External API

From:
https://pandorafms.com/manual/!current/
Permanent link:
https://pandorafms.com/manual/!current/en/documentation/pandorafms/technical_reference/02_annex_externalapi
2024/06/10 14:36

External API

Pandora FMS External API

The Pandora FMS External API is used doing remote calls (through HTTP or HTTPS) on the `/include/api.php` file. This is the method that has been defined in Pandora FMS to integrate applications from third parties to Pandora FMS. It basically consists on a call with the parameters formatted to receive a value or a list of values that after its application will be used to do operations.

A call to the API is as simple as this:

```
http://<Pandora FMS Console install >/include/api.php<parameters >
```

The API can only receive the following parameters:

- op (required): It is the first parameter that specifies the nature of the operation, which could be `get` or `set` or `help`:
  - `get`: It returns a value or values.
  - `set`: It sends a value or values.
  - `help`: It returns a little help from the calls
- op2 (required): The call with an explanatory name of the one that it works on.
- `id` (*depends on the command*): The first call parameter.
- `id2` (*depends on the command*): The second call parameter.
- `other` (*depends on the command*): The third call parameter, sometimes it could be a list of serial values.
- `other_mode` (*depends on the command*): Serial format, list of posible values:
  - `url_encode`: the othervalue is an alphanumeric formatted as UrlEncode.
  - `url_encode_separator_`<separator>:The value will be a serial value list with the divider character, for example:

```
...other=pears|melons|watermelon&other_mode=url_encode_separator_|
```

You can use any separator, in this documentation the **|**
character (%7C *percent-encoded*) is used preferentially.
See also how works the API checker.

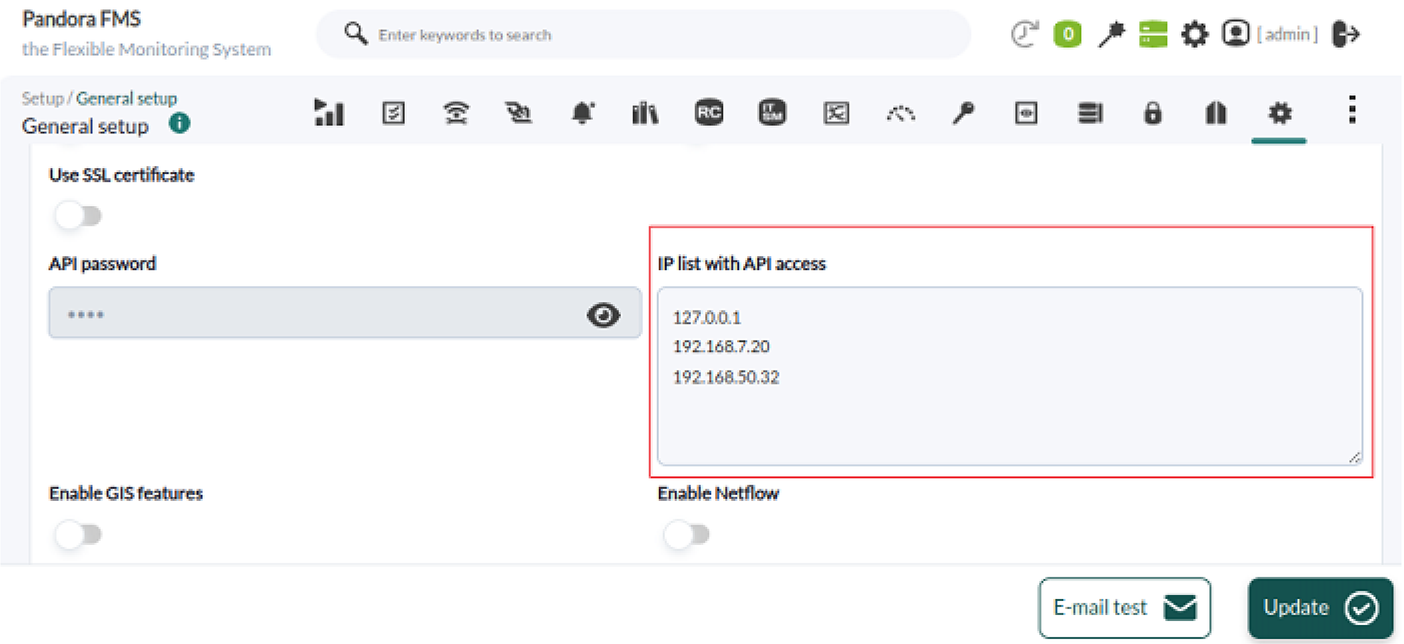
- `returnType` (*depends on the command*): return format of the value or values. Any of current available values:
  - `string`: It returns the value as it is as an alphanumeric one.
  - `csv`: It returns the values as a CSV separated with the `;` character (fields) and with CR (files) by default.
  - `csv_head`: It returns the same as with `csv`, except that it adds a first file with the field names to return.
- Security credentials: see “Security” section.

Security

The API authenticates access via API source and/or password and/or user credentials.

By origin

At the moment, security is based on an IP addresses list that will have access to the tool.



IP list with API access:

```
127.0.0.1
192.168.7.20
192.168.50.32
```

If you enter the character * in the box text, the ACL check will be omitted relegating the security to the protocol and to the environment.

By API password

You can also set a password for API actions (API password).

- `apipass`: By default, and used in this documentation, 1234.

By user credentials

Access is by user name and password; additionally from version 768 or later it is possible to authenticate by bearer token.

By user name and password

To access API actions, it is necessary to give a Pandora FMS valid user name and password.

- `user`: Valid Pandora FMS user; `admin` is used in this documentation.
- `pass`: The user password; `pandora` is used in this documentation.

In API calls, passwords are uncoded. Be careful and use SSL connections to avoid sniffers from third party. The API allows POST petitions to encrypt them when using SSL/HTTPS.

Return

When the API denies the access, the simple string `auth error` is returned.

Examples

In this documentation the API password `1234` is used, the user name is `admin` (superadmin type) and the user password is `pandora`.

Pandora FMS has installed, by default, a user called `internal_API` which has neither a password nor a superadmin. For this user, as well as for the others that are created, the appropriate profiles should be established for each one of the commands and operations described here.

See call syntax:

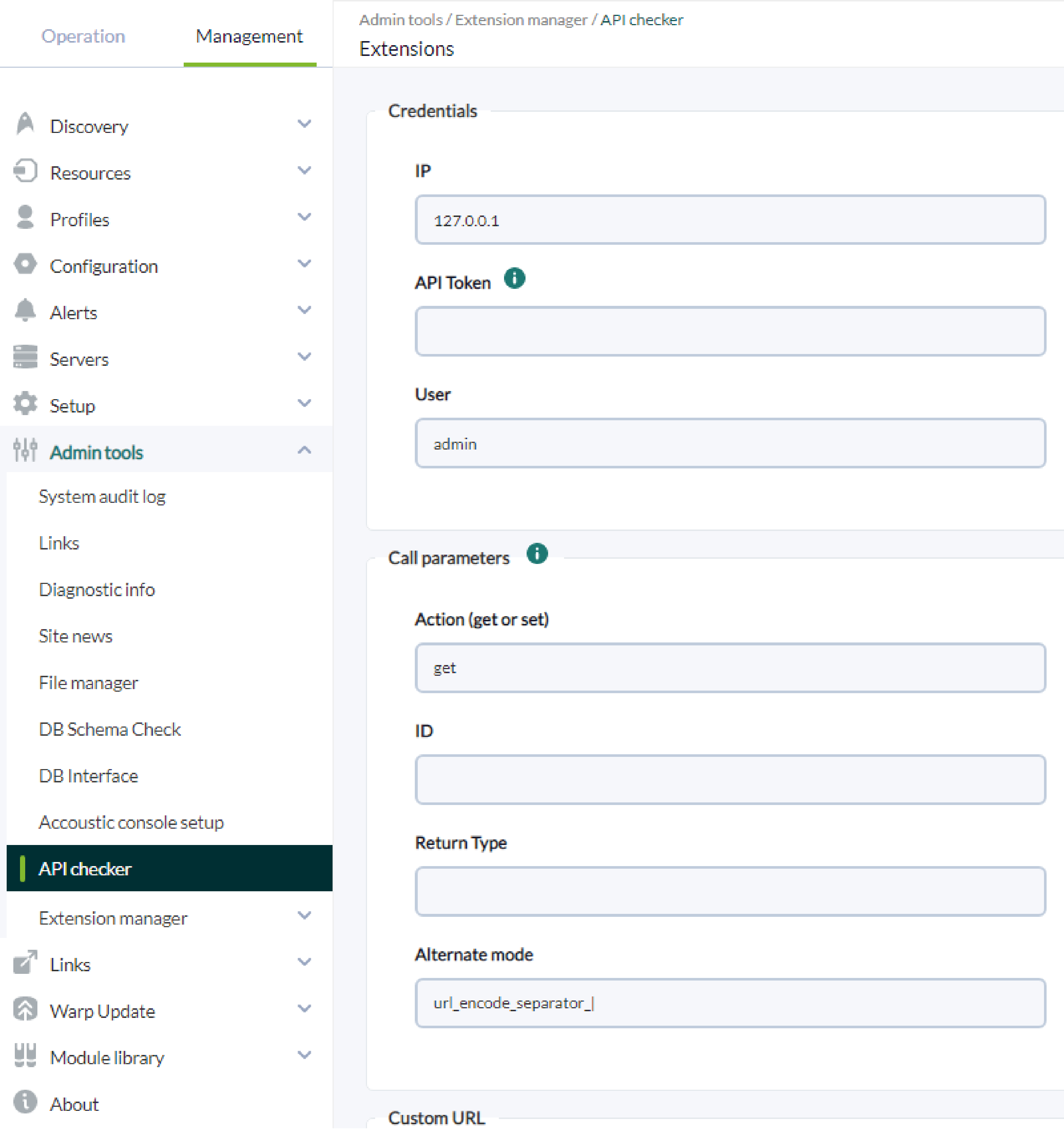
```
.../include/api.php?op=get&op2=plugins&return_type=csv&other=;&apipass=1234&user
=admin&pass=pandora
```

Access conditions:

- The source IP address is in the ACL IP addresses list (or is used *).
- The API password is *not set* or is `1234`.
- The user `admin` exists and is a superadmin user, and its password is `pandora`.

API checker

Menu Management → Admin tools → API checker.



Admin tools / Extension manager / API checker

Extensions

Credentials

IP

127.0.0.1

API Token

User

admin

Call parameters

Action (get or set)

