxxxxx -d
"request_type=deleteHost&client_name=DEFAULT&ip=192.168.20.15"

```

```

<?xml version='1.0' encoding='UTF-8'?>
<deleteHostResult>
  <error>
</error>
  <Host>
</Host>
</deleteHostResult>

```

2.5.5 List hosts

Use request type “listHosts” to get a list of hosts. This option supports filters which allow to filter for different host attributes.

Required attributes:

request_type=listHosts

client_name=*CLIENT_NAME*

Optional attributes:

ip_version=v4|v6

filter=*attributeA::value1|value2,attributeB::value*

Supported filter attributes: hostname, site, category, comment, description, all custom columns names

Example: filter=site::*site_A|site_B*,category::*server*

In this example the filter matches for networks with (site=site_A OR site=site_B) AND category=server

[[MATCH_ALL]] – The [[MATCH_ALL]] option forces the filter to match also substrings.

Example:

filter=site::site_A would match exactly the site site_A (case sensitive)

filter=site::site_A[[MATCH_ALL]] would match for example the following sites:

site_A

site_A1

Site_a_b

123-site_A

include_id=yes|no

Include the unique object ID as first list entry within the result. Default: no

Example output: <Host>3467,10.0.3.4,test1,Lon1,server,,,</Host>

If the no_csv option is specified, the id is always included.

no_csv=yes|no

Default behavior is to display the host entries in form of a CSV list (see example 1). Setting no_csv to “yes” forces a full XML/JSON output. The individual host entries would be also

displayed in XML/JSON style (see example 2). Default: no

`limit=integer between 1 and 9999`

Limit the number of resulting networks

`page=integer between 0 and 9999`

Shows page *n* of a limited output (starting with 0). Only together with the “limit” option.

Example 1:

```
$ curl --url "http://localhost/gestioip/api/api.cgi" -u gipadmin:xxxxxx
-d"request_type=listHosts&client_name=DEFAULT&filter=hostname::test1"
```

```
<?xml version='1.0' encoding='UTF-8'?>
<listHostsResult>
  <error>
</error>
  <HostList>
    <Host>10.0.3.4, test1, Lon1, server, , , </Host>
    <Host>10.100.3.4, test1, Lon1, server, , </Host>
    <Host>192.168.7.93, test1, Lon2, workstation, , , </Host>
  </HostList>
</listHostsResult>
```

Example 2:

```
$ curl --url "http://localhost/gestioip/api/api.cgi" -u gipadmin:xxxxxx
-d"request_type=listHosts&client_name=DEFAULT&filter=hostname::test1&no_csv=yes"
```

```
<?xml version='1.0' encoding='UTF-8'?>
<listHostsResult>
  <error>
</error>
  <HostList>
    <Host>
      <id>699</id>
      <IP>10.0.3.4</IP>
      <hostname>test1</hostname>
      <descr></descr>
      <site>Lon1</site>
      <cat>server</cat>
      <int_admin>n</int_admin>
      <comment></comment>
      <update_type></update_type>
      <alive>up</alive>
      <last_response>01/06/2018 12:42:47</last_response>
      <ip_version>v4</ip_version>
      <customColumns>
        <CM></CM>
        <custom_test></custom_test>
        <model></model>
        <OS></OS>
        <URL></URL>
        <vendor></vendor>
      </customColumns>
```

```

    </Host>
    <Host>
      <id>699</id>
      <IP>10.0.3.4</IP>
      <hostname>test1</hostname>
      <descr></descr>
      <site>Lon1</site>
...
    </Host>
  </HostList>
</listHostsResult>

```

2.6 Working with Networks

The API allows to create, read, update and delete networks, query the first free network within a root network, list networks (with filter functions), show the first free IP address of a network, reserve the first free address of a network and list all free IP addresses of a network.

2.6.1 Create networks

Use request_type “createNetwork” to create new networks.

Required attributes:

```

request_type=createNetwork
client_name=CLIENT_NAME
ip=IP_ADDRESS_NETWORK
new_BM=BITMASK

```

Optional attributes:

```

network_type="root|non-root" (default: non-root)
new_descr=DESCR
new_site=SITE
new_cat=CATEGROY
new_comment=COMMENT
new_sync=y|n
new_custom_column=VALUE

```

Note: custom column vlan must be specified with the attribute “new_vlan_id”. The ID is the

internal ID of the VLAN. Get the internal ID of the VLAN by doing a request with request_type=readVlan“

Example:

```
$ curl --url "http://localhost/gestioip/api/api.cgi" -u gipadmin:xxxxxx -d
"request_type=createNetwork&client_name=DEFAULT&ip=192.168.22.0&new_descr=Test
Network&new_BM=25&new_site=Madr&new_cat=Prod&new_vlan_id=121"
```

```
<?xml version='1.0' encoding='UTF-8'?>
<createNetworkResult>
  <error>
  </error>
  <Network>
    <new_id>697</new_id>
    <new_IP>192.168.22.0</new_IP>
    <new_BM>25</new_BM>
    <new_descr>Test Network</new_descr>
    <new_site>Madr</new_site>
    <new_cat>Prod</new_cat>
    <new_comment></new_comment>
    <new_sync>n</new_sync>
    <customColumns>
      <new_vlan>151 - Vlan Test Network</new_vlan>
    </customColumns>
  </Network>
</createNetworkResult>
```

2.6.2 Read network information

Use request_type “readNetwork” to obtain all network attributes.

Required attributes:

request_type=readNetwork

client_name=*CLIENT_NAME*

ip=*IP_ADDRESS_NETWORK*

BM=*BM* (only required for root networks)

Optional attributes:

network_type="root|non-root" (default: non-root)

Example: Read information of a non-root network

```
curl --url "http://localhost/gestioip/api/api.cgi" -u gipadmin:xxxxxx -d
"request_type=readNetwork&client_name=DEFAULT&ip=192.168.22.0"
```

