

GestióIP IPAM

v3.2

IP address management software

API Guide

v0.6

www.gestioip.net

Table of Contents

1 Introduction.....	3
2 API usage.....	3
2.1 Accessing the API.....	3
2.1.1 Passing parameters to the API.....	3
2.2 Getting help.....	3
2.3 Output format.....	4
2.4 Working with Hosts.....	6
2.4.1 Create hosts.....	6
2.4.2 Read host information.....	7
2.4.3 Update hosts.....	8
2.4.4 Delete hosts.....	9
2.4.5 List hosts.....	10
2.5 Working with Networks.....	11
2.5.1 Create networks.....	11
2.5.2 Read network information.....	12
2.5.3 Update networks.....	12
2.5.4 Delete networks.....	13
2.5.5 List Networks.....	14
2.5.6 Get the first free network within a root network.....	15
2.5.7 List all free IP addresses of a network.....	16
2.5.8 Show the first free address of a network.....	17
2.6 Working with VLANs.....	18
2.6.1 Create Vlans.....	18
2.6.2 Read VLAN information.....	19
2.6.3 Update VLANs.....	19
2.6.4 Delete VLANs.....	20
2.6.5 Get the first free VLAN number.....	21
2.7 Working with VLAN Providers.....	22
2.7.1 Create VLAN Provider.....	22
2.7.2 Read VLAN Provider information.....	23
2.7.3 Update VLAN Providers.....	23
2.7.4 Delete VLAN Providers.....	24

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 address of a network, list all free IP addresses of a network

Vlans:

create, read, update, delete

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 support 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|readNetwork|readNetworkHelp|
updateNetwork|updateNetworkHelp|createNetwork|createNetworkHelp|deleteNetwork|
deleteNetworkHelp|listNetworks|listNetworksHelp|firstFreeNetwork|
firstFreeNetworkHelp|freeNetworkAddresses|freeNetworkAddressesHelp|
firstFreeNetworkAddress|firstFreeNetworkAddressHelp|readVlan|readVlanHelp|
updateVlan|updateVlanHelp|createVlan|createVlanHelp|deleteVlan|deleteVlanHelp|
firstFreeVlan|firstFreeVlanHelp|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>insertVlan</request_type>
    <client_name>CLIENT_NAME</client_name>
```

API Guide GestióIP IPAM v3.2

```
<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 Working with Hosts

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

2.4.1 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=yln
 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>
    <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>

```

2.4.2 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>
    <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>
    </customColumns>
  </Host>
</readHostResult>

```

```

        <OS></OS>
        <URL></URL>
        <linked IP></linked IP>
        <vendor>cisco</vendor>
    </customColumns>
</Host>
</readHostResult>

```

2.4.3 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>
    <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>
  </Host>
</UpdateHostResult>

```



```

<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>

```

2.4.4 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>

```

2.4.5 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:

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::prod

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

Example:

```
$ 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>
```

2.5 Working with Networks

The API allows to create, read, update and delete networks.

2.5.1 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_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>

```

2.5.2 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>
    <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>

```

2.5.3 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>
    <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>
```

2.5.4 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:xxxxxxx -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:xxxxxxx -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>
```

2.5.5 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)

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.

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

Example:

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

2.5.6 Get the first free network within a root network

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

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&rootnet_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.

2.5.7 List all free IP addresses of a network

Use request_type “freeNetworkAddresses” to list all free IP addresses.

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>
```

2.5.8 Show the first free address of a network

Use request_type “firstFreeNetworkAddress” 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 Working with VLANs

The API allows to create, read, update and delete VLANs.

2.6.1 Create Vlan

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|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=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>
```

2.6.2 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>
```

2.6.3 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_vlan_comment=&new_font_color=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>
```

2.6.4 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>
```

2.6.5 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.

```
mysql> select * FROM vlans where provider_id = 7;
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| id | vlan_num | vlan_name          | comment          | provider_id | bg_color | font_color | switches | asso_vlan | client_id |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 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 Working with VLAN Providers

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

2.7.1 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>
```

2.7.2 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>
```

2.7.3 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>
```

2.7.4 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>
```