get

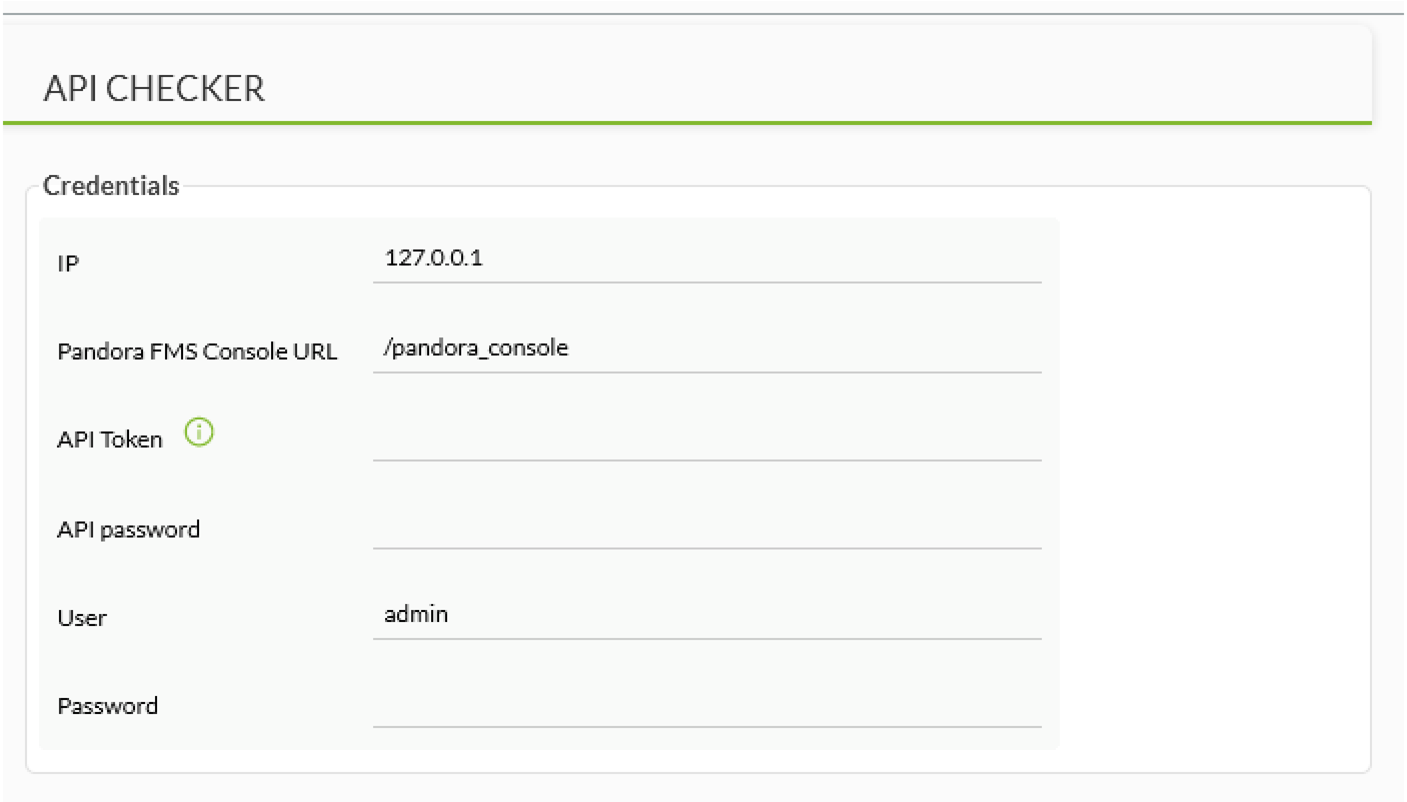
ID

Return Type

Alternate mode

url_encode_separator_|

Custom URL

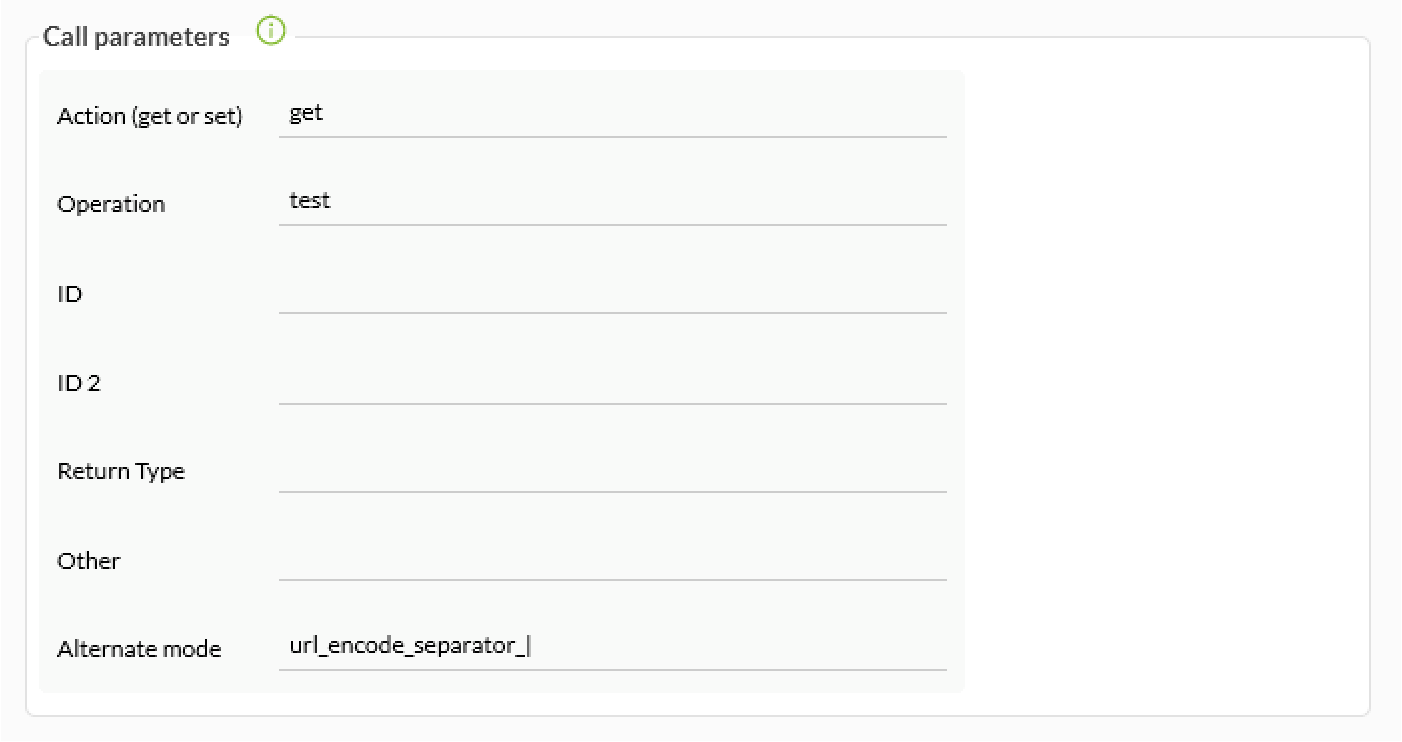


API CHECKER

Credentials

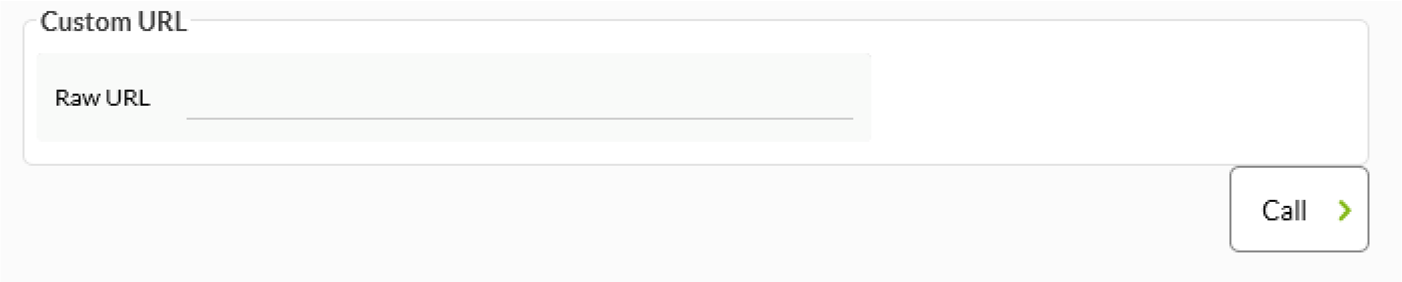
IP	127.0.0.1
Pandora FMS Console URL	/pandora_console
API Token ⓘ	
API password	
User	admin
Password	

Version 768 or later: You can authenticate with API Token by sending in the HTTP headers a bearer token generated by each user and for their own private and particular use. See “Edit my user” for more details.



Call parameters ⓘ

Action (get or set)	get
Operation	test
ID	
ID 2	
Return Type	
Other	
Alternate mode	url_encode_separator_|



Custom URL

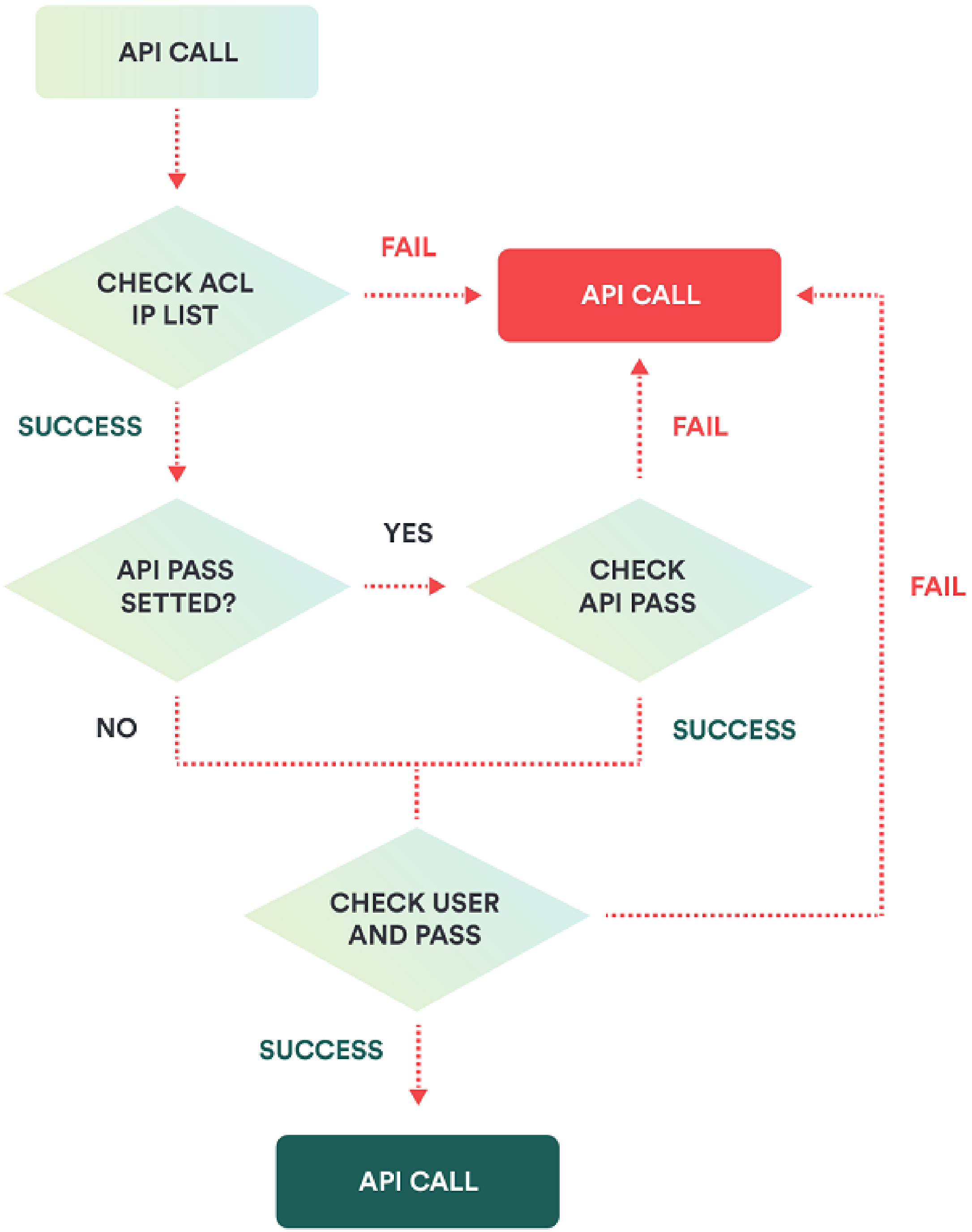
Raw URL

Security Workflow

It is implemented thanks to three elements:

- Filtering by IP address: Only the listed IP addresses will be able to access the API.
- Global password for the API: If defined, it is mandatory to use it in all calls.
- User and password: User of type superadmin and valid in the Web Console. Your permissions will be used for each requested operation. From version NG 768 onwards it is also possible to access by bearer token.

It is explained in this workflow:



API Calls

They are divided into two groups, depending on whether they return or write data in Pandora FMS.

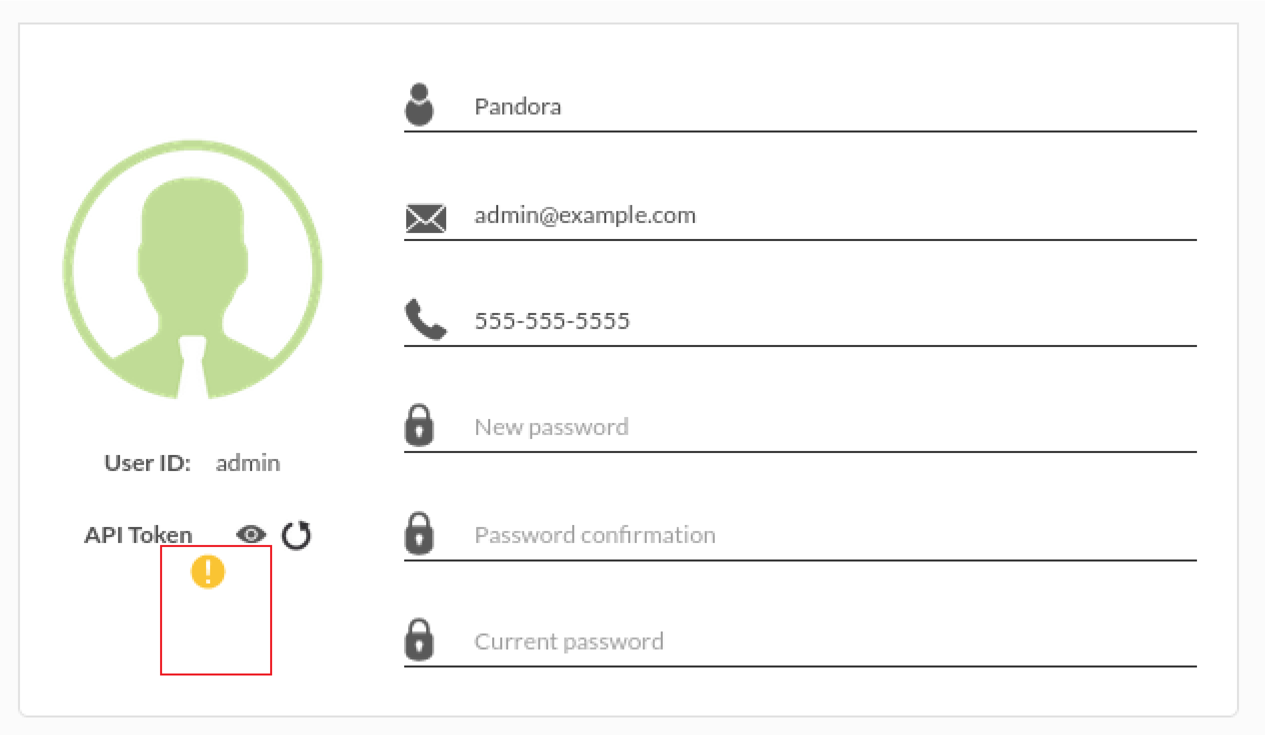
There is an exception: The info retrieving call.

Authentication by bearer token

Version 768 or later: You can authenticate with API Token by sending in the HTTP headers a bearer token generated by each user and for their own private and particular use. See “Edit my user” for more details.

Authentication based on bearer token allows each of the users registered in Pandora FMS to generate their own identifier in one-to-one correspondence. This bearer token must be inserted in the HTTP headers, that is, it is not part of the complete URL of the API. The sending mechanism corresponds to the application that makes the connection with the PFMS API.

Version 768 or later: If a warning icon appears ❗next to the API Token, configure the `php.conf` file.



Pandora

admin@example.com

555-555-5555

New password

Password confirmation

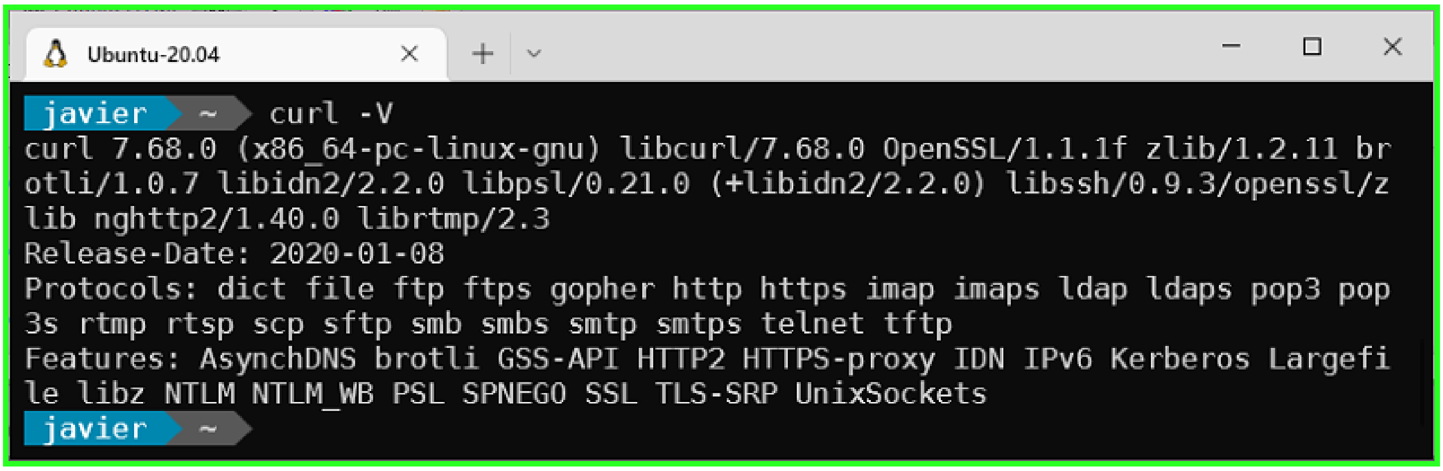
Current password

User ID: admin

API Token ❗

The following examples use the command line program cURL, version 7.68 .