```
<?xml version='1.0' encoding='UTF-8'?>
<readNetworkResult>
  <error>
  </error>
  <Network>
    <id>697</id>
    <IP>192.168.22.0</IP>
    <BM>25</BM>
    <descr>Test Network</descr>
    <site>Madr</site>
    <cat>prod</cat>
    <comment></comment>
    <sync>n</sync>
    <ip_version>v4</ip_version>
    <customColumns>
      <vlan>151 - Vlan Test Network</vlan>
    </customColumns>
  </Network>
</readNetworkResult>
```

2.6.3 Update networks

Use request_type “updateNetwork” to update network attributes.

Required attributes:

request_type=updateNetwork

client_name=*CLIENT_NAME*

ip=*IP_ADDRESS_NETWORK*

BM=*BM* (only required for root networks)

Optional attributes:

network_type=”root|non-root” (default: non-root)

new_descr=*DESCR*

new_site=*SITE*

new_cat=*CATEGROY*

new_comment=*COMMENT*

new_sync=y|n

new_custom_column=*VALUE*

Example:

```
$ curl --url "http://localhost/gestioip/api/api.cgi" -u gipadmin:xxxxxx -d
"request_type=updateNetwork&client_name=DEFAULT&ip=192.168.22.0&new_descr=Change
```

```
d Network comment&new_site=&new_cat=test&new_vlan_id=122"
```

```
<?xml version='1.0' encoding='UTF-8'?>
<updateNetworkResult>
  <Network>
    <id>697</id>
    <IP>192.168.22.0</IP>
    <BM>25</BM>
    <new_descr>Changed Network comment</new_descr>
    <new_site></new_site>
    <new_cat>test</new_cat>
    <new_comment></new_comment>
    <new_sync>n</new_sync>
    <ip_version>v4</ip_version>
    <customColumns>
      <new_vlan>202 - VLAN TEST</new_vlan>
    </customColumns>
  </Network>
</updateNetworkResult>
```

2.6.4 Delete networks

Use request_type “deleteNetwork” to delete a network. Deleting root networks will not delete networks which falls in it’s range. Deleting non-root networks will delete all host entries within this network.

Required attributes:

```
request_type=deleteNetwork
client_name=CLIENT_NAME
ip=IP_ADDRESS_NETWORK
```

Optional attributes:

```
BM=BITMASK OF THE NETWORK TO DELETE
network_type=root
```

If the network which should be deleted is a root network, set “network_type” to “root”.

Example: delete a non-root network:

```
$ curl --url "http://localhost/gestioip/api/api.cgi" -u gipadmin:xxxxxx -d
"request_type=deleteNetwork&client_name=DEFAULT&ip=192.168.22.0"
<?xml version='1.0' encoding='UTF-8'?>
<deleteNetworkResult>
  <error>
</error>
```



```

    <Network>
  </Network>
</deleteNetworkResult>

```

Example: delete a root network.

```

$ curl --url "http://localhost/gestioip/api/api.cgi" -u gipadmin:xxxxxx -d
"request_type=deleteNetwork&client_name=DEFAULT&ip=172.16.4.0&BM=23&network_type
=root"
<?xml version='1.0' encoding='UTF-8'?>
<deleteNetworkResult>
  <error>
</error>
  <Network>
</Network>
</deleteNetworkResult>

```

2.6.5 List Networks

Use request type “listNetworks” to get a list of networks. This option supports filters which e.g. allow to list only networks from a specific client/site and/or category.

Required attributes:

request_type=listNetworks

client_name=*CLIENT_NAME*

Optional attributes:

ip_version=v4|v6

filter=*attributeA::value1|value2,attributeB::value*

Supported filter attributes: site, category, comment, description, all custom columns names

Example: filter=site::*site_A|site_B*,category::*prod*

In this example the filter matches for networks with (site=site_A OR site=site_B) AND category=prod

[[MATCH_ALL]] – The [[MATCH_ALL]] option forces the filter to match substrings.

Example:

filter=site::site_A would match exactly the site “site_A” (case sensitive)

filter=site::site_A[[MATCH_ALL]] would match for example the following sites:

site_A

site_A1

Site_a_b

123-site_A

network_type=root

Default behavior is to return a list of non-root networks. Use the attribute “network_type=root “ to receive a list of the root networks of this client.

include_id=yes|no

Include the unique object ID as first list entry within the result. Default: no

Example output: <network>14567,192.168.0.0/24,site,category,...</network>

no_csv=yes|no

Default behavior is to display the networks entries in form of a CSV list (see example 1). Setting no_csv to “yes” forces a full XML/JSON output. The individual networks entries would be also displayed in XML/JSON style (see example 2). Default: no

limit=*integer between 1 and 9999*

Limit the number of resulting networks

page=*integer between 0 and 9999*

Shows page *n* of a limited output (starting with 0). Only together with the “limit” option.

A query with no filter option will list all networks of the specified client.

Example 1:

```
$ curl --url "http://localhost/gestioip/api/api.cgi" -u gipadmin:xxxxxxx -d
"request_type=listNetworks&client_name=DEFAULT"
```

```
<?xml version='1.0' encoding='UTF-8'?>
<listNetworksResult>
  <error>
  </error>
  <NetworkList>
    <network>192.168.0.0/24,site,category,...</network>
    <network>192.168.1.0/24</network>
    <network>192.168.2.0/24</network>
    <network>192.168.7.0/24</network>
    <network>172.16.30.0/24</network>
    <network>172.16.31.0/24</network>
    <network>172.16.32.0/24</network>
    ...
  </NetworkList>
</listNetworkResult>
```

Example 2:

```
$ curl --url "http://localhost/gestioip/api/api.cgi" -u gipadmin:xxxxxx -d
"request_type=listNetworks&client_name=DEFAULT&no_csv=yes"
```

```
<?xml version='1.0' encoding='UTF-8'?>
<listNetworksResult>
  <error>
  </error>
  <NetworkList>
    <Network>
      <id>1116</id>
      <IP>192.168.0.0</IP>
      <BM>24</BM>
      <descr></descr>
      <site>Lon1</site>
      <cat>corp</cat>
      <comment></comment>
      <sync>n</sync>
      <ip_version>v4</ip_version>
      <rootnet></rootnet>
      <customColumns>
        <usage>256,156,100</usage>
        <vlan></vlan>
      </customColumns>
    </Network>
    <Network>
      <id>1119</id>
      <IP>192.168.1.0</IP>
      <BM>24</BM>
      <descr></descr>
    ...
  </Network>
</NetworkList>
</listNetworkResult>
```

2.6.6 Get the first free network within a root network

Use request_type “firstFreeNetwork” to obtain the first free end-network address within a root network in function of the specified bitmask.

An exception are IPv6 networks with a BM < 64. As in IPv6 no exist end-networks with a BM < 64, in the case that you choose a BM < 64, the resulting networks are root-networks.

Required attributes:

request_type=firstFreeNetwork

client_name=*CLIENT_NAME*

rootnet_ip=*IP_ADDRESS_ROOT_NETWORK*

rootnet_BM=*BITMASK_ROOT_NETWORK*

BM=*BITMASK_OF_THE_DESIRED_FREE_NETWORK*

Example:

```
$ curl --url "http://localhost/gestioip/api/api.cgi" -u gipadmin:xxxxxx -d
"request_type=firstFreeNetwork&client_name=DEFAULT&rootnet_ip=192.168.0.0&rootne
t_BM=16&BM=24"
```

```
<?xml version='1.0' encoding='UTF-8'?>
<firstFreeNetworkResult>
  <error>
  </error>
  <Network>
    <IP>192.168.3.0</IP>
    <BM>24</BM>
  </Network>
</firstFreeNetworkResult>
```

Instead of the attributes `rootnet_IP` and `rootnet_BM` the API disposes about a “special” feature which allows to filter for the attributes “region”, “city” and “site”. To use this feature you need to create the custom network columns “region” and “city” first.

2.6.7 List all free IP addresses of a network

Use `request_type` “freeNetworkAddresses” to list all free IP addresses of a network.

Required attributes:

`request_type=freeNetworkAddresses`

`client_name=CLIENT_NAME`

`ip=IP_ADDRESS_NETWORK`

Example:

```
$ curl --url "http://localhost/gestioip/api/api.cgi" -u gipadmin:xxxxxx -d
"request_type=freeNetworkAddresses&client_name=DEFAULT&ip=192.168.22.0"
```

```
<?xml version='1.0' encoding='UTF-8'?>
<freeNetworkAddressesResult>
  <error>
  </error>
  <Network>
    <IP>192.168.22.0</IP>
    <freeAddress>192.168.22.5</freeAddress>
    <freeAddress>192.168.22.10</freeAddress>
    <freeAddress>192.168.22.11</freeAddress>
    <freeAddress>192.168.22.22</freeAddress>
    <freeAddress>192.168.22.23</freeAddress>
    <freeAddress>192.168.22.24</freeAddress>
    <freeAddress>192.168.22.25</freeAddress>
    <freeAddress>192.168.22.26</freeAddress>
```

```

    <freeAddress>192.168.22.27</freeAddress>
    <freeAddress>192.168.22.28</freeAddress>
    <freeAddress>192.168.22.29</freeAddress>
    <freeAddress>192.168.22.30</freeAddress>
    ...
  </Network>
</freeNetworkAddressesResult>

```

2.6.8 Show the first free address of a network

Use request_type “firstFreeNetworkAddress” to show the first free IP addresses of a network.

Required attributes:

request_type= firstFreeNetworkAddress

client_name=*CLIENT_NAME*

ip=*IP_ADDRESS_NETWORK*

Example:

```
$ curl --url "http://localhost/gestioip/api/api.cgi" -u gipadmin:xxxxxx -d
"request_type=firstFreeNetworkAddress&client_name=DEFAULT&ip=192.168.22.0"
```

```

<?xml version='1.0' encoding='UTF-8'?>
<firstFreeNetworkAddressResult>
  <error>
  </error>
  <Network>
    <IP>192.168.22.0</IP>
    <freeAddress>192.168.22.5</freeAddress>
  </Network>
</firstFreeNetworkAddressResult>

```

2.6.9 Reserve the first free address of a network

Use request_type “reserveFirstFreeNetworkAddress” to reserve the first free IP addresses of a network.

Required attributes:

request_type= reserveFirstFreeNetworkAddress

client_name=*CLIENT_NAME*

ip=*IP_ADDRESS_NETWORK*

Optional attributes:

new_hostname=HOSTNAME (if not specified, the hostname will automatically be set to “reserved”)

new_descr=*DESCR*

`new_site=SITE`

`new_cat=CATEGROY`

`new_int_admin=yln`

`new_comment=COMMENT`

`new_custom_column=VALUE`

Example:

```
$ curl --url "https://localhost/gestioip/api/api.cgi" -u gipadmin:XXXXXX -d
"request_type=reserveFirstFreeNetworkAddress&client_name=DEFAULT&ip=192.168.29.0
&new_comment=New comment&new_hostname=new_host"
```

```
<?xml version='1.0' encoding='UTF-8'?>
<reserveFirstFreeNetworkAddressResult>
  <error>
  </error>
  <Host>
    <id>800</id>
    <IP>192.168.29.10</IP>
    <new_hostname>new_host</new_hostname>
    <new_descr></new_descr>
    <new_site></new_site>
    <new_cat></new_cat>
    <new_int_admin>n</new_int_admin>
    <new_comment>New comment</new_comment>
    <new_update_type></new_update_type>
    <alive>never checked</alive>
    <last_response></last_response>
    <ip_version>v4</ip_version>
    <customColumns>
      <new_contact></new_contact>
      <new_MAC></new_MAC>
      <new_model></new_model>
      <new_OS></new_OS>
      <new_serial></new_serial>
      <new_vendor></new_vendor>
    </customColumns>
  </Host>
</reserveFirstFreeNetworkAddressResult>
```

2.6.10 List all used IP addresses of a network

Use `request_type` “usedNetworkAddresses” to list all used IP addresses of a network.

Required attributes:

`request_type=freeNetworkAddresses`

`client_name=CLIENT_NAME`

`ip=IP_ADDRESS_NETWORK`

Optional attributes:

include_id=yes|no

no_csv=yes|no

Default behavior is to display the host entries in form of a CSV list (see example 1). Setting no_csv to “yes” forces a full XML/JSON output. The individual host entries would be also displayed in XML/JSON style (see example 2). Default: no

Include the unique object ID as first list entry within the result. Default: no

Example output:

Example 1:

```
$ curl --url "https://localhost/gestioip/api/api.cgi" -u gipadmin:XXXXXX -d
"request_type=usedNetworkAddresses&client_name=DEFAULT&ip=10.20.10.0"
<?xml version='1.0' encoding='UTF-8'?>
<usedNetworkAddressesResult>
  <error>
</error>
  <Network>
    <IP>10.20.10.0</IP>
    <usedAddress>10.20.10.1, host1, Lon1, FW,,,,, juniper,, </usedAddress>
    <usedAddress>10.20.10.4, host_4, Lon1,,,,,,, </usedAddress>
    <usedAddress>10.20.10.5, host_5, Lon1,,, test,,,,, </usedAddress>
    <usedAddress>10.20.10.9, host_9, Lon1,,,,,,, </usedAddress>
    <usedAddress>10.20.10.76, test, Lon1,,,,,,, </usedAddress>
  </Network>
</usedNetworkAddressesResult>
```

Example 2:

```
$ curl --url "https://localhost/gestioip/api/api.cgi" -u gipadmin:XXXXXX -d
"request_type=usedNetworkAddresses&client_name=DEFAULT&ip=10.20.10.0&no_csv=yes"
<?xml version='1.0' encoding='UTF-8'?>
<usedNetworkAddressesResult>
  <error>
</error>
  <Network>
    <IP>10.20.10.0</IP>
    <HostList>
      <Host>
        <id>2625</id>
        <IP>10.20.10.1</IP>
        <hostname>host1</hostname>
        <descr></descr>
        <site>Lon1</site>
        <cat>FW</cat>
        <int_admin>n</int_admin>
```

```

<comment></comment>
<update_type>NULL</update_type>
<alive>up</alive>
<last_response>20/02/2019 20:06:26</last_response>
<ip_version>v4</ip_version>
<customColumns>
  <vendor>Juniper</vendor>
</customColumns>
</Host>
<Host>
  <id>2626</id>
  <IP>192.168.1.10</IP>
  ...

```

2.7 Working with VLANs

The API allows to create, read, update, delete, query the first free and list VLANs.

2.7.1 Create VLANs

Use request_type “createVlan” to create VLANs.

Required attributes:

request_type=createVlan

client_name=*CLIENT_NAME*

new_number=*VLAN_NUMBER* (number between 1 and 4095)

new_name=*VLAN NAME*

Optional attributes:

new_bg_color=LightCyan|LightBlue|dodgerblue|LimeGreen|SeaGreen|pink|white|black|brown|gold|red|DarkOrange

new_font_color=LightCyan|LightBlue|dodgerblue|LimeGreen|SeaGreen|pink|white|black|brown|gold|red|DarkOrange

new_comment=*NEW VLAN COMMENT*

new_provider=*NEW PROVIDER NAME*

Example:

```

$ curl --url "http://localhost/gestioip/api/api.cgi" -u gipadmin:xxxxxx -d
"request_type=createVlan&client_name=DEFAULT&new_number=537&new_name=new
VLAN&new_comment=TEST VLAN"

```



```
<?xml version='1.0' encoding='UTF-8'?>
<createVlanResult>
  <error>
  </error>
  <vlan>
    <new_id>347</new_id>
    <new_number>537</new_number>
    <new_name>new VLAN</new_name>
    <new_comment>TEST VLAN</new_comment>
    <new_provider></new_provider>
    <new_font_color>black</new_font_color>
    <new_bg_color>white</new_bg_color>
  </vlan>
</createVlanResult>
```

2.7.2 Read VLAN information

Use request_type “readVlan” to obtain all attributes of a VLAN.

Required attributes:

request_type=readVlan

client_name=*CLIENT_NAME*

vlan_id=*VLAN_ID* or

vlan_number=*VLAN_NUMBER* or

vlan_name=*VLAN NAME*

If vlan_number or vlan_name are ambiguous, this two attributes can be combined.

Example:

```
$ curl --url "http://localhost/gestioip/api/api.cgi" -u gipadmin:xxxxxx -d
"request_type=readVlan&client_name=DEFAULT&vlan_number=537"
```

```
<?xml version='1.0' encoding='UTF-8'?>
<readVlanResult>
  <error>
  </error>
  <vlan>
    <id>347</id>
    <number>537</number>
    <name>new VLAN</name>
    <comment>TEST VLAN</comment>
    <provider></provider>
    <font_color>black</font_color>
    <bg_color>white</bg_color>
  </vlan>
</readVlanResult>
```

2.7.3 Update VLANs

Use request_type “updateVlan” to update VLAN attributes.

Required attributes:

request_type=updateVlan

client_name=*CLIENT_NAME*

vlan_id=*VLAN_ID*

Optional attributes:

new_number=*VLAN_NUMBER*

new_name=*VLAN NAME*

new_bg_color=LightCyan|LightBlue|dodgerblue|LimeGreen|SeaGreen|pink|white|black|brown|red|DarkOrange

new_font_color=LightCyan|LightBlue|dodgerblue|LimeGreen|SeaGreen|pink|white|black|brown|red|DarkOrange

new_comment=*NEW VLAN COMMENT*

new_provider=*NEW PROVIDER NAME*

Example:

```
$ curl --url "http://localhost/gestioip/api/api.cgi" -u gipadmin:xxxxxx -d
"request_type=updateVlan&client_name=DEFAULT&vlan_id=347&new_comment=&new_font_c
olor=brown"
```

```
<?xml version='1.0' encoding='UTF-8'?>
<updateVlanResult>
  <error>
  </error>
  <vlan>
    <id>347</id>
    <new_number>537</new_number>
    <new_name>new VLAN</new_name>
    <new_comment></new_comment>
    <new_provider></new_provider>
    <new_font_color>brown</new_font_color>
    <new_bg_color>white</new_bg_color>
  </vlan>
</updateVlanResult>
```

2.7.4 Delete VLANs

Use request_type “deleteVlan” to delete VLANs.

Required attributes:

```
request_type=deleteVlan
client_name=CLIENT_NAME
vlan_id=VLAN_ID
```

Example:

```
$ curl --url "http://localhost/gestioip/api/api.cgi" -u gipadmin:xxxxxx -d
"request_type=deleteVlan&client_name=DEFAULT&vlan_id=347"
```