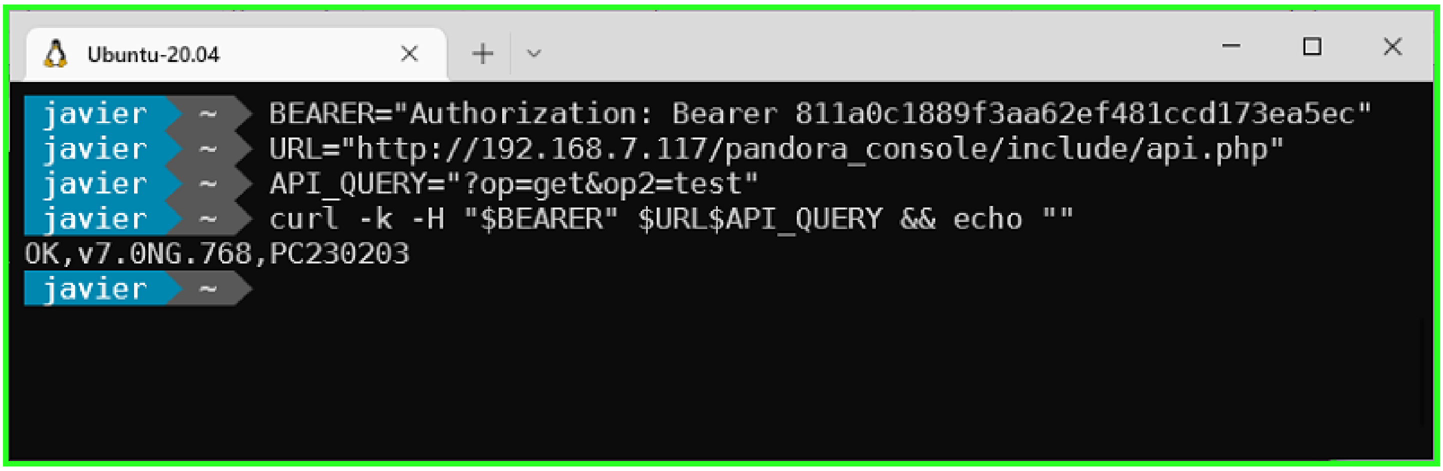
- Check the curl version with `curl -V`.



```
javier  ~  curl -V
curl 7.68.0 (x86_64-pc-linux-gnu) libcurl/7.68.0 OpenSSL/1.1.1f zlib/1.2.11 br
otli/1.0.7 libidn2/2.2.0 libpsl/0.21.0 (+libidn2/2.2.0) libssh/0.9.3/openssl/z
lib nghttp2/1.40.0 librtmp/2.3
Release-Date: 2020-01-08
Protocols: dict file ftp ftps gopher http https imap imaps ldap ldaps pop3 pop
3s rtmp rtsp scp sftp smb smbs smtp smtps telnet tftp
Features: AsynchDNS brotli GSS-API HTTP2 HTTPS-proxy IDN IPv6 Kerberos Largefi
le libz NTLM NTLM_WB PSL SPNEGO SSL TLS-SRP UnixSockets
javier  ~
```

- If you need more help using curl, invoke `curl -h`.
- The `-k` and `-H` parameters are used to accept connections with self-signed certificates for HTTPS and send header information, respectively.
- Consult the bearer token corresponding to the user who will perform the API query.
- Now you can perform an API query, the following example uses get test (replace with your previously queried bearer token):

```
BEARER="Authorization: Bearer 811a0c1889f3aa62ef481ccd173ea5ec"
URL="http://192.168.7.117/pandora_console/include/api.php"
API_QUERY="?op=get&op2=test"
curl -k -H "$BEARER" $URL$API_QUERY && echo ""
```



```
javier  ~  BEARER="Authorization: Bearer 811a0c1889f3aa62ef481ccd173ea5ec"
javier  ~  URL="http://192.168.7.117/pandora_console/include/api.php"
javier  ~  API_QUERY="?op=get&op2=test"
javier  ~  curl -k -H "$BEARER" $URL$API_QUERY && echo ""
OK,v7.0NG.768,PC230203
javier  ~
```

- If you need to debug the connection use the `-v` parameter (not to be confused with `-V` used to display the version).

Info retrieving

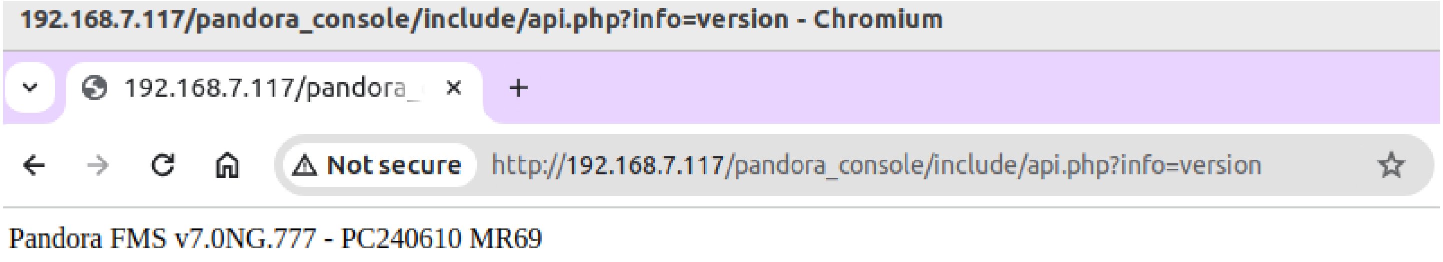
It returns the version of Pandora FMS Console in a similar way as the `get test` call but without checking the API connection.

This call is useful to verify that this path allows a Pandora FMS installation and to retrieve the version before authentication.

The returned info can be retrieved from the login screen, so it is not considered a security vulnerability.

```
http://127.0.0.1/pandora_console/include/api.php?info=version
```

A return sample could be: `Pandora FMS v7.0NG.768 - PC230120 MR60`


192.168.7.117/pandora_console/include/api.php?info=version - Chromium
192.168.7.117/pandora_
Not secure
http://192.168.7.117/pandora_console/include/api.php?info=version

Pandora FMS v7.0NG.777 - PC240610 MR69

GET

It returns the required data.

get test

It checks the connection to the API and returns the version of Pandora FMS Console.

This feature is in Command Center (Metaconsole).

Call syntax: Without parameters

Examples

This example will return OK,[version],[build]

```
http://127.0.0.1/pandora_console/include/api.php?op=get&op2=test&apipass=1234&us
er=admin&pass=pandora
```

A return sample could be: *OK,v7.0NG.751,PC201215*

get license

To consult the Pandora FMS use licence status, depending on the access level of the user's credentials.

Call syntax:

- `op=get` (required).
- `op2=license` (required).
- `return_type=json` (required).

Example (see call syntax):

```
…/include/api.php?op=get&op2=license&return_type=json&apipass=1234&user=admin&pa
ss=pandora
```

For the current version returns:

```
{
    "type": "array",
    "data": {
        "expiry_date": "2099/09/04",
        "limit": 200,
        "limit_mode": "0",
        "nms": "0",
        "dhpm": "1",
        "licensed_to": "PandoraFMS(PandoraFMS <sales@pandorafms.com>)       ",
        "count": "4",
        "count_enabled": "4",
        "count_disabled": "0",
        "license_mode": "Perpetual",
        "expiry_caption": "Support expires"
    }
}
```

get all_agents

It returns a list of agent filters according to the filter in some other parameter.

Call syntax:

- `op=get` (required).
- `op2=all_agents` (required).
- `return_type=csv` or `return_type=json` (required).
- `other=`< serialized_parameters > (optional). Serialized parameters to filter the agent search:

- < filter_so >
- < filter_group >
- < filter_module_states > `unknown`, `warning`, `critical`, `no_modules`.
- < filter_alias >
- < filter_policy >
- < csv_separator >
- Recursion (1 or `0`).

Examples (see call syntax):

This example will return all agents whose id_os is equal to 1 (CSV format):

```
…/pandora_console/include/api.php?op=get&op2=all_agents&return_type=csv&other=1|
|||||0&other_mode=url_encode_separator_%7C&apipass=1234&user=admin&pass=pandora
```

This example will return all agents whose id_group equals 2 (JSON format):

```
…/pandora_console/include/api.php?op=get&op2=all_agents&return_type=json&other=|
2|||||0&other_mode=url_encode_separator_%7C&apipass=1234&user=admin&pass=pandora
```

This example will return all agents whose state equal to critical (CSV format):

```
…/pandora_console/include/api.php?op=get&op2=all_agents&return_type=csv&other=||
critical||||0&other_mode=url_encode_separator_%7C&apipass=1234&user=admin&pass=p
andora
```

This example will return all agents whose their alias contains 'pa' (CSV format):

```
…/pandora_console/include/api.php?op=get&op2=all_agents&return_type=csv&other=||
|pa|||0&other_mode=url_encode_separator_%7C&apipass=1234&user=admin&pass=pandora
```

This example will return all agents whose the policy associated equals 2 (CSV format) and recursion:

```
…/pandora_console/include/api.php?op=get&op2=all_agents&return_type=csv&other=||
||2||1&other_mode=url_encode_separator_%7C&apipass=1234&user=admin&pass=pandora
```

This example will return all agents whose id_os is equal to 1, id_group equals 2, state equal to critical, their alias contains 'pa', and the policy associated equals 2 (CSV format with @ separators):

```
…/pandora_console/include/api.php?op=get&op2=all_agents&return_type=csv&other=1|
2|critical|pa|2|@|0&other_mode=url_encode_separator_%7C&apipass=1234&user=admin&
pass=pandora
```

get license_remaining

> = 7.0NG 752

To check the number of Agents or Modules available according to the Pandora FMS use licence and according to the access level of the user credentials used.

Call syntax:

- op=get (required)
- op2= license_remaining (required)
- return_type=json (required)

Examples

```
http://127.0.0.1/pandora_console/include/api.php?op=get&op2=license_remaining&re
turn_type=json&apipass=1234&user=admin&pass=pandora
```

get module_last_value

Returns the last module value. This module is filtered by the ID which has gone through the id parameter. With the other parameter you may add an error code that your application knows and it is out of range of module values.

Call syntax:

- op=get (required)
- op2=module_last_value (required)
- id=<índex> (required). It should be an agent module index.
- other=<error return> (optional). What you wish to receive if there is an error(usually not located in the database).

- Error return codes are:

- 'error_message'. It returns an error in a text message.
- 'error_value'<separator><code or value>. It returns this code or error value. But it is necessary to enclose it with 'other_mode', like other_mode=url_encode_separator_<separador> to place the divider on another one.

Examples

```
http://127.0.0.1/pandora_console/include/api.php?op=get&op2=module_last_value&id
=63&other=error_value|0&other_mode=url_encode_separator_|&apipass=1234&user=admi
n&pass=pandora
http://127.0.0.1/pandora_console/include/api.php?op=get&op2=module_last_value&id
=62&apipass=1234&user=admin&pass=pandora
```

get agent_module_name_last_value

It returns the last module value. This module is filtered by the agent name which has gone through id parameter and module name which has gone through id2 parameter. With the other parameter, you may add an error code recognized by your application and which is out of range of module values.

Call Syntax:

- op=get (required)
- op2=agent_module_name_last_value (required)
- id=<alphanumeric>(required). It contains the agent name.
- id2=<alphanumeric> (required). It contains the module name.
- other=<error return> (optional). What you wish to receive if there is an error (that usually has not been found in the DB).

- Codes of error return are:

- 'error_message'. It returns error in a text message.
- 'error_value'<separator><code or value>. It returns this code or error value, but it must come with 'other_mode' such as other_mode=url_encode_separator_<separator> to use the divider on another one.

Examples

```
http://127.0.0.1/pandora_console/include/api.php?op=get&op2=agent_module_name_la
st_value&id=miguel-portatil&id2=cpu_user&apipass=1234&user=admin&pass=pandora
```

get agent_module_name_last_value_alias

⇒ 7.0NG

It returns the last module value. This module is filtered by the agent alias which has gone through id parameter and module name which has gone through parameter id2. With the other parameter, you may add an error recognized by your application and which is out range of module values.

Call Syntax:

- op=get (required)
- op2=module_last_value_alias (required)
- id=<alphanumeric>(required). It contains the agent alias.
- id2=<alphanumeric> (required). It contains the module name.
- other=<error return> (optional). What you wish to receive if there is an error (that usually has not been found in the DB).

- Codes of error return are:

- 'error_message'. It returns error in a text message.
- 'error_value'<separator><code or value>. It returns this code or error value, but it must come with 'other_mode' such as other_mode=url_encode_separator_<separator> to use the divider on another one.

Examples

```
http://127.0.0.1/pandora_console/include/api.php?op=get&op2=agent_module_name_la
st_value_alias&id=pandorafms&id2=Memory_Used&apipass=1234&user=admin&pass=pandor
a
```

get module_value_all_agents

Given a module name, returns a list of agents containing that module name. The result returns the agent identifier, agent alias, requested module value and agent name, separated by semicolon (`;`). For example:

```
653;agente
network;0.00000;a8c83b348451040dc91c1327303adf8945aefce6ac59ab9bf856c9f4e0cf6c6f
```

Call syntax:

- `op=get` (required).
- `op2=module_value_all_agents` (required).
- `id=`< module_name > (required).

Example (see call syntax):

A list of all the agents that have the module with `Host Alive` name:

```
.../include/api.php?op=get&op2=module_value_all_agents&id=Host%20Alive&apipass=1
234&user=admin&pass=pandora
```

get agent_modules

Returns the list of modules of an agent requested by its numerical identifier.

Syntax:

- `op=get` (required).
- `op2=agent_modules` (required).
- `return_type=csv` (required) Output format.
- `other=`< serialized values > (required) Serialized values in order to filter by agent:
- < id_agent >

It is absolutely essential to use

`other_mode=url_encode_separator_< separator >`

with this parameter!

Example (see call syntax):

```
.../include/api.php?op=get&op2=agent_modules&return_type=csv&other=14&other_mode
=url_encode_separator_|&apipass=1234&user=admin&pass=pandora
```

get module_id

It returns the id of an agent module.

Call syntax:

- op=get (required)
- op2=module_id (required)
- id=id agent (required)
- other=module name (required)

Examples

```
http://127.0.0.1/pandora_console/include/api.php?op=get&op2=module_id&id=5&other
=Host%20Alive&apipass=1234&user=admin&pass=pandora
```

get module_custom_id

Returns the value of the *Custom_id* field of a specific module of an agent passing as parameter the module id.

Call syntax:

- op=get (required)
- op2=module_custom_id (required)
- id=id_agent_module (required)

Examples

```
http://127.0.0.1/pandora_console/include/api.php?op=get&op2=module_custom_id&id=
5&apipass=1234&user=admin&pass=pandora
```

set module_custom_id

Changes or clears the value of the *Custom ID* field passing as parameters the module id and the value of the custom id.

Call syntax:

- op=set (required)
- op2=module_custom_id (required)
- id=id_agent_module (required)
- id2=value, or vacuum to clean (required)

Examples

```
http://127.0.0.1/pandora_console/include/api.php?op=set&op2=module_custom_id&id=
5&id2=1521&apipass=1234&user=admin&pass=pandora
```

get locate_agent

It gets the id server where the agent is located, and prints all the results like a csv.

Call syntax:

- op=get (required)
- op2=locate_agent (required)
- return_type=<csv> (required). Output format.
- id=id_agent (required)
- other_mode= other_mode=url_encode_separator_<separator> (required)

Examples

```
http://127.0.0.1/pandora_console/include/api.php?op=get&op2=locate_agent&return_type=csv&apipass=1234&user=admin&pass=pandora&id=Pepito&other_mode=url_encode_separator_%7C
```

get policies

It returns the list of agent policies, whose id which has gone through the other parameter.

Call syntax:

- op=get (required)
- op2=policies (required)
- return_type=<csv> (required)
- other=<serialized values> (optional). Serialized values for filtering policies by policy agent:

- <id_agent>

Examples

```
http://127.0.0.1/pandora_console/include/api.php?op=get&op2=policies&return_type=csv&other=&other_mode=url_encode_separator_|&apipass=1234&user=admin&pass=pandora
```

get tree_agents

It returns a complete list structured by the groups in the first level, agents in the second level and modules in the third level. This list is filtered by the other parameter.

Call Syntax:

- op=get (required)
- op2=tree_agents (required)
- return_type=<return kind> (required). That could be 'csv' or 'csv_head'.
- other=<string or serialized parameters> (optional). In this case, it could be the divider or a parameter list ordered and separated by the divider character. Both cases are described here:

- <separator> The divider yes of the `csv`.
- <separator csv>|<character that replaces the CR|<fields 1>,<fields 2>,<fields N>. It will place the following parameters in order (the divider character | could be specified in `other_mode`):

- <separator csv>. Field divider in the CSV.
- <character that replaces the CR>. Character that will be replaced if it finds in any returned character the character RC, in order to avoid the ambiguity with the standard use of the RC character to specify registers/files in the CSV. If you make an string go through other, the replacing character is the blank space.

- <fields 1>,<fields2>,<fields N> :the fields to show in the CSV are:
    - type_row
    - group_id
    - group_name
    - group_parent
    - disabled
    - custom_id
    - group_description
    - group_contact
    - group_other
    - agent_id
    - alias
    - agent_direction
    - agent_commentary
    - agent_id_group
    - agent_last_contact
    - agent_mode
    - agent_interval
    - agent_id_os
    - agent_os_version
    - agent_version
    - agent_last_remote_contact
    - agent_disabled
    - agent_id_parent
    - agent_custom_id
    - agent_server_name
    - agent_cascade_protection
    - agent_name
    - module_id_agent_modulo
    - module_id_agent
    - module_id_module_type
    - module_description
    - module_name
    - module_max
    - module_min
    - module_interval
    - module_tcp_port
    - module_tcp_send
    - module_tcp_rcv
    - module_snmp_community
    - module_snmp_oid
    - module_ip_target
    - module_id_module_group

- module_flag
- module_id_module
- module_disabled
- module_id_export
- module_plugin_user
- module_plugin_pass
- module_plugin_parameter
- module_id_plugin
- module_post_process
- module_prediction_module
- module_max_timeout
- module_custom_id
- module_history_data
- module_min_warning
- module_max_warning
- module_min_critical
- module_max_critical
- module_min_ff_event
- module_delete_pending
- module_id_agent_state
- module_data
- module_timestamp
- module_state
- module_last_try
- module_utimestamp
- module_current_interval
- module_running_by
- module_last_execution_try
- module_status_changes
- module_last_status
- module_plugin_macros
- module_macros
- module_critical_inverse (only in version 6.0SP1 or later)
- module_warning_inverse (only in version 6.0SP1 or later)
- alert_id_agent_module
- alert_id_alert_template
- alert_internal_counter
- alert_last_fired
- alert_last_reference
- alert_times_fired
- alert_disabled
- alert_force_execution
- alert_id_alert_action
- alert_type
- alert_value
- alert_matches_value
- alert_max_value
- alert_min_value
- alert_time_threshold
- alert_max_alerts
- alert_min_alerts
- alert_time_from

- alert_time_to
- alert_monday
- alert_tuesday
- alert_wednesday
- alert_thursday
- alert_friday
- alert_saturday
- alert_sunday
- alert_recovery_notify
- alert_field2_recovery
- alert_field3_recovery
- alert_id_alert_template_module
- alert_fires_min
- alert_fires_max
- alert_id_alert_command
- alert_command
- alert_internal
- alert_template_modules_id
- alert_templates_id
- alert_template_module_actions_id
- alert_actions_id
- alert_commands_id
- alert_templates_name
- alert_actions_name
- alert_commands_name
- alert_templates_description
- alert_commands_description
- alert_template_modules_priority
- alert_templates_priority
- alert_templates_field1
- alert_actions_field1
- alert_templates_field2
- alert_actions_field2
- alert_templates_field3
- alert_actions_field3
- alert_templates_id_group
- alert_actions_id_group'

Examples

```
http://127.0.0.1/pandora_console/include/api.php?op=get&op2=tree_agents&return_type=csv&other=;&apipass=1234&user=admin&pass=pandora
http://127.0.0.1/pandora_console/include/api.php?op=get&op2=tree_agents&return_type=csv&other=;|%20|type_row,group_id,agent_name&other_mode=url_encode_separator_|&apipass=1234&user=admin&pass=pandora
```

get module_data

Returns a list of values of a module, this module is filtered by the id of module pass as id in the url.

And the list of values is from the now to the period limit passed as second parameter into the other parameter, the first is the CSV separator.

Call syntax:

- op=get (required)
- op2=module_data (required)
- id=<id_modulo> (required)
- other_mode=url_encode_separator_<separator> (required)
- other=<serialized parameters> (optional)

- The CSV divider character (point by default)
- The period (in seconds)
- Start time (<year><month><day>T<hour>:<minute>)
- End time (<year><month><day>T<hour>:<minute>)

Examples

```
http://127.0.0.1/pandora_console/include/api.php?op=get&op2=module_data&id=17&other=;|604800|20121201T13:40|20171215T13:40&other_mode=url_encode_separator_|&apipass=1234&user=admin&pass=pandora
```

get graph_module_data

Returns the chart of a module as an image file, this chart is generated with the same method of static graphs of Pandora FMS. It is necesary to include the width, height, period, label and start date of chart (knowing in Pandora FMS the date indicated in graphs is always the data end) into the other parameter.

Call syntax:

- op=set (required)
- op2=module_data (required)
- id=<id_modulo> (required)
- other=<serialized parameters> (required). They are the following in this order:

- <period>
- <width>
- <height>
- <label>
- <start_date>

Examples

```
http://127.0.0.1/pandora_console/include/api.php?op=get&op2=graph_module_data&id=17&other=604800|555|245|pepito|2009-12-07&other_mode=url_encode_separator_|&apipass=1234&user=admin&pass=pandora
```

get events

It returns a list of events filtered by the `other` parameter.

Call syntax:

- `op=get` (required).
- `op2=events` (required).
- `id=`< id_node > (optional in Command Center -Metaconsole-, unnecessary in node).
- `return_type=csv` or `return_type=json` (required).
- `other_mode=url_encode_separator_|` (required).
- `other=`< serialized parameters > (optional), are the following in this order:
    1. < separator > (required).
    2. < criticality > any of the following values from `0` to number `6`:
        1. `0:Maintenance`
        2. `1:Informational`
        3. `2:Normal`
        4. `3:Warning`
        5. `4:Critical`
        6. `5:Minor`
        7. `6:Major`
    3. < agent alias >
    4. < module name >
    5. < filter by alert > use `1` to filter events generated by alerts.
    6. < event owner user >
    7. < minimum date and time (since) > in Unix® time format, without milliseconds (EPOCH).
    8. < maximum date and time (until) > in Unix® time format, without milliseconds (EPOCH).
    9. < state > `0` to filter non-validated events, `1` for validated events.
    10. < text > keyword to search for in the events.
    11. < page size > sets the number of records to return for pagination (offset). If the offset number is omitted, it will only return the first offset (the first offset is the number zero `0`).
    12. < page number > according to the page size of the previous point, returns earlier or later record blocks (offset).
    13. < style > only accepts two key values:
        1. `total`: it simply returns the total number of events, for example in JSON format delivers: `{“type”:“json”,“data”:{“count”:990}}`.
        2. `more_criticity`: returns the number of most critical events, for example if requested in JSON format: `{“type”:“json”,“data”:[{“criticity”:“4”,“user_can_manage”:“1”,“user_can_write”:“1”,“server_id”:5,“server_name”:“stod”}]}`.
    14. < group_id > returns non-validated events belonging to a group of agents, use numeric values only.
    15. < tag > any of the tags registered in PFMS (see get tags). JSON format required, for example: `[“2”,“5”]` (see call syntax).
    16. < event type > any of the following values:
        - `going_unknown`
        - `unknown`
        - `recon_host_detected`
        - `system`
        - `error`
        - `new_agent`

  - going_up_warning
  - going_up_critical
  - going_down_warning
  - going_down_normal
  - going_down_critical
  - going_up_normal
  - configuration_change
  - ncm
  - not_normal

Example (see call syntax):

To obtain the `ncm` (Network configuration manager) events by querying a Command Center (Metaconsole) whose node is 5:

```
…include/api.php?op=get&op2=events&user=admin&pass=pandora&return_type=json&apip
ass=1234&other_mode=url_encode_separator_|&other=|||||||||||||||ncm&id=5
```

get all_alert_templates

It returns the list of alert templates defined in Pandora FMS.

Call syntax:

- op=get (required)
- op2=all_alert_templates (required)
- other=cvs_separator (optional)

Examples

```
http://127.0.0.1/pandora_console/include/api.php?op=get&op2=all_alert_templates&
return_type=csv&other=;&apipass=1234&user=admin&pass=pandora
```

get module_groups

It returns the list of module groups.

Call syntax:

- op=get (required)
- op2=module_groups (required)
- other=cvs_separator (optional)

Examples

```
http://127.0.0.1/pandora_console/include/api.php?op=get&op2=module_groups&return
_type=csv&other=;&apipass=1234&user=admin&pass=pandora
```

get plugins

It returns the list of Pandora FMS server plugins.

Call syntax:

- op=get (required)
- op2=plugins (required)
- other=cvs_separator (optional)

Examples

```
http://127.0.0.1/pandora_console/include/api.php?op=get&op2=plugins&return_type=csv&other=;&apipass=1234&user=admin&pass=pandora
```

get tags

It returns the list of tags defined in Pandora FMS.

Call syntax:

- op=get (required)
- op2=tags (required)
- return_type=csv or return_type=json (required)
- other=< field separator > if you request in CSV format you must specify separator, otherwise it will return the fields together.

Example (see call syntax):

This example will return all tags in the system in CSV format with semicolon separator.

```
…/include/api.php?op=get&op2=tags&return_type=csv&other=;&apipass=1234&user=admin&pass=pandora
```

get module_from_conf

It returns the configuration of a local module.

Call syntax:

- op=get (required)
- op2=update_module_in_conf (required)
- id=<agent id> (required)
- id2=<module name> (required)

Examples

```
http://127.0.0.1/pandora_console/include/api.php?op=get&op2=module_from_conf&api
pass=1234&user=admin&pass=pandora&id=9043&id2=example_name
```

It returns an empty string if no modules are found.

get total_modules

Total modules by group.

Call syntax:

- `op=get` (required).
- `op2=total_modules` (required).
- `id=< id group >` (required up to version 768; *optional from version 769 onwards*).

Examples (see call syntax)

To obtain the total number of modules in the Networking *module group:*

```
.../include/api.php?op=get&op2=total_modules&id=2&apipass=1234&user=admin&pass=p
andora
```

The following groups of modules come by default when installing Pandora FMS:

1. General.
2. Networking.
3. Application.
4. System.
5. Miscellaneous.
6. Performance.
7. Database.
8. Enviromental.
9. Users.

If you set `0` as group id, or simply omit it, it will return the total number of modules:

```
.../include/api.php?op=get&op2=total_modules&apipass=1234&user=admin&pass=pandor
a
```

get total_agents

Total agents by group.

Call syntax:

- op=get (required)
- op2=total_agents (required)
- id=<id group> (required)

Examples

```
http://localhost/pandora_console/include/api.php?op=get&op2=total_agents&id=2&apipass=1234&user=admin&pass=pandora
```

get agent_name

Agent name for a given id

Call syntax:

- op=get (required)
- op2=agent_name (required)
- id=<agent id> (required)

Examples

```
http://localhost/pandora_console/include/api.php?op=get&op2=agent_name&id=1&apipass=1234&user=admin&pass=pandora
```

get agent_alias

> = 7.0NG

Agent alias for a given id.

Call syntax:

- op=get (required)
- op2=agent_alias (required)
- id=<agent id> (required)
- id2=<node id> (required in the meta console, unnecessary in the node)

Examples

- Node:

```
http://localhost/pandora_console/include/api.php?op=get&op2=agent_alias&id=1&apipass=1234&user=admin&pass=pandora
```

- Command Center (Metaconsole):

```
http://localhost/pandora_console/enterprise/meta/include/api.php?op=get&op2=agent_alias&id=1&id2=1&apipass=1234&user=admin&pass=pandora
```

get module_name

Module name for a given id.

Call syntax:

- op=get (required)
- op2=module_name (required)
- id=<module id> (required)

Examples

```
http://localhost/pandora_console/include/api.php?op=get&op2=module_name&id=1&apipass=1234&user=admin&pass=pandora
```

get alert_action_by_group

Total alert execution with an action by group.

Call syntax:

- op=get (required)
- op2=alert_action_by_group (required)
- id=<group id> (required)
- id2=<action id> (required)

Examples

```
http://localhost/pandora_console/include/api.php?op=get&op2=alert_action_by_group&id=0&id2=3&apipass=1234&user=admin&pass=pandora
```

get alert_actions

List all alert actions.

Call syntax:

- op=`get` (required).
- op2=`alert_actions` (required).
- `return_type`=< list_format > choice `csv` or `json` (required).
- `other`=< serialized parameter > (optionals):
  - < action_name >
  - < separator >

Examples:

To get the list of all alert actions in JSON *format* (see call syntax)*:*

```
.../include/api.php?op=get&op2=alert_actions&return_type=json&other_mode=url_encode_separator_|&apipass=1234&user=admin&pass=pandora
```

To get the list of all alert actions in CSV *format* (see call syntax)*:*

```
.../include/api.php?op=get&op2=alert_actions&return_type=csv&other_mode=url_encode_separator_|&apipass=1234&user=admin&pass=pandora
```

get alert_actions_meta

This feature is in Command Center (Metaconsole).

List of node alert actions from Command Center (Metaconsole).

Call syntax:

- op=get (required)
- op2=alert_actions_meta (required)
- return_type=<return type> (required). It can be 'csv' o 'json'.
- other=<serialized parameters> (optional):

- <server_name> (optional)
- <action_name> (optional)
- <separator> (optional)

Examples

```
http://localhost/pandora_console/include/api.php?op=get&op2=alert_actions_meta&apipass=1234&user=admin&pass=pandora&other=nodo|Create|&other_mode=url_encode_separator_|&return_type=json
```

get all_alert_commands

It returns all the alert commands.

Call syntax:

- op=get (required)
- op2=all_alert_commands (required)
- other=cvs_separator(optional)

Examples

```
http://localhost/pandora_console/include/api.php?op=get&op2=all_alert_commands&return_type=csv&other=$&apipass=1234&user=admin&pass=pandora
```

get event_info

It returns all event data by typing in the event id.

This feature is in Command Center (Metaconsole).

Call syntax:

- op=get (required)
- op2=event_info (required)
- id=<id_event> (required)

Examples

```
http://127.0.0.1/pandora_console/include/api.php?op=get&op2=event_info&id=80&api
pass=1234&user=admin&pass=pandora
```

get tactical_view

It returns the following value list (this values can be seen in the tactical page in Pandora FMS Console)

- monitor_checks
- monitor_not_init
- monitor_unknown
- monitor_ok
- monitor_bad
- monitor_warning
- monitor_critical
- monitor_not_normal
- monitor_alerts
- monitor_alerts_fired
- monitor_alerts_fire_count
- total_agents
- total_alerts
- total_checks
- alerts
- agents_unknown
- monitor_health
- alert_level
- module_sanity
- server_sanity
- total_not_init
- monitor_non_init
- agent_ok

- agent_warning
- agent_critical
- agent_unknown
- agent_not_init
- global_health

Call syntax:

- op=get (required)
- op2=tactical_view (required)

Example

```
http://localhost/pandora_console/include/api.php?op=get&op2=tactical_view&apipas
s=1234&user=admin&pass=pandora
```

get pandora_servers

> = 5.0

It returns the list of Pandora FMS servers.