```
<?xml version='1.0' encoding='UTF-8'?>
<deleteVlanResult>
  <error>
  </error>
  <vlan>
  </vlan>
</deleteVlanResult>
```

2.7.5 Get the first free VLAN number

Use request_type “firstFreeVlan” to obtain the first free VLAN number.

Required attributes:

```
request_type=firstFreeVlan
client_name=CLIENT_NAME
```

Optional attributes:

```
vlan_provider_name=VLAN_PROVIDER_NAME
```

The attribute vlan_provider_name offers the possibility to request a free VLAN number from a dedicated provider (pool). This implies that there is only one pool of VLAN numbers per provider defined and that the pool start and end number are defined as VLAN (customer-dms-end-range).

Example:

If there are less than two VLANs for a provider defined the query will fail. If there are two or more VLANs for a provider defined, the request will return the next free VLAN number for this provider.

API Guide GestióIP IPAM v3.5

```
mysql> select * FROM vlans where provider_id = 7;
+----+-----+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+-----+
| id | vlan_num | vlan_name          | comment          | provider_id | bg_color | font_color |
+-----+-----+-----+-----+-----+-----+-----+
| 50 | 100      | customer100-dmz    |                  | 7           | white    | black      |
|    |          | NULL              | 4               |             |          |            |
+-----+-----+-----+-----+-----+-----+-----+
| 51 | 101      | customer101-dmz    |                  | 7           | white    | black      |
|    |          | NULL              | 4               |             |          |            |
+-----+-----+-----+-----+-----+-----+-----+
| 52 | 109      | customer102-dmz    |                  | 7           | white    | black      |
|    |          | NULL              | 4               |             |          |            |
+-----+-----+-----+-----+-----+-----+-----+
| 56 | 199      | customer-dmz-end-range | End Range Customer DMZ | 7           | white    | black      |
|    |          | NULL              | 4               |             |          |            |
+-----+-----+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)
```

With the following request the API will return a VLAN number “102” which is the first free VLAN for provider cust-dmz (id=7).

```
http://server/gestioip/api/api.cgi?
request_type=firstFreeVlan&client_name=lotr&vlan_provider_name=cust-dmz
```

Example:

```
$ curl --url "http://localhost/gestioip/api/api.cgi" -u gipadmin:xxxxxx -d
"request_type=firstFreeVlan&client_name=DEFAULT"
```

```
<?xml version='1.0' encoding='UTF-8'?>
<getFreeVLANResult>
  <error>
  </error>
  <VLAN>
    <number>56</number>
  </VLAN>
</getFreeVLANResult>
```

2.7.6 List VLANs

Use request_type “listVlans” to obtain a list of all VLANs or all VLAN of a specific client. If “ALL” is specified as “client_name”, the client name will be added to each VLAN.

Required attributes:

```
request_type=listVlans
```

```
client_name=CLIENT_NAME (use “ALL” as client_name to obtain a list of all VLANs)
```

Example:

```
$ curl --url "https://localhost/gestioip/api/api.cgi" -u gipadmin:xxxxxx -d
"request_type=listVlans&client_name=DEFAULT"
<?xml version='1.0' encoding='UTF-8'?>
<listVlansResult>
  <error>
</error>
  <VlanList>
    <Vlan>2,1,Default,default VLAN,,white,black</Vlan>
    <Vlan>3,100,server1,,white,black</Vlan>
    <Vlan>5,101,server2,TEST VLAN,,white,black</Vlan>
    <Vlan>9,200,intern,,white,black</Vlan>
    <Vlan>10,300,dmz,server DMZ,,gold,black</Vlan>
  </VlanList>
</listVlansResult>
<getFreeVLANResult>
```

2.8 Working with VLAN Providers

The API allows to create, read, update and delete VLAN Providers.

2.8.1 Create VLAN Provider

Use request_type “createVlanProvider” to create VLAN Providers.

Required attributes:

request_type=createVlanProvider

client_name=*CLIENT_NAME*

new_name=*VLAN_PROVIDER_NAME*

Optional attributes:

new_comment=*VLAN_PROVIDER_COMMENT*

Example:

```
$ curl --url "http://localhost/gestioip/api/api.cgi" -u gipadmin:xxxxxx -d
"request_type=createVlanProvider&client_name=DEFAULT&new_name=TEST VLAN
Provider&new_comment=TEST VLAN provider comment"

<?xml version='1.0' encoding='UTF-8'?>
<createVlanProviderResult>
  <error>
</error>
  <vlanProvider>
```

```

    <new_id>26</new_id>
    <new_name>TEST VLAN Provider</new_name>
    <new_comment>TEST VLAN Provider comment</new_comment>
  </vlanProvider>
</createVlanProviderResult>

```

2.8.2 Read VLAN Provider information

Use request_type “readVlanProvider” to obtain all attributes of a VLAN Provider.

Required attributes:

request_type=readVlanProvider

client_name=*CLIENT_NAME*

id=*VLAN_PROVIDER_ID* or

name=*VLAN_PROVIDER_NAME*

Example:

```
$ curl --url "http://localhost/gestioip/api/api.cgi" -u gipadmin:xxxxxx -d
"request_type=readVlanProvider&client_name=DEFAULT&name=TEST VLAN Provider"
```

```

<?xml version='1.0' encoding='UTF-8'?>
<readVlanProviderResult>
  <error>
  </error>
  <vlanProvider>
    <id>26</id>
    <name>TEST VLAN Provider</name>
    <comment>TEST VLAN Provider comment</comment>
  </vlanProvider>
</readVlanProviderResult>

```

2.8.3 Update VLAN Providers

Use request_type “updateVlanProvider” to update VLAN Provider attributes.

Required attributes:

request_type=updateVlanProvider

client_name=*CLIENT_NAME*

id=*VLAN_PROVIDER_ID* or

name=*VLAN_PROVIDER_NAME*

Optional attributes:

`new_name=NEW VLAN PROVIDER NAME`

`new_comment=NEW VLAN PROVIDER COMMENT`

Example:

```
$ curl --url "http://localhost/gestioip/api/api.cgi" -u gipadmin:xxxxxx -d
"request_type=updateVlanProvider&client_name=DEFAULT&id=26&new_name=New
Name&new_comment="
```