Call syntax:

- op=get (required)
- op2=pandora_servers (required)
- other=cvs_separator (optional)
- return_type=csv (required)

Example

```
http://localhost/pandora_console/include/api.php?op=get&op2=pandora_servers&retu
rn_type=csv&apipass=1234&user=admin&pass=pandora
```

It returns the fields in this order:

- name
- status (1 - up, 0 - down)
- type (human readable string)
- master (1 - master, 0 - not master)
- running modules
- total modules
- max delay (sec)
- delayed modules
- threads
- queued_modules
- timestamp of update (human readable string)

get custom_field_id

It translates the name of the custom field into the ID it has in the data base.

Call syntax:

- op=get (required).
- op2=custom_field_id (required).
- other=< parameter > (required) In this case, custom field name.

Example (see call syntax):

```
…/include/api.php?op=get&op2=custom_field_id&other=Serial%20Number&other_mode=url_encode_separator_|&apipass=1234&user=admin&pass=pandora
```

get gis_agent

> = 5.0

It returns the last GIS agent data.

Call syntax:

- op=set (required)
- op2=gis_agent (required)
- id=<index> (required). It must be an agent index.

Example

```
http://localhost/pandora_console/include/api.php?apipass=1234&user=admin&pass=pandora&op=get&op2=gis_agent&id=1
```

get special_days

> = 5.1

It returns special day's list.

Call syntax:

- op=get (required)
- op2=special_days (required)
- other=<csv separator> (optional). CSV separator

Example

```
http://127.0.0.1/pandora_console/include/api.php?apipass=1234&user=admin&pass=pa
```

```
ndora&op=get&op2=special_days
```

get module_properties

> = 5.1SP2

It returns a list of module properties. The list could be filtered by parameter.

Call Syntax:

- op=get (required)
- op2=module_properties (required)
- id=module-id (required)
- return_type=<return kind> (required). That could be `csv` or `csv_head`.
- other=<string or serialized parameters> (optional). In this case, it could be the divider or a parameter list ordered and separated by the divider character. Both cases are exposed:

- <separator>. The divider `yes` of the `csv`.
- <separator csv>|<. Character that replaces the CR|<fields 1>,<fields 2>,<fields N>. It will make up the following parameters in order (the divider character | could be specified in `other_mode`):

- <separator csv>. Field divider in the CSV.
- <character that replaces the CR>. Character that will be replaced if it finds in any returned character the character RC, in order to avoid the ambiguity with the standard use of the RC character to specify registers/files in the CSV. If you make a string go through other, the replacing character is the blank space.
- <fields 1>,<fields2>,<fields N>. The fields to output in CSV are:
    - module_id_agent_module
    - module_id_agent
    - module_id_module_type
    - module_description
    - module_name
    - module_max
    - module_min
    - module_interval
    - module_tcp_port
    - module_tcp_send
    - module_tcp_rcv
    - module_snmp_community
    - module_snmp_oid
    - module_ip_target
    - module_id_module_group
    - module_flag
    - module_id_module
    - module_disabled
    - module_id_export
    - module_plugin_user
    - module_plugin_pass
    - module_plugin_parameter
    - module_id_plugin
    - module_post_process
    - module_prediction_module

  - module_max_timeout
  - module_custom_id
  - module_history_data
  - module_min_warning
  - module_max_warning
  - module_min_critical
  - module_max_critical
  - module_min_ff_event
  - module_delete_pending
  - module_id_agent_state
  - module_data
  - module_timestamp
  - module_state
  - module_last_try
  - module_utimestamp
  - module_current_interval
  - module_running_by
  - module_last_execution_try
  - module_status_changes
  - module_last_status
  - module_plugin_macros
  - module_macros
  - module_critical_inverse (only in version 6.0SP1 or later)
  - module_warning_inverse (only in version 6.0SP1 or later)

Examples

```
http://127.0.0.1/pandora_console/include/api.php?op=get&op2=module_properties&id=6233&return_type=csv&other=;&apipass=1234&user=admin&pass=pandora
http://127.0.0.1/pandora_console/include/api.php?op=get&op2=module_properties&id=1103&return_type=csv&other=;|%20|module_id_agent_module,module_id_module_type,module_name,module_last_try,module_state&other_mode=url_encode_separator_|&apipass=1234&user=admin&pass=pandora
```

get module_properties_by_name

> = 5.1SP2

It returns a list of module properties. The list could be filtered by parameter.

Call Syntax:

- op=get (required)
- op2=module_properties_by_name (required)
- id=agent_name (required)
- id2=agentmodule_name (required)
- return_type=<return kind> (required). That could be 'csv' or 'csv_head'.
- other=<string or serialized parameters> (optional). In this case, it could be the divider or a parameter list in order and separated by the divider character. (the same as 'get module_properties'. For details, see 'get module_properties')

Examples

```
http://127.0.0.1/pandora_console/include/api.php?op=get&op2=module_properties_by_name&id=my_agent&id2=my_module&return_type=csv&other=;&apipass=1234&user=admin&pass=pandora
http://127.0.0.1/pandora_console/include/api.php?op=get&op2=module_properties_by_name&id=my_agent&id2=my_module&return_type=csv&other=;|%20|module_id_agent_module,module_id_module_type,module_type,module_name,module_last_try,module_state&other_mode=url_encode_separator_|&apipass=1234&user=admin&pass=pandora
```

get module_properties_by_alias

> = 7.0NG

It returns a list of module properties. The list could be filtered by parameter.

Call Syntax:

- op=get (required)
- op2=module_properties_by_alias (required)
- id=agent_alias (required)
- id2=agentmodule_name (required)
- return_type=<return kind> (required). That could be 'csv' or 'csv_head'.
- other=<string or serialized parameters> (optional). In this case, it could be the divider or a parameter list ordered and separated by the divider character (the same as 'get module_properties'. For details, see 'get module_properties').

Examples

```
http://127.0.0.1/pandora_console/include/api.php?op=get&op2=module_properties_by_alias&id=pandorafms&id2=Memory_Used&return_type=csv&other=;&apipass=1234&user=admin&pass=pandora
```

get module_graph

It returns a Module graph encoded with base64..

Call Syntax:

- op=get (required)
- op2=module_graph (required)
- id=<module id> (required)
- other=<period in seconds> (optional) 3600 by default (1 hour). Period of time used in the data recovery.

From version NG 752 onwards, more parameters are available, please pay attention to the mandatory fields and their separators | or its Escape character `%7C` .

- op=get (required)
- op2=module_graph (required)
- id=<module id> (required)
- other=
  - Time range in seconds (default 3600) in which data will be collected from the date of the request backwards (required).
  - Separator.
  - 0 for base64 graphics, 1 for image (required).
  - Separator.
  - 0 without thresholds, 1 with thresholds - if the Module has active thresholds - (required).
  - Separator.
  - Height of the graph, in pixels (required). Additionally it needs 49 pixels to display the map keys.

Examples Prior to version NG 752:

```
http://localhost/pandora_console/include/api.php?op=get&op2=module_graph&id=5&other=3600%7C1&other_mode=url_encode_separator_%7C&apipass=1234&api=1&user=admin&pass=pandora
```

Version NG 752 or higher:

```
http://localhost/pandora_console/include/api.php?op=get&op2=module_graph&id=2&other=3600%7C1%7C1%7C500%7C&other_mode=url_encode_separator_%7C|&apipass=1234&api=1&user=admin&pass=pandora
```

get all_planned_downtimes

> = 5.1

It returns all matches of planned downtime.

op=get op2=all_planned_downtimes return_type=csv other=<name> <group> <type_downtime> <type_execution> <type_periodicity> (for filtering)

Example

```
http://127.0.0.1/pandora_console/include/api.php?op=get&op2=all_planned_downtimes&other=test|0|quiet|periodically|weekly&other_mode=url_encode_separator_|&return_type=json&apipass=1234&user=admin&pass=pandora
http://127.0.0.1/pandora_console/include/api.php?op=get&op2=all_planned_downtimes&other=test|0|quiet|periodically|weekly&other_mode=url_encode_separator_|&return_type=csv&apipass=1234&user=admin&pass=pandora
```

get planned_downtimes_items

> = 5.1

It returns all matches of planned downtime items.

op=get op2=planned_downtimes_items return_type=csv other=<name> <group> <type_downtime> <type_execution> <type_periodicity> (for filter)

Example

```
http://127.0.0.1/pandora_console/include/api.php?op=get&op2=planned_downtimes_it
ems&other=test|0|quiet|periodically|weekly&other_mode=url_encode_separator_|&ret
urn_type=json&apipass=1234&user=admin&pass=pandora
http://127.0.0.1/pandora_console/include/api.php?op=get&op2=planned_downtimes_it
ems&other=test|0|quiet|periodically|weekly&other_mode=url_encode_separator_|&ret
urn_type=csv&apipass=1234&user=admin&pass=pandora
```

get id_group_agent_by_name

> = 5.1

It returns the group id of an agent.

Call Syntax:

- op=get
- op2=id_group_agent_by_name
- other=<string or serialized parameters>. Agent name and filtering parameters (optional).
    - <agent_name>

Example

```
http://127.0.0.1/pandora_console/include/api.php?apipass=1234&user=admin&pass=pa
ndora&op=get&op2=id_group_agent_by_name&other_mode=url_encode_separator_|&other=
192.168.50.40
```

get id_group_agent_by_alias

> = 7.0NG

It returns the group id of an agent.

Call Syntax:

- op=get
- op2=id_group_agent_by_alias

- other=<string or serialized parameters>. Agent alias and filtering parameters (optional)
  - <agent_alias>

Example

```
http://127.0.0.1/pandora_console/include/api.php?apipass=1234&user=admin&pass=pandora&op=get&op2=id_group_agent_by_alias&other_mode=url_encode_separator_%7C&other=pandorafms
```

get group_agent_by_name

> = 5.1

It returns the group name for a agent.

Call Syntax:

- op=get
- op2=group_agent_by_name
- other=<string or serialized parameters> Agent name and filtering parameters (optional).
  - <agent_name>

Example

```
http://127.0.0.1/pandora_console/include/api.php?apipass=1234&user=admin&pass=pandora&op=get&op2=group_agent_by_name&other_mode=url_encode_separator_|&other=192.168.50.40
```

get group_agent_by_alias

> = 7.0NG

It returns the group name for an agent.

Call Syntax:

- op=get
- op2=group_agent_by_alias
- other=<string or serialized parameters>. Agent alias and filtering parameters (optional).
  - <agent_alias>

Example

```
http://127.0.0.1/pandora_console/include/api.php?apipass=1234&user=admin&pass=pandora&op=get&op2=group_agent_by_alias&other_mode=url_encode_separator_%7C&other=pandorafms
```

get group_id_by_name

> = 7.0NG

It returns the group id from the name.

Call syntax:

- op=get
- op2=group_id_by_name
- other=<unique parameter> (required):
    - <group_name>

Example

```
http://127.0.0.1/pandora_console/include/api.php?apipass=1234&user=admin&pass=pa
ndora&op=get&op2=group_id_by_name&other=Servers
```

get timezone

> = 7.0NG

It returns the timezone with which pandora is configured.

Call syntax:

- op=get
- op2= timezone

Example

```
http://127.0.0.1/pandora_console/include/api.php?apipass=1234&user=admin&pass=pa
ndora&op=get&op2=timezone
```

get cluster_status

> = 7.0

It returns the cluster status by id

- op=get
- op2=cluster_status
- id=cluster id

```
http://127.0.0.1/pandora_console/include/api.php?op=get&op2=cluster_status&id=1&
apipass=1234&user=admin&pass=pandora
```

get cluster_id_by_name

> = 7.0

It returns the id of the cluster by name

- op=get
- op2=cluster_id_by_name
- id=cluster name

```
http://127.0.0.1/pandora_console/include/api.php?op=get&op2=cluster_id_by_name&id=paco&apipass=1234&user=admin&pass=pandora
```

get agents_id_name_by_cluster_id

> = 7.0

It returns the id ⇒ cluster agents names by cluster id

- op=get
- op2=agents_id_name_by_cluster_id
- id=cluster id

```
http://127.0.0.1/pandora_console/include/api.php?op=get&op2=agents_id_name_by_cluster_id&id=1&apipass=1234&user=admin&pass=pandora
```

get agents_id_name_by_cluster_name

> = 7.0 It returns the id ⇒ cluster agents names by cluster name

- op=get
- op2=agents_id_name_by_cluster_name
- id=cluster id

```
http://127.0.0.1/pandora_console/include/api.php?op=get&op2=agents_id_name_by_cluster_name&id=paco&apipass=1234&user=admin&pass=pandora
```

get modules_id_name_by_cluster_id

> = 7.0

It returns the id ⇒ all the module names of the cluster agents by cluster id

- op=get
- op2=modules_id_name_by_cluster_id
- id=cluster id

```
http://127.0.0.1/pandora_console/include/api.php?op=get&op2=modules_id_name_by_cluster_id&id=1&apipass=1234&user=admin&pass=pandora
```

get modules_id_name_by_cluster_name

> = 7.0

It returns the id ⇒ all the module names of the cluster agents by cluster name

- op=get
- op2=modules_id_name_by_cluster_name
- id=cluster name

```
http://127.0.0.1/pandora_console/include/api.php?op=get&op2=modules_id_name_by_cluster_name&id=paco&apipass=1234&user=admin&pass=pandora
```

get cluster_items

> = 7.0

It returns the items of a json object with the format: item id ⇒ (name,id,module_agent,type,pcrit,pwarn), by cluster id

pcrit: % for the critic state if its AA or 1 o 0 if its AP

pwarn: % for the warning state if its AA or null if its AP

- op=get
- op2=cluster_items
- id=cluster id

```
http://127.0.0.1/pandora_console/include/api.php?op=get&op2=cluster_items&id=1&apipass=1234&user=admin&pass=pandora
```

get policy

> = 7.0NG. 725

It returns all data from a policy. If no policy id is specified, it will return all data from all policies.

Call syntax:

- op=get
- op2=policy
- id=id_policy (optional)
- return_type=(json, csv, string)

Example

```
http://localhost/pandora_console/include/api.php?op=get&op2=policy&apipass=1234&
```

```
user=admin&pass=pandora&return_type=json.
```

get collections_policy

> = 7.0NG. 725

It returns all policy colections. If no policy id is specified, it returns all policy colections.

Call syntax:

- op=get
- op2=collections_policy
- id=id_policy (optional)
- return_type=(json, csv, string)

Example

```
http://localhost/pandora_console/include/api.php?op=get&op2=collections_policy&a
pipass=1234&user=admin&pass=pandora&return_type=json.
```

get plugins_policy

> = 7.0NG. 725

It returns all policy plugins. If no policy id is specified, it returns all policy plugins.

Call syntax:

- op=get
- op2=plugins_policy
- id=id_policy (optional)
- return_type=(json, csv, string).

Example

```
http://localhost/pandora_console/include/api.php?op=get&op2=plugins_policy&apipa
ss=1234&user=admin&pass=pandora&return_type=json.
```

get inventory_policy

> = 7.0NG. 725

It returns all inventory modules of a policy. If no policy id is specified, it returns all inventory modules of all policies.

Call syntax:

- op=get
- op2=inventory_policy
- id=id_policy (optional)
- return_type=(json, csv, string)

Example

```
http://localhost/pandora_console/include/api.php?op=get&op2=inventory_policy&api
pass=1234&user=admin&pass=pandora&return_type=json&id=2
```

get unlinking_policy

> = 7.0NG. 725

It returns all unlinked modules of a policy. If no policy id is specified, it returns all unlinked modules of all policies.

Call syntax:

- op=get
- op2=unlinking_policy
- id=id_policy(optional)
- return_type=(json, csv, string)

Example

```
http://localhost/pandora_console/include/api.php?op=get&op2=unlinking_policy&api
pass=1234&user=admin&pass=pandora&return_type=json.
```

get alerts_policy

> = 7.0NG. 725

It returns the alerts of a policy, taking into account:

- If a policy id is specified, it returns the alerts of that policy. If not, it returns all alerts of all policies.
- If the type of alert is specified: (three options).
  - 0: it shows all by default.
  - 1: Shows only the normal alerts.
  - 2: Shows only the external alerts.

Call syntax:

- op=get
- op2=alerts_policy
- id=id_policy(optional)
- id2=0:all(by default), 1:normal, 2:external.
- return_type=(json, csv, string)

Example

```
http://localhost/pandora_console/include/api.php?op=get&op2=alerts_policy&apipass=1234&user=admin&pass=pandora&return_type=json&id2=2.
```

get alerts_actions_policy

> = 7.0NG. 725

It returns all actions of the alerts of a policy, taking into account:

- If a policy id is specified, it returns the actions of the alerts of that policy. If not, it returns all the actions of all the alerts of all policies.
- If an alert id is specified, it returns the action of that alert.

Call syntax:

- op=get
- op2=alerts_actions_policy
- id=id_policy(optional)
- id2=id_policy_alert(optional)
- return_type=(json, csv, string)

Example

```
http://localhost/pandora_console/include/api.php?op=get&op2=alerts_actions_policy&apipass=1234&user=admin&pass=pandora&return_type=json&id=1&id2=3
```

get agents_policy

> = 7.0NG. 725

It returns all policy agents. If no policy id is specified, it returns all agents of all policies.

Call syntax:

- op=get
- op2=agents_policy
- id=id_policy(optional)
- return_type=(json, csv, string)

Example

```
http://localhost/pandora_console/include/api.php?op=get&op2=agents_policy&apipass=1234&user=admin&pass=pandora&return_type=json.
```

get groups_policy

> = 7.0NG. 725

It returns all groups of a policy. If no policy id is specified, it returns all groups of all policies.

Call syntax:

- op=get
- op2=groups_policy
- id=id_policy(optional)
- return_type=(json, csv, string)

Example

```
http://localhost/pandora_console/include/api.php?op=get&op2=groups_policy&apipass=1234&user=admin&pass=pandora&return_type=json&id=2
```

get queue_policy

> = 7.0NG. 725

It returns the queue of a policy. If no id policy is specified, its returns all queues of all policies.

Call syntax:

- op=get
- op2=queue_policy
- id=id_policy(optional)
- return_type=(json, csv, string)

Example

```
http://localhost/pandora_console/include/api.php?op=get&op2=queue_policy&apipass=1234&user=admin&pass=pandora&return_type=json
```

get traps

> = 7.0NG. 728

It returns the specified source traps, they can also be filtered by timestamp and other data.

Call syntax:

- op=get
- op2=traps
- other=<optional parameters>
  - <timestamp>

  - <limit>
  - <offset>
  - <status>
  - <oid_custom>

Example

```
https://localhost/pandora_console/include/api.php?op=get&op2=traps&other=1537128
000%7C9000%7C0%7C1%7C.1.2.3.6.4&other_mode=url_encode_separator_%7C&apipass=1234
&user=admin&pass=pandora
```

get event_responses

It returns the response list of events the user has access to in Pandora FMS.

Call syntax:

- op=get (required)
- op2=event_reponses (required)
- return_type=csv|json (required)

Example

This example will return all system event responses in csv format.

```
http://localhost/pandora_console/include/api.php?op=get&op2=event_responses&retu
rn_type=csv&apipass=1234&user=admin&pass=pandora
```

get users

It returns the list of Pandora FMS users.

Call syntax:

- op=get (required)
- op2=users (required)
- return_type=csv|json (required)

Example

This example will return all users in csv format.

```
http://localhost/pandora_console/include/api.php?op=get&op2=users&return_type=cs
v&apipass=1234&user=admin&pass=pandora&other_mode=url_encode_separator_&
```

get list all user

This feature is in Command Center (Metaconsole).

It lists all Pandora FMS users.

Call syntax:

- op=get (required)
- op2=list_all_user(required)
- return_type=csv|json (required) Example

It can return Json or CSV, this return is remarked through the URL. Only the type of return is remarked, it does not use any specific variable

```
http://localhost/pandora_console/include/api.php?op=get&op2=list_all_user&return_type=json&apipass=1234&user=admin&pass=pandora
```

get info user name

This feature is in Command Center (Metaconsole).

It lists user data.

Call syntax:

- op=get (required)
- op2=list_all_user(required)
- return_type=csv|json (required)
- other= usernarme (requiered)

Example

It obtains all the information about a specific user. It can return Json or CSV. This return is specified through the URL. Only the username is used to obtain said information:

```
http://localhost/pandora_console/include/api.php?op=get&op2=info_user_name&return_type=json&other=admin&other_mode=url_encode_separator_|&apipass=1234&user=admin&pass=pandora
```

get user_profiles_info

It returns the list of Pandora FMS user profiles.

Call syntax:

- op=get (required)
- op2=user_profiles_info (required)
- return_type=csv|json (required)

Example

This example will return all user profiles in json format.

```
http://localhost/pandora_console/include/api.php?op=get&op2=user_profiles_info&r
eturn_type=csv&apipass=1234&user=admin&pass=pandora&other_mode=url_encode_separa
tor_&
```

get migrate_agent

> = 7.21 ONLY METACONSOLE

It looks up whether an especific agent exists in migration queue, returning the data of migration_queue.

- op=get
- op2=migrate_agent
- id=id_agente a migrar
- return_type=string, json, etc

Example

```
http://127.0.0.1/pandora_console/include/api.php?op=get&op2=migrate_agent&apipas
s=1234&user=admin&pass=pandora&id=2&return_type=json
```

get language

> = 7.0NG 730

It returns the language with which Pandora FMS is configured.

Call syntax:

- op=get
- op2=language

Example

```
http://127.0.0.1/pandora_console/include/api.php?apipass=1234&user=admin&pass=pa
ndora&op=get&op2=language
```

get session_timeout

> = 7.0NG 730

It returns the session time with which Pandora FMS is configured.

Call syntax:

- op=get
- op2=session_timeout

Example

```
http://127.0.0.1/pandora_console/include/api.php?apipass=1234&user=admin&pass=pa
ndora&op=get&op2=session_timeout
```

get groups

> = 7.0NG 732

It returns the existing groups.

Call syntax:

- op=get
- op2=groups
- other=<optional parameters>:

- <csv_separator>: ';' by default.
- <return_all_group>. It returns the group 'All'.
- <return_all_columns>. It returns all columns.

Example

```
http://127.0.0.1/pandora_console/include/api.php?op=get&op2=groups&other=;%7C1%7
C0&other_mode=url_encode_separator_%7C&apipass=1234&user=admin&pass=pandora
```

get filter user group

This feature is in Command Center (Metaconsole).

It obtains user groups through filtering.

Call syntax:

- op=get (requiered)
- op2=filter_user_group (requiered)

- return_type=csv|json(requiered)
- other= id group(requiered)|Enabled or not(Disabled)(requiered)

Example

It can return Json or CSV. This return is remarked through the URL.

Be careful, one of the two variables (group id or Enabled) may be empty, but at least one of them should be filled out.

```
http://localhost/pandora_console/include/api.php?op=get&op2=filter_user_group&return_type=json&other=0|0&other_mode=url_encode_separator_|&apipass=1234&user=admin&pass=pandora
```

get inventory_modules

It returns the list of Pandora inventory modules.

Call syntax:

- op=get (required)
- op2=inventory_modules (required)
- return_type=csv|json (required)

Examples

This example will return information from all of the system's inventories in json format.

```
http://localhost/pandora_console/include/api.php?op=get&op2=inventory_modules&return_type=json&apipass=1234&user=admin&pass=pandora
```

get inventory_modules_by_name

Obtains the inventory modules from an agent using the agent name.

Call syntax:

- op=get (required)
- op2=inventory_modules_by_name (required)
- id=<agent name> (required)

Examples

```
http://localhost/pandora_console/include/api.phpop=get&op2=inventory_modules_by_
name&id=name&return_type=csv&apipass=1234&user=admin&pass=pandora
```

get inventory_modules_by_alias

Obtains the inventory modules from an agent using the agent alias.

Call syntax:

- op=get (required)
- op2=inventory_modules_by_alias (required)
- id=<agent alias> (required)

Examples

```
http://localhost/pandora_console/include/api.phpop=get&op2=inventory_modules_by_
alias&id=alias&return_type=csv&apipass=1234&user=admin&pass=pandora
```

get inventory_module_data

Obtains the data from an inventory module using agent name and module.

Call syntax:

- op=get (required)
- op2=inventory_module_data (required)
- id=<agent name> (required)
- id2=<inventory module name> (optional)
- other=<serialized parameters> (optional), the following ones in this order:
  - <separator>
  - <date_from>: <year><month><day>T<hour>:<minute>
  - <date_to>: <year><month><day>T<hour>:<minute>
  - <use_agent_alias> (optional) values 0 and 1.

Examples

```
http://127.0.0.1/pandora_console/include/api.phpop=get&op2=inventory_module_data
&id=e14ae3b959b08a1fb7a057281401a08063cf04eb714efa5fbf1cf4043cfa1314&id2=Routes&
other=,%7C20191010T13:40%7C20191211T13:40%7C0&return_type=csv&other_mode=url_enc
ode_separator_%7C&apipass=1234&user=admin&pass=pandora
```

get list_collections

It returns the list of all the collections in Pandora FMS.

Call syntax:

- op=get (required)
- op2=list_collections (required)

Examples

This example will return information from all collections in json format.

```
http://localhost/pandora_console/include/api.php?op=get&op2=list_collections&api
pass=1234&user=admin&pass=pandora
```

get list_collection_files

It returns the list of all files in a collection. If it does not receive a collection id, it will return all files from all collections.

Call syntax:

- op=get (required)
- op2=list_collections_files (required)
- id=id_collection

Examples

This example will return the names of all files in a collection in json format.

```
http://localhost/pandora_console/include/api.php?op=get&op2=list_collection_file
s&id=1&apipass=1234&user=admin&pass=pandora
```

get event_mcid

It returns the ID of the events in the Command Center (Metaconsole), using the ID of the node and the ID of the event from the node.

This feature is in Command Center (Metaconsole).

Call syntax:

- op=get (required)
- op2=event_mcid (required)
- id=<server_id> (required) the ID of one of the nodes.
- id2=<id_source_event> (required) The ID of the event from the node.

Example

```
http://172.16.0.3/pandora_console/include/api.php?op=get&op2=event_mcid&return_t
```

```
ype=json&id=0&id2=0&apipass=1234&user=admin&pass=pandora
```

get is_centralized

Returns whether a node is centralized (`1`) or not (`0`). If the node does not exist it will return a text string informing about it. It can be used in Command Center (Metaconsole) or directly in the node.

Call syntax:

- op=get (required).
- op2=is_centralized (required).
- id=(required for Command Center (Metaconsole), no need for node).

Example at node:

```
http://localhost/pandora_console/include/api.php?op=get&op2=is_centralized&apipa
ss=1234&user=admin&pass=pandora
```

Example at Command Center (Metaconsole):

```
http://localhost/pandora_console/include/api.php?op=get&op2=is_centralized&id=3&
apipass=1234&user=admin&pass=pandora
```

SET

It sends data.

set new_agent

It creates a new agent with the data sent as parameters.

Call syntax:

- `op=set` (required).
- `op2=new_agent` (required).
- `id=` (Command Center (Metaconsole)) Numerical identifier of the node to which the agent will belong (required).
- `other=`<serialized parameters> (required). They are the agent configuration and data, serialized in the following order:
  - <agent_alias>
  - <ip>
  - <id_parent>
  - <id_group>
  - <cascade_protection>
  - <cascade_protection_module>

- <interval_sec>
- <id_os>
- <name_server>
- <custom_id>
- <learning_mode>
- <disabled>
- <description>
- <alias_as_name>

Examples

Call example for node (see call syntax):

```
.../include/api.php?op=set&op2=new_agent&other=agent_api|1.1.1.1|0|4|0|0|300|10|pandorafms|8|10||description|&other_mode=url_encode_separator_|&apipass=1234&user=admin&pass=pandora
```

Call example for Command Center (Metaconsole) (see call syntax):

```
.../include/api.php?op=set&op2=new_agent&id=1&other=agent_api|1.1.1.1|0|4|0|0|300|10|pandorafms|8|10||description|&other_mode=url_encode_separator_|&apipass=1234&user=admin&pass=pandora
```

- Note: If <alias_as_name> is `1`, the agent name will be the same as the alias. If it is `0`, the agent name will be automatically generated.

set update_agent

It updates an agent with data as parameters.

Call syntax:

- `op=set` (required)
- `op2=update_agent` (required)
- `id=`<id_agent> (required)
- `other=`<serialized parameters> (required). They are agent configuration and data, serialized in the following order:
  - <agent_alias>
  - <ip_address>
  - <id_parent>
  - <id_group>
  - <cascade_protection>
  - <cascade_protection_module>
  - <interval_sec>
  - <id_os>
  - <name_server>
  - <custom_id>
  - <learning_mode>
  - <disabled>
  - <description>

  - <os_version>

Example (see call syntax):

```
.../include/api.php?op=set&op2=update_agent&id=5&other=agent_name%7C1.1.1.1%7C0%
7C4%7C0%7C0%7C30%7C8%7Clocalhost.localdomain%7C%7C0%7C0%7Cla%20description|Ubunt
u&other_mode=url_encode_separator_%7C&apipass=1234&user=admin&pass=pandora
```

set update_agent_field

This call updates the field(s) of one or more Agents (the latter if it uses aliases and there are several matching ones.

Call syntax:

- op=set (required)
- op2=update_agent (required)
- id=<id_agent> or <alias> (required), <alias> depends of id2 parameter.
- id2= <0> if is<id_agente> (opcional), <1> if is <alias> (required).
- other=<serialized parameters> (required). They are agent configuration and data, serialized in the following order:

- <agent_alias>
- <ip>
- <id_parent>
- <id_group>
- <cascade_protection>
- <cascade_protection_module>
- <interval_sec>
- <id_os>
- <name_server>
- <custom_id>
- <learning_mode>
- <disabled>
- <description>

Examples

In case of using the Agent's identifier:

```
http://127.0.0.1/pandora_console/include/api.php?op=set&op2=update_agent_field&i
d=1&other=id_os,1|alias,pandora|direccion,192.168.10.16|id_parent,1cascade_prote
ction,1|cascade_protection_module,1|intervalo,5||modo|3|&other_mode=url_encode_s
eparator_|&apipass=1234&user=admin&pass=pandora
```

In case of using aliases, it modifies all the agents that contain that alias:

```
http://127.0.0.1/pandora_console/include/api.php?op=set&op2=update_agent_field&i
```

```
d=pandora&id2=1&other=id_os,1|alias,pandora|direccion,192.168.10.16|id_parent,1c
ascade_protection,1|cascade_protection_module,1|intervalo,5||modo|3|&other_mode=
url_encode_separator_|&apipass=1234&user=admin&pass=pandora
```

set delete_agent

It deletes an agent that has the name as parameter.