```
<?xml version='1.0' encoding='UTF-8'?>
<updateVlanProviderResult>
  <error>
  </error>
  <vlanProvider>
    <id>347</id>
    <new_name>New Name</new_name>
    <new_comment></new_comment>
  </vlanProvider>
</updateVlanProviderResult>
```

2.8.4 Delete VLAN Providers

Use `request_type` “deleteVlanProvider” to delete VLAN Providers.

Required attributes:

`request_type=deleteVlanProvider`

`client_name=CLIENT_NAME`

`id=VLAN_ID` or

`name=VLAN_PROVIDER_NAME`

Example:

```
$ curl --url "http://localhost/gestioip/api/api.cgi" -u gipadmin:xxxxxx -d
"request_type=deleteVlanProvider&client_name=DEFAULT&id=26"
```

```
<?xml version='1.0' encoding='UTF-8'?>
<deleteVlanProviderResult>
  <error>
  </error>
  <vlanProvider>
  </vlanProvider>
</deleteVlanProviderResult>
```

2.8.5 List VLAN Providers

Use request_type “listVlanProviders” to obtain a list of all VLAN providers or all VLAN Providers of a specific client. If “ALL” is specified as “client_name”, the client name will be added to each VLAN.

Required attributes:

request_type=listVlans

client_name=*CLIENT_NAME* (use “ALL” as client_name to obtain a list of all VLANs)

Optional attributes:

no_csv=yes|no

Default behavior is to display the host entries in form of a CSV list (see example 1). Setting no_csv to “yes” forces a full XML/JSON output. The individual host entries would be also displayed in XML/JSON style (see example 2). Default: no

limit=*integer between 1 and 9999*

Limit the number of resulting networks

page=*integer between 0 and 9999*

Shows page *n* of a limited output (starting with 0). Only together with the “limit” option.

2.9 Working with Clients

The API allows to create, read, update, delete and list Clients.

2.9.1 Create a new Client

Use request_type “createClient” to create a new client.

Required attributes:

request_type=createClient

new_name=*NEW_CLIENT_NAME*

new_site=*COMA SEPARATED LIST OF NEW SITES*

Optional attributes:

new_phone=*PHONE NUMBER*

new_fax=*FAX NUMBER*

new_address=*ADDRESS*

new_comment=*COMMENT*

new_contact_name_1=*NAME*

new_contact_phone_1=*PHONE NUMBER*

new_contact_cell_1=*CELL PHONE NUMBER*

new_contact_comment_1=*COMMENT*

new_contact_name_2=*NAME*

new_contact_phone_2=*PHONE NUMBER*

new_contact_cell_2=*CELL PHONE NUMBER*

new_contact_comment_2=*COMMENT*

new_contact_name_3=*NAME*

new_contact_phone_3=*PHONE NUMBER*

new_contact_cell_3=*CELL PHONE NUMBER*

new_contact_comment_3=*COMMENT*

Example:

```
curl --url "http://localhost/gestioip/api/api.cgi" -u gipadmin:xxxxxxxx -d
"request_type=createClient&new_name=CLIENT_2&new_site=Site1,Site2,Site3&new_phon
e=12345678&new_fax=22334455&new_address=Gran Via 1, 08022 Barcelona,
```

```
SPAIN&new_contact_name_1=John Johnson&new_contact_name_2=Clara
Wine&&new_contact_phone_2=123123123&new_contact_email_2=cwine@example.com"
```

```
<?xml version='1.0' encoding='UTF-8'?>
```

```
<createClientResult>
  <error>
</error>
  <client>
    <id>7</id>
    <new_name>CLIENT_2</new_name>
    <new_site>Site1,Site2,Site3</new_site>
    <new_phone>12345678</new_phone>
    <new_fax>22334455</new_fax>
    <new_comment></new_comment>
    <new_contact_name_1>John Johnson</new_contact_name_1>
    <new_contact_phone_1></new_contact_phone_1>
    <new_contact_cell_1></new_contact_cell_1>
    <new_contact_email_1></new_contact_email_1>
    <new_contact_comment_1></new_contact_comment_1>
    <new_contact_name_2>Clara Wine</new_contact_name_2>
    <new_contact_phone_2>123123123</new_contact_phone_2>
    <new_contact_cell_2></new_contact_cell_2>
    <new_contact_email_2>cwine@example.com</new_contact_email_2>
    <new_contact_comment_2></new_contact_comment_2>
    <new_contact_name_3></new_contact_name_3>
    <new_contact_phone_3></new_contact_phone_3>
    <new_contact_cell_3></new_contact_cell_3>
    <new_contact_email_3></new_contact_email_3>
    <new_contact_comment_3></new_contact_comment_3>
  </client>
</createClientResult>
```

2.9.2 Read Client

Use request_type “readClient” to create a new client.

Required attributes:

request_type=readClient

client_name=*CLIENT_NAME*

Example:

```
curl --url "http://localhost/gestioip/api/api.cgi" -u gipadmin:xxxxxxx -d
"request_type=readClient&client_name=CLIENT_2"<?xml version='1.0' encoding='UTF-
8'?>
```

```
<readClientResult>
  <error>
</error>
  <client>
    <client>CLIENT_2</client>
    <phone>12345678</phone>
    <fax>22334455</fax>
    <address>Gran Via 1, 08022 Barcelona, SPAIN</address>
    <comment></comment>
    <contact_name_1>John Johnson</contact_name_1>
    <contact_phone_1></contact_phone_1>
    <contact_cell_1></contact_cell_1>
    <contact_email_1></contact_email_1>
    <contact_comment_1></contact_comment_1>
    <contact_name_2>Clara Wine</contact_name_2>
    <contact_phone_2>123123123</contact_phone_2>
    <contact_cell_2></contact_cell_2>
    <contact_email_2>cwine@example.com</contact_email_2>
    <contact_comment_2></contact_comment_2>
    <contact_name_3></contact_name_3>
    <contact_phone_3></contact_phone_3>
    <contact_cell_3></contact_cell_3>
    <contact_email_3></contact_email_3>
    <contact_comment_3></contact_comment_3>
    <default_resolver>yes</default_resolver>
    <dns_server_1></dns_server_1>
    <dns_server_2></dns_server_2>
    <dns_server_3></dns_server_3>
  </client>
</readClientResult>
```

2.9.3 Update a Client

Use request_type “updateClient” to update an existing client.

Required attributes:

request_type=updateClient

client_name=CLIENT_NAME

Optional attributes:

new_name=NEW_CLIENT_NAME

new_phone=PHONE NUMBER

new_fax=FAX NUMBER

new_address=ADDRESS

new_comment=COMMENT

new_contact_name_1=NAME

new_contact_phone_1=PHONE NUMBER

new_contact_cell_1=CELL PHONE NUMBER

new_contact_comment_1=COMMENT

new_contact_name_2=NAME

new_contact_phone_2=PHONE NUMBER

new_contact_cell_2=CELL PHONE NUMBER

new_contact_comment_2=COMMENT

new_contact_name_3=NAME

new_contact_phone_3=PHONE NUMBER

new_contact_cell_3=CELL PHONE NUMBER

new_contact_comment_3=COMMENT

Example:

```
curl --url "http://localhost/gestioip/api/api.cgi" -u gipadmin:xxxxxxxx -d
"request_type=updateClient&client_name=CLIENT_2&new_phone=333333333&new_contact_
email_1=jjohnson@example.com"
<?xml version='1.0' encoding='UTF-8'?>
<updateClientResult>
  <error>
  </error>
```

```

<client>
  <id>7</id>
  <new_name></new_name>
  <new_phone>333333333</new_phone>
  <new_fax>22334455</new_fax>
  <new_comment></new_comment>
  <new_contact_name_1>John Johnson</new_contact_name_1>
  <new_contact_phone_1></new_contact_phone_1>
  <new_contact_cell_1></new_contact_cell_1>
  <new_contact_email_1>jjohnson@example.com</new_contact_email_1>
  <new_contact_comment_1></new_contact_comment_1>
  <new_contact_name_2>Clara Wine</new_contact_name_2>
  <new_contact_phone_2>123123123</new_contact_phone_2>
  <new_contact_cell_2></new_contact_cell_2>
  <new_contact_email_2>cwine@example.com</new_contact_email_2>
  <new_contact_comment_2></new_contact_comment_2>
  <new_contact_name_3></new_contact_name_3>
  <new_contact_phone_3></new_contact_phone_3>
  <new_contact_cell_3></new_contact_cell_3>
  <new_contact_email_3></new_contact_email_3>
  <new_contact_comment_3></new_contact_comment_3>
</client>
</updateClientResult>

```

2.9.4 Delete a Client

Use request_type “deleteClient” to delete clients.

Required attributes:

request_type=deleteClient

client_name=CLIENT_NAME

Example:

```

curl --url "http://localhost/gestioip/api/api.cgi" -u gipadmin:hola123 -d
"request_type=deleteClient&client_name=CLIENT_2"<?xml version='1.0'
encoding='UTF-8'?>
<deleteClientResult>

```

```

    <error>
  </error>
</deleteClientResult>

```

The absence of an error means that the client was deleted successfully.

2.9.5 List Clients

Use request_type “listClients” to list all existing clients. If you specify the *client_name* attribute, only the specified client will be shown.

Required attributes:

request_type=listClients

Optional attributes:

client_name=*CLIENT_NAME*

limit=*integer between 1 and 9999*

Limit the number of resulting networks

page=*integer between 0 and 9999*

Shows page n of a limited output (starting with 0). Only together with the “limit” option.

Example:

```

curl --url "https://localhost/gestioip/api/api.cgi" -u gipadmin:xxxxxx -d
"request_type=listClients&client_name=DEFAULT"
<?xml version='1.0' encoding='UTF-8'?>
<listClientsResult>
  <error>
</error>
  <client>
    <client>DEFAULT</client>
    <phone></phone>
    <fax></fax>
    <address></address>
    <comment></comment>
    <contact_name_1></contact_name_1>
    <contact_phone_1></contact_phone_1>
    <contact_cell_1></contact_cell_1>
    <contact_email_1></contact_email_1>
    <contact_comment_1></contact_comment_1>
    <contact_name_2></contact_name_2>
    <contact_phone_2></contact_phone_2>
    <contact_cell_2></contact_cell_2>
    <contact_email_2></contact_email_2>

```

```
<contact_comment_2></contact_comment_2>
<contact_name_3></contact_name_3>
<contact_phone_3></contact_phone_3>
<contact_cell_3></contact_cell_3>
<contact_email_3></contact_email_3>
<contact_comment_3></contact_comment_3>
<default_resolver>yes</default_resolver>
<dns_server_1></dns_server_1>
<dns_server_2></dns_server_2>
<dns_server_3></dns_server_3>
</client>
</listClientsResult>
```

2.10 Discovery

The API permits to execute the following discovery functions:

- Network discovery via SNMP
- Host discovery via DNS
- Host discovery via SNMP

It will start the corresponding script and will return the PID.

2.10.1 Network discovery via SNMP

For a network discovery via SNMP, the API executes the script `/usr/share/gestioip/bin/get_networks_snmp.pl`

Required attributes:

`request_type=runNetworkDiscovery`

Optional attributes:

`client_name=CLIENT_NAME`

`add_if_descr=[yes|no]`

`ascend=[yes|no]`

`config_file_name=CONFIG_FILE_NAME (without path)`

`descend=[yes|no]`

`force_site=SITE`

`get_vrf_route=[yes|no]`

`if_descr_ident=[Alias|Descr]`

`import_host_routes=[yes|no]`

`mail=[yes|no]`

`mail_sum=[yes|no]`

`nodes_file=NODES_FILE_NAME (without path)`

`output_type=xml|json`

`route_type=CSV-LIST OF THE ROUTE TYPES [local|static|other|ospf|rip|isis|eigrp|netmgmt|icmp|egp|ggp|hello|isis|cisco|igrp|lbn|spf|igmp|bgp|ldp]`

`set_sync_flag=[yes|no]`

verbose=[yes|no]

write_new_to_file=FILENAME

For an explication of the command arguments have a look at the GestióIP User Guide or execute “/usr/share/gestioip/bin/get_networks_snmp.pl --help”.

Example:

```
$ curl --url "https://localhost/gestioip/api/api.cgi" -u gipadmin:xxxxxx -d
"request_type=runNetworkDiscovery&client_name=DEFAULT&descend=yes"
<?xml version='1.0' encoding='UTF-8'?>
<runHostDiscoverySNMPResult>
  <error>
  </error>
  <childProcess>
    <pid>7100</pid>
    <command>/usr/share/gestioip/bin/get_networks_snmp.pl</command>
  </childProcess>
</runHostDiscoverySNMPResult>
```

2.10.2 Host discovery via DNS

For a host discovery via SNMP, the API executes the script /usr/share/gestioip/bin/ip_update_gestioip_dns.pl

Required attributes:

request_type=runHostDiscoveryDNS

Optional attributes:

client_name=CLIENT_NAME

config_file_name=CONFIG_FILE_NAME (without path)

CSV_networks=NETWORK-LIST

disable_audit=[yes|no]

ignore_dns=[yes|no]

location=LOCATION-LIST

mail=[yes|no]

mail_sum=[yes|no]

network_file=NETWORK_FILE_NAME (without path)

```
output_type=xml|json
range=IP-RANGE
verbose=[yes|no]
```

For an explication of the command arguments have a look at the GestióIP User Guide or execute “/usr/share/gestioip/bin/ip_update_gestioip_dns --help”.

Example:

```
$ curl --url "https://localhost/gestioip/api/api.cgi" -u gipadmin:hola123 -d
"request_type=runHostDiscoveryDNS&client_name=DEFAULT&range=192.168.1.10-
192.168.1.20"
<?xml version='1.0' encoding='UTF-8'?>
<runHostDiscoveryDNSResult>
  <error>
  </error>
  <childProcess>
    <pid>8778</pid>
    <command>/usr/share/gestioip/bin/ip_update_gestioip_dns.pl</command>
  </childProcess>
</runHostDiscoveryDNSResult>
```

2.10.3 Host discovery via SNMP

For a host discovery via SNMP, the API executes the script /usr/share/gestioip/bin/ip_update_gestioip_snmp.pl

Required attributes:

```
request_type=runHostDiscoverySNMP
```

Optional attributes:

```
client_name=CLIENT_NAME
```

```
config_file_name=CONFIG_FILE_NAME (without path)
```

```
CSV_networks=NETWORK-LIST
```

```
disable_audit=[yes|no]
```

```
ignore_arp_cache=[yes|no]
```

```
location=LOCATION-LIST
```

```
mail=[yes|no]
```

mail_sum=[yes|no]
network_file=NETWORK_FILE_NAME (without path)
output_type=xml|json
range=IP-RANGE
verbose=[yes|no]

For an explication of the command arguments have a look at the GestióIP User Guide or execute “/usr/share/gestioip/bin/ip_update_gestioip_snmp --help”.

Example:

```
$ curl --url "https://localhost/gestioip/api/api.cgi" -u gipadmin:xxxxxx -d  
"request_type=runHostDiscoverySNMP&client_name=DEFAULT&CSV_networks=10.1.2.0/24,  
172.16.40.0/26&verbos=yes"  
<?xml version='1.0' encoding='UTF-8'?>  
<runHostDiscoveryDNSResult>  
  <error>  
  </error>  
<childProcess>  
  <pid>9473</pid>  
  <command>/usr/share/gestioip/bin/ip_update_gestioip_snmp.pl</command>  
</childProcess>  
</runHostDiscoveryDNSResult>
```

3 Usage examples

3.1 Example Powershell

```

$username = 'gipadmin'
$password = 'XXXXXX'

$url = 'http://gestioip.example.com/gestioip/api/api.cgi'
$client = 'DEFAULT'

$auth = $username + ':' + $password
$encoded = [System.Text.Encoding]::UTF8.GetBytes($auth)
$encodedPassword = [System.Convert]::ToBase64String($encoded)
$headers = @{"Authorization"="Basic $($encodedPassword)"}

## Query first free address of network 10.0.0.0 and create a new host entry (see API Guide chapter
2.5.9)

$requestType="reserveFirstFreeNetworkAddress"
$params("&ip=10.0.0.0&new_hostname=server123&new_descr=description of server123")

$r = Invoke-WebRequest -Uri ($url + "?request_type=$requestType&client_name=$client$params") -Header
$headers -Method Get
$bn = ($r.Content)
echo $bn
$bn_error = ([xml]$r.Content).reserveFirstFreeNetworkAddressResult.error.string

if ([string]::IsNullOrEmpty($bn_error)) {
    # get IP of new host
    $bn_ip = ([xml]$r.Content).reserveFirstFreeNetworkAddressResult.Host.IP
    echo "Request successful: host created: $bn_ip"
} else {
    echo $bn_error
}

## Delete the host which was just created (see API Guide chapter 2.4.4)

$requestType="deleteHost"
$params("&ip=$bn_ip")

$r = Invoke-WebRequest -Uri ($url + "?request_type=$requestType&client_name=$client$params") -Header
$headers -Method Get

```

API Guide GestióIP IPAM v3.5

```
$bn = ($r.Content)
echo $bn
$bn_error = ([xml]$r.Content).deleteHostResult.error.string

if ([string]::IsNullOrEmpty($bn_error)) {
    echo "Request successful: host deleted: $bn_ip"
} else {
    echo $bn_error
}

## Read network information (see User Guide chapter 2.5.2)

$RequestType="readNetwork"
$params("&ip=10.0.0.0"

$r = Invoke-WebRequest -Uri ($url + "?request_type=$RequestType&client_name=$Client$params") -Header
$headers -Method Get
$bn = ($r.Content)

$bn_error = ([xml]$r.Content).readNetworkResult.error.string

if ([string]::IsNullOrEmpty($bn_error)) {
    echo "Request successful:"
    echo $bn
} else {
    echo $bn_error
}
```

3.2 Example Ansible Playbook

Here an example for an Ansible Playbook to create a new network.

```
- name: Create Network in GestioIP
  uri:
    url: https://gestioip.server.com/api/api.cgi
    method: POST
    user: your_username
    password: your_pass
    body:
"request_type=createNetwork&client_name=DEFAULT&ip=192.168.22.0&new_BM=24&new_de
scr=Test Network"
    status_code: 200, 201
```