Call syntax:

- op=set (required)
- op2=delete_agent (required)
- id=<name_agent> (required). It should be an agent name.
- other (optional)=<use_agent_alias> (Values 0 and 1 supported)

Examples

```
http://127.0.0.1/pandora_console/include/api.php?op=set&op2=delete_agent&id=agen
te_erroneo&apipass=1234&user=admin&pass=pandora
```

set create_module_template

It creates an alert from a template as id parameter, in a module chosen by the module id agent id in other.

Call syntax:

- op=set (required)
- op2=create_module_template (required)
- id=<id_template> (required). It should be a template id.
- other=<id_module>|<id_agent>

Examples

```
http://127.0.0.1/pandora_console/include/api.php?op=set&op2=create_module_templa
te&id=1&other=1|10&other_mode=url_encode_separator_|&apipass=1234&user=admin&pas
s=pandora
```

set apply_module_template

It applies module template to agent.

Module template is an group which contains network check modules. These templates can be applied directly to agents, avoiding adding modules one by one.

Call Syntax:

- op=set (required)
- op2=apply_module_template (required)
- id<id_template> (required). Id of the template that will be applied on the module.
- id2<id_agente> (required). Id of the agent in which the modules will be created.

Examples

```
http://127.0.0.1/pandora_console/include/api.php?op=set&op2=apply_module_templat
e&id=2&id2=2&apipass=1234&user=admin&pass=pandora
```

set create_network_module

It creates a network module from data as parameters.

Call syntax:

- `op=set` (required).
- `op2=create_network_module` (required).
- `id=`< agent_name > (required). It should be an agent name.
- `other=`< serialized parameters > (required). They are module configuration and data, serialized in the following order:

1. < name_module >
2. < disabled >
3. < id_module_type >
4. < id_module_group >
5. < min_warning >
6. < max_warning >
7. < str_warning >
8. < min_critical >
9. < max_critical >
10. < str_critical >
11. < ff_threshold >
12. < history_data >
13. < ip_target >
14. < tcp_port >
15. < snmp_community >
16. < snmp_oid >
17. < module_interval >
18. < post_process >
19. < min_value >
20. < max_value >
21. < custom_id >
22. < description >
23. < enable_unknown_events >
24. < module_macros > It should be a base64 encoded JSON document made up by an object with one property for each macro. The format of the JSON properties should be “<macro name>”: “<macro value>”.

25. < each_ff >
26. < ff_threshold_normal >
27. < ff_threshold_warning >
28. < ff_threshold_critical >
29. < critical_inverse >
30. < warning_inverse >
31. < ff_type >
32. < ignore_unknown > Accepts values 0 and 1.
33. < number_of_intervals_in_warning > Allows state scaling by specifying the maximum number of consecutive intervals in which the module remains in warning state. If this value is exceeded, the module will escalate to critical status.

Example (see call syntax):

```
…/include/api.php?op=set&op2=create_network_module&id=example&other=test|0|7|1|10|15|0|16|18|0|15|0|127.0.0.1|0||0|180|0|0|0|0|latency%20ping|1|||||||||||&other_mode=url_encode_separator_|&apipass=1234&user=admin&pass=pandora
```

set create_plugin_module

It creates a module plugin with data as parameters.

Call syntax:

- `op=set` (required).
- `op2=create_plugin_module` (required).
- `id=`< agent_name > (required). It should be an agent name.
- `other=`< serialized parameters > (required). They are module configuration and data, serialized in the following order:

1. < name_module >
2. < disabled >
3. < id_module_type >
4. < id_module_group >
5. < min_warning >
6. < max_warning >
7. < str_warning >
8. < min_critical >
9. < max_critical >
10. < str_critical >
11. < ff_threshold >
12. < history_data >
13. < ip_target >
14. < tcp_port >
15. < snmp_community >
16. < snmp_oid >
17. < module_interval >
18. < post_process >
19. < min_value >
20. < max_value >

21. < custom_id >
22. < description >
23. < id_plugin >
24. < plugin_user >
25. < plugin_pass >
26. < plugin_parameter >
27. < enable_unknown_events >
28. < macros > The values must be in base 64 encoded JSON format.
29. < module_macros > It should be a base64 encoded JSON document made up by an object with one property for each macro. The format of the JSON properties should be “<macro name>”: “<macro value>”.
30. < each_ff >
31. < ff_threshold_normal >
32. < ff_threshold_warning >
33. < ff_threshold_critical >
34. < critical_inverse >
35. < warning_inverse >
36. < ff_type >
37. < use_agent_alias > Accepts values `0` and 1.
38. < ignore_unknown > Accepts values `0` and 1.
39. < number_of_intervals_in_warning > Enables state scaling by specifying the maximum number of consecutive intervals in which the module remains in the warning state. Exceeding this value will escalate the module to critical state.

Example (see call syntax):

```
…/include/api.php?op=set&op2=create_plugin_module&id=example&other=prueba|0|1|2|0|0||0|0||0|0|127.0.0.1|0||0|300|0|0|0|0|plugin%20module%20from%20api|4|2|admin|pass|-p%20max||||||||||||2001&other_mode=url_encode_separator_|&apipass=1234&user=admin&pass=pandora
```

The content of the JSON document must be an object composed of several objects with the following properties:

- `macro`: Macro name. It should be *_field1_*, *_field2_*, …, *_fieldN_*.
- `desc`: Descriptive macro name.
- `help`: Macro description.
- `value`: Macro value.
- `hide`: Set to 1 to hide the macro value (useful for storing passwords).

Example:

```
{
 "1": {
   "macro": "_field1_",
   "desc": "Target IP",
   "help": "",
   "value": "192.168.0.1",
   "hide": ""
 },
 "2": {
   "macro": "_field2_",
   "desc": "Port",
   "help": "",
   "value": "80",
   "hide": ""
 }
}
```

set create_data_module

It creates a module with the given parameters.

With this call, you may add database module data but the configuration file of the agents associated to the module cannot be modified.

Call syntax:

- `op=get` (required).
- `op2=create_data_module` (required)
- `id=`< agent_name > (required). It should be an agent name.
- `other=`< serialized parameters > (required). They are module configuration and data, serialized in the following order:

1. < name_module >
2. < disabled >
3. < id_module_type >
4. < description >
5. < id_module_group >
6. < min_value >
7. < max_value >
8. < post_process >
9. < module_interval >
10. < min_warning >
11. < max_warning >
12. < str_warning >
13. < min_critical >
14. < max_critical >
15. < str_critical >
16. < history_data >
17. < enable_unknown_events >
18. < module_macros > It should be a base64 encoded JSON document made up by an object with one property for each macro. The format of the JSON properties should be “< macro name >”: “< macro value >”.
19. < ff_threshold >
20. < each_ff >

21. < ff_threshold_normal >
22. < ff_threshold_warning >
23. < ff_threshold_critical >
24. < ff_timeout >
25. < critical_inverse >
26. < warning_inverse >
27. < ff_type >
28. < ignore_unknown > Values `0` and `1` supported.
29. < number_of_intervals_in_warning > Allows state scaling by specifying the maximum number of consecutive intervals in which the module remains in warning state. If this value is exceeded, the module will escalate to critical status.

Example (see call syntax):

```
…/include/api.php?op=set&op2=create_data_module&id=test&other=test2|0|1|data%20m
odule%20from%20api|1|10|20|10.50|180|10|15||16|20||0|||||||||||||2001&other_mode
=url_encode_separator_|&apipass=1234&user=admin&pass=pandora
```

set create_snmp_module

It creates an SNMP module.

Call syntax:

- `op=set` (required).
- `op2=create_snmp_module` (required).
- `id=`< agent_name > (required). It should be an agent name.
- `other=`< serialized parameters > (required). They are module configuration and data, serialized in the following order:

1. < name_module >
2. < disabled >
3. < id_module_type >
4. < id_module_group >
5. < min_warning >
6. < max_warning >
7. < str_warning >
8. < min_critical >
9. < max_critical >
10. < str_critical >
11. < ff_threshold >
12. < history_data >
13. < ip_target >
14. < module_port >
15. < snmp_version >
16. < snmp_community >
17. < snmp_oid >
18. < module_interval >
19. < post_process >
20. < min_value >

21. < max_value >
22. < custom_id >
23. < description >
24. < snmp3_priv_method [AES|DES] >
25. < snmp3_priv_pass >
26. < snmp3_sec_level [authNoPriv|authPriv|noAuthNoPriv] >
27. < snmp3_auth_method [MD5|SHA] >
28. < snmp3_auth_user >
29. < snmp3_auth_pass >
30. < enable_unknown_events >
31. < each_ff >
32. < ff_threshold_normal >
33. < ff_threshold_warning >
34. < ff_threshold_critical >
35. < ff_type >
36. < ignore_unknown > Accepts values `0` and `1`.
37. < number_of_intervals_in_warning > Allows state scaling by specifying the maximum number of consecutive intervals in which the module remains in warning state. If this value is exceeded, the module will escalate to critical status.

Example (see call syntax):

First example (snmp v: 1):

```
…/include/api.php?op=set&op2=create_snmp_module&id=example&other=test5|0|15|1|10|15||16|18||15|0|127.0.0.1|60|1|public|.1.3.6.1.2.1.1.1.0|180|0|0|0|0|SNMP%20module%20from%20API||||||||2001&other_mode=url_encode_separator_|&apipass=1234&user=admin&pass=pandora
```

Second example (snmp v: 3, snmp3_priv_method: AES, snmp3_priv_pass: example_priv_passw, snmp3_sec_level: authNoPriv, snmp3_auth_method:MD5, snmp3_auth_user: example_user, snmp3_auth_pass: example_priv_passw):

```
…/include/api.php?op=set&op2=create_snmp_module&id=example&other=test7|0|15|1|10|15||16|18||15|0|127.0.0.1|60|3|public|.1.3.6.1.2.1.1.1.0|180|0|0|0|0|SNMP%20module%20from%20API|AES|example_priv_passw|authNoPriv|MD5|example_user|example_auth_passw|||||||||2001&other_mode=url_encode_separator_|&apipass=1234&user=admin&pass=pandora
```

set update_network_module

It updates the network module.

Call syntax:

- `op=get` (required).
- `op2=update_network_module` (required).
- `id=`< id_modulo > (required) It should be a module id (`tagente_modulo`).
- `other=`< serialized parameters > (required) They are module configuration and data, serialized in

the following order:

1. < id_agent >
2. < disabled >
3. < id_module_group >
4. < min_warning >
5. < max_warning >
6. < str_warning >
7. < min_critical >
8. < max_critical >
9. < str_critical >
10. < min_ff_even >
11. < ff_threshold >
12. < history_data >
13. < ip_target >
14. < tcp_port >
15. < snmp_community >
16. < snmp_oid >
17. < module_interval >
18. < post_process >
19. < min_value >
20. < max_value >
21. < custom_id >
22. < description >
23. < disabled_types_event >
24. < module_macros > It should be a base64 encoded JSON document made up by an object with one property for each macro. The format of the JSON properties should be “<macro name>”: “<macro value>”.
25. < each_ff >
26. < ff_threshold_normal >
27. < ff_threshold_warning >
28. < ff_threshold_critical >
29. < critical_inverse >
30. < warning_inverse >
31. < ff_type >
32. < number_of_intervals_in_warning > Allows state scaling by specifying the maximum number of consecutive intervals in which the module remains in warning state. If this value is exceeded, the module will escalate to critical status.

Example (see call syntax):

```
…/include/api.php?op=set&op2=update_network_module&id=132&other=|0|6|2|10|15||16
|18||7|0|127.0.0.1|0||0|300|30.00|0|0|0|latency%20ping%20modified%20by%20the%20A
pi|||||||||||2020&other_mode=url_encode_separator_|&apipass=1234&user=admin&pass
=pandora
```

set update_plugin_module

It updates the plugin module.

Call syntax:

- `op=set` (required).
- `op2=update_plugin_module` (required).
- `id=`< module_id > (required). It should be a module id.
- `other=`< serialized parameters > (required). They are module configuration and data, serialized in the following order:

1. < id_agent >
2. < disabled >
3. < id_module_group >
4. < min_warning >
5. < max_warning >
6. < str_warning >
7. < min_critical >
8. < max_critical >
9. < str_critical >
10. < ff_threshold >
11. < history_data >
12. < ip_target >
13. < tcp_port >
14. < snmp_community >
15. < snmp_oid >
16. < module_interval >
17. < post_process >
18. < min_value >
19. < max_value >
20. < custom_id >
21. < description >
22. < id_plugin >
23. < plugin_user >
24. < plugin_pass >
25. < plugin_parameter >
26. < disabled_types_event >
27. < macros > Values must be in base 64 encoded JSON format.
28. < module_macros > The format of the JSON properties should be "<macro name>": "<macro value>". See example at the end of this section.
29. < each_ff >
30. < ff_threshold_normal >
31. < ff_threshold_warning >
32. < ff_threshold_critical >
33. < critical_inverse>
34. < warning_inverse>
35. < policy_linked >
36. < ff_type >
37. < ignore_unknown > Accepts values `0` and `1`.
38. < number_of_intervals_in_warning > Allows state scaling by specifying the maximum number of consecutive intervals in which the module remains in warning state. If this value is exceeded, the module will escalate to critical status.

Example (see call syntax):

```
…/include/api.php?op=set&op2=update_plugin_module&id=2343&other=44|0|2|0|0||0|0|
|0|0|127.0.0.1|0||0|300|0|0|0|0|pluginmodule%20from%20api|2|admin|pass|-
p%20max&other_mode=url_encode_separator_|&apipass=1234&user=admin&pass=pandora
```

The content should be an object made up by objects with the following properties:

- *"macro"*: Macro name. Should be *_field1_*, *_field2_*, …, *_fieldN_*.
- *"desc"*: Descriptive name of the macro.
- *"help"*: Macro description.
- *"value"*:Macro value.

JSON macro format example:

```
{
 "1": {
   "macro": "_field1_",
   "desc": "Target IP",
   "help": "",
   "value": "192.168.0.1"
 },
 "2": {
   "macro": "_field2_",
   "desc": "Port",
   "help": "",
   "value": "80"
 }
}
```

Another example with state scaling:

```
…/include/api.php?op=set&op2=update_plugin_module&id=135&other=|0|2|0|0||0|0||0|
0|127.0.0.1|0||0|300|0|0|0|0|plugin%20module%20from%20api|2|admin|pass|-
p%20max|||||||||||||2020&other_mode=url_encode_separator_|&apipass=1234&user=adm
in&pass=pandora
```

set update_data_module

With this call, a database module data can be added but the configuration file of the agents associated to the module cannot be modified.

It updates the local module.

Call syntax:

- `op=set` (required).
- `op2=update_data_module` (required).

- id=< id_module_agent > (required) module's numeric ID to update.
- other=< serialized parameters > (required) module data and module configuration in serialized order:

1. < id_agent >
2. < disabled > use 1 for disable, 0 *for no changes*.
3. < description >
4. < id_module_group > see get_module_groups.
5. < min >
6. < max >
7. < post_process > (Only for Software Agents with remote configuration enabled).
8. < module_interval >
9. < min_warning >
10. < max_warning >
11. < str_warning >
12. < min_critical >
13. < max_critical >
14. < str_critical >
15. < history_data > If it is set at 0, module data will not be saved at tagente_datos, only tagent_estado will be updated.
16. < disabled_types_event > JSON forma, example: {“going_unknown”:1} .
17. < module_macros > It should be a base64 encoded JSON document made up by an object with one property for each macro. The format of the JSON properties should be “*<macro name>*”: “*<macro value>*”.
18. < min_ff_event > Number of times a change of state condition has to occur before the change of state can take place (*flipflop*).
19. < each_ff >
20. < min_ff_event_normal >
21. < min_ff_event_warning >
22. < min_ff_event_critical >
23. < ff_timeout >
24. < critical_inverse >
25. < warning_inverse >
26. < policy_linked >
27. < ff_type >
28. < ignore_unknown > Accepts values 0 and 1.

Example (see call syntax):

```
.../include/api.php?op=set&op2=update_data_module&id=123&other=|0|data%20module%
20modified%20from%20API|6|0|0|50.00|300|10|15||16|18||0||||||||||||||2020&other_
mode=url_encode_separator_|&apipass=1234&user=admin&pass=pandora
```

set update_data_module_policy

It updates a data module in a policy and returns an id from the new module.

Call syntax:

- op=set (required).

- op2=update_data_module_policy (required).
- id=< id_policy > (required). Numeric identification of the target policy.
- other=< serialized parameters > (required). They are module configuration and data, serialized in the following order:

1. < id_policy_module >
2. < description >
3. < id_module_group >
4. < min >
5. < max >
6. < post_process >
7. < module_interval >
8. < min_warning >
9. < max_warning >
10. < min_critical >
11. < max_critical >
12. < str_critical >
13. < history_data >
14. < configuration_data >
15. < disabled_types_event >
16. < module_macros > It should be a base64 encoded JSON document made up by an object with one property for each macro. The format of the JSON properties should be: < macro name >:< macro value >.
17. < ignore_unknown > Accepts values 0 and 1.

Example (see call syntax):

```
.../include/api.php?op=set&op2=update_data_module_policy&id=1&other=10|data%20module%20updated%20by%20Api|2|0|0|50.00|10|20|180||21|35||1|module_begin%0dmodule_name%20pandora_process%0dmodule_type%20generic_data%0dmodule_exec%20ps%20aux%20|%20grep%20pandora%20|%20wc%20-l%0dmodule_end&other_mode=url_encode_separator_|&apipass=1234&user=admin&pass=pandora
```

set update_snmp_module

It updates an SNMP module.

Call syntax:

- op=set (required).
- op2=update_snmp_module (required).
- id=< module_id > (required). It should be a module id.
- other=< serialized parameters > (required). They are module configuration and data, serialized in the following order:

1. < id_agent >
2. < disabled >
3. < id_module_group >
4. < min_warning >
5. < max_warning >

6. < str_warning >
7. < min_critical >
8. < max_critical >
9. < str_critical >
10. < ff_threshold >
11. < history_data >
12. < ip_target >
13. < module_port >
14. < snmp_version >
15. < snmp_community >
16. < snmp_oid >
17. < module_interval >
18. < post_process >
19. < min_value >
20. < max_value >
21. < custom_id >
22. < description >
23. < snmp3_priv_method [AES|DES] >
24. < snmp3_priv_pass >
25. < snmp3_sec_level [authNoPriv|authPriv|noAuthNoPriv] >
26. < snmp3_auth_method [MD5|SHA] >
27. < snmp3_auth_user >
28. < snmp3_auth_pass >
29. < disabled_types_event >
30. < each_ff >
31. < ff_threshold_normal >
32. < ff_threshold_warning >
33. < ff_threshold_critical>
34. < policy_linked >
35. < ff_type >
36. < ignore_unknown > Accepts values `0` and `1`.
37. < number_of_intervals_in_warning > Allows state scaling by specifying the maximum number of consecutive intervals in which the module remains in warning state. If this value is exceeded, the module will escalate to critical status.

Example (see call syntax):

(SNMP v: 3, snmp3_priv_method: AES, snmp3_priv_pass: example_priv_passw, snmp3_sec_level: authNoPriv, snmp3_auth_method:MD5, snmp3_auth_user: pepito_user, snmp3_auth_pass: example_priv_passw):

```
…/include/api.php?op=set&op2=update_snmp_module&id=33432&other=44|0|6|20|25||26|
30||15|1|127.0.0.1|60|3|public|.1.3.6.1.2.1.1.1.0|180|50.00|10|60|0|SNMP%20modul
e&20modified%20by%20API|AES|example_priv_passw|authNoPriv|MD5|example_user|examp
le_auth_passw&other_mode=url_encode_separator_|&apipass=1234&user=admin&pass=pan
dora
```

Another example with state scaling and SNMP v1:

```
…/include/api.php?op=set&op2=update_snmp_module&id=137&other=|0|6|20|25||26|30||
15|1|127.0.0.1|60|1|public|.1.3.6.1.2.1.1.1.0|180|50.00|10|60|||||||||||2020&oth
```

```
er_mode=url_encode_separator_|&apipass=1234&user=admin&pass=pandora
```

Another example with state scaling and SNMP v3:

```
…/include/api.php?op=set&op2=update_snmp_module&id=138&other=|0|6|20|25||26|30||
15|1|127.0.0.1|60|3|public|.1.3.6.1.2.1.1.1.0|180|50.00|10|60|0|SNMP%20module%20
modified%20by%20API|AES|example_priv_passw|authNoPriv|MD5|example_user|example_a
uth_passw|||||||||2020&other_mode=url_encode_separator_|&apipass=1234&user=admin
&pass=pandora
```

set apply_policy

It applies the policy, once it has gone through id, in one or several agents.

Call syntax:

- op=set (required)
- op2=apply_policy (required)
- id=<id_policy> (required)
- id2=<id_agent> (optional). Id or name of the agent as indicated in the other parameter. If it is empty, the policy will be applied to all its agents.
- other=<serialized parameters>
  - <name_agent (Integer)>. It indicates whether the agent will be sent by Id (0), by name (1) or by alias (2).
  - <server_id> (required when using Command Center (Metaconsole)). Id of the server in which the policy will be applied.

Examples

```
http://192.168.70.102/pandora_console/include/api.php?op=set&op2=apply_policy&id
=22&id2=3e&other=0|1&other_mode=url_encode_separator_|&apipass=1234&user=admin&p
ass=pandora
```

set apply_all_policies

It applies all policies within Pandora FMS.

Call syntax:

- op=set (required)
- op2=apply_all_policies (required)

Examples

```
http://127.0.0.1/pandora_console/include/api.php?op=set&op2=apply_all_policies&a
pipass=1234&user=admin&pass=pandora
```

set add_network_module_policy

It adds a network module in the policy after it has gone through id in the parameter.

Call syntax:

- `op=set` (required).
- `op2=add_network_module_policy` (required).
- `id=`< id_policy > (required). It should be a policy Id.
- `other=`< serialized parameters > (required). They are module configuration and data, serialized in the following order:

1. < name_module >
2. < id_module_type >
3. < description >
4. < id_module_group >
5. < min_value >
6. < max_value >
7. < post_process >
8. < module_interval >
9. < min_warning >
10. < max_warning >
11. < str_warning >
12. < min_critical >
13. < max_critical >
14. < str_critical >
15. < history_data >
16. < ff_threshold >
17. < disabled >
18. < module_port >
19. < snmp_community >
20. < snmp_oid >
21. < custom_id >
22. < enable_unknown_events >
23. < module_macros > It should be a base64 encoded JSON document made up by an object with one property for each macro. The format of the JSON properties should be "<macro name>": "<macro value>".
24. < each_ff >
25. < ff_threshold_normal >
26. < ff_threshold_warning >
27. < ff_threshold_critical >
28. < ff_type >
29. < ignore_unknown > Accepts values `0` and `1`.

Example (see call syntax):

```
…/include/api.php?op=set&op2=add_network_module_policy&id=1&other=network_module
_policy_example_name|6|network%20module%20created%20by%20Api|2|0|0|50.00|180|10|
20||21|35||1|15|0|66|||0&other_mode=url_encode_separator_|&apipass=1234&user=adm
```

```
in&pass=pandora
```

set add_plugin_module_policy

It adds a plugin module in the policy that has gone through id in the parameter.

Call syntax:

- `op=set` (required).
- `op2=add_plugin_module_policy` (required).
- `id=`< id_policy > (required). It should be a policy Id.
- `other=`< serialized parameters > (required). They are module configuration and data, serialized in the following order:

1. < name_module >
2. < id_module_type >
3. < description >
4. < id_module_group >
5. < min_value >
6. < max_value >
7. < post_process >
8. < module_interval >
9. < min_warning >
10. < max_warning >
11. < str_warning >
12. < min_critical >
13. < max_critical >
14. < str_critical >
15. < history_data >
16. < ff_threshold >
17. < disabled >
18. < module_port >
19. < snmp_community >
20. < snmp_oid >
21. < custom_id >
22. < enable_unknown_events >
23. < module_macros > It should be a base64 encoded JSON document made up by an object with one property for each macro. The format of the JSON properties should be “<macro name>”: “<macro value>”. See example at end of this section.
24. < each_ff >
25. < ff_threshold_normal >
26. < ff_threshold_warning >
27. < ff_threshold_critical >
28. < ff_type >
29. < ignore_unknown > Accepts values `0` and `1`.

Example (see call syntax):

```
…/include/api.php?op=set&op2=add_network_module_policy&id=1&other=network_module
_policy_example_name|6|network%20module%20created%20by%20Api|2|0|0|50.00|180|10|
20||21|35||1|15|0|66|||0&other_mode=url_encode_separator_|&apipass=1234&user=adm
in&pass=pandora
```

The content should be an object made up by objects with the following properties:

- *"macro"*: Macro name. It should be *_field1_*, *_field2_*, ..., *_fieldN_*.
- *"desc"*: Descriptive macro name.
- *"help"*: Macro description.
- *"value"*: Macro value.

It should be a base64 encoded JSON document.

Example

```
{
 "1": {
   "macro": "_field1_",
   "desc": "Target IP",
   "help": "",
   "value": "192.168.0.1"
 },
 "2": {
   "macro": "_field2_",
   "desc": "Port",
   "help": "",
   "value": "80"
 }
}
```

set add_data_module_policy

It adds a local module in the policy which has gone through id in the parameter.

Call syntax:

- op=set (required)
- op2=add_data_module_policy (required)
- id=<id_policy> (required). It should be a policy Id.
- other=<serialized parameters> (required). They are module configuration and data, serialized in the following order:
    - <name_module>
    - <id_module_type>
    - <description>
    - <id_module_group>
    - <min_value>
    - <max_value>
    - <post_process>
    - <module_interval>
    - <min_warning>

  - <max_warning>
  - <str_warning>
  - <min_critical>
  - <max_critical>
  - <str_critical>
  - <history_data>
  - <configuration_data>. This is the definition block of the agent that will be entered in the config file of the policy agent.
  - <enable_unknown_events> (only in version 5 or later)
  - <module_macros> (only in version 5). It should be a base64 encoded JSON document made up by an object with one property for each macro. The format of the JSON properties should be “<macro name>”: “<macro value>”.
  - <ff_threshold> (only in version 5.1)
  - <each_ff> (only in version 5.1)
  - <ff_threshold_normal> (only in version 5.1)
  - <ff_threshold_warning> (only in version 5.1)
  - <ff_threshold_critical> (only in version 5.1)
  - <ff_timeout> (only in version 5.1)
  - <ff_type> (only in version 734)

Examples

```
http://127.0.0.1/pandora_console/include/api.php?op=set&op2=add_data_module_poli
cy&id=1&other=data_module_policy_example_name~2~data%20module%20created%20by%20A
pi~2~0~0~50.00~10~20~180~~21~35~~1~module_begin%0dmodule_name%20pandora_process%
0dmodule_type%20generic_data%0dmodule_exec%20ps%20aux%20|%20grep%20pandora%20|%2
0wc%20-
l%0dmodule_end&other_mode=url_encode_separator_~&apipass=1234&user=admin&pass=pa
ndora
```

set update_plugin_module_policy

Updates a plugin module in the policy passed by identifier in the parameter.

Call syntax:

- `op=set` (required).
- `op2=update_plugin_module_policy` (required).
- `id=` < id_policy > (required) must be a policy identifier.
- `other=` < serialized parameters > (required) are the module configuration and data, serialized in the following order:

1. < id_policy_module >
2. < disabled >
3. < id_module_group >
4. < min_warning >
5. < max_warning >
6. < str_warning >
7. < min_critical >
8. < max_critical >

9. < str_critical >
10. < ff_threshold >
11. < history_data >
12. < module_port >
13. < snmp_community >
14. < snmp_oid >
15. < module_interval >
16. < post_process >
17. < min_value >
18. < max_value >
19. < custom_id >
20. < description >
21. < id_plugin >
22. < plugin_user >
23. < plugin_pass >
24. < plugin_parameter >
25. < disabled_types_event >
26. < macros > Values must be in base 64 encoded JSON format. See example at the end of this section.
27. < module_macros > The values must be in base 64 encoded JSON format. The format of the data in the JSON document must be “< *macro name* >”: “< *macro value* >”.
28. < ignore_unknown > Accepts values `0` and `1`.

Example (see call syntax):

```
…/include/api.php?op=set&op2=update_plugin_module_policy&id=1&other=23|0|1|0|0||0|0||15|0|166|||180|150.00|0|0|0|plugin%20module%20updated%20from%20api|2|example_user|pass|-p%20min&other_mode=url_encode_separator_|
```

The content of the JSON document must be an object composed of several objects with the following properties:

- “*macro*”: Name of the macro. Must be *_field1_*, *_field2_*, …, *_fieldN_*.
- “*desc*”: Descriptive name of the macro.
- “*help*”: Macro description.
- “*value*”: Macro value.

set add_snmp_module_policy

It adds a SNMP module in the policy which has gone through id in the parameter.

Call syntax:

- `op=set` (required).
- `op2=add_snmp_module_policy` (required).
- `id=`< id_policy > (required). It should be a policy Id.
- `other=`< serialized parameters > (required). They are module configuration and data, serialized in the following order:

1. < name_module >
2. < disabled >

3. < id_module_type >
4. < id_module_group >
5. < min_warning >
6. < max_warning >
7. < str_warning >
8. < min_critical >
9. < max_critical >
10. < str_critical >
11. < ff_threshold >
12. < history_data >
13. < module_port >
14. < snmp_version >
15. < snmp_community >
16. < snmp_oid >
17. < module_interval >
18. < post_process >
19. < min_value >
20. < max_value >
21. < custom_id >
22. < description >
23. < snmp3_priv_method [AES|DES] >
24. < snmp3_priv_pass >
25. < snmp3_sec_level [authNoPriv|authPriv|noAuthNoPriv] >
26. < snmp3_auth_method [MD5|SHA] >
27. < snmp3_auth_user >
28. < snmp3_auth_pass >
29. < enable_unknown_events >
30. < each_ff >
31. < ff_threshold_normal >
32. < ff_threshold_warning >
33. < ff_threshold_critical >
34. < ff_type >
35. < ignore_unknown > Accepts values `0` and `1`.

Example (see call syntax):

```
…/include/api.php?op=set&op2=add_snmp_module_policy&id=1&other=example%20SNMP%20
module%20name|0|15|2|0|0||0|0||15|1|66|3|public|.1.3.6.1.2.1.1.1.0|180|50.00|10|
60|0|SNMP module modified by
API|AES|example_priv_passw|authNoPriv|MD5|example_user|example_auth_passw&other_
mode=url_encode_separator_|&apipass=1234&user=admin&pass=pandora
```

set add_agent_policy_by_id

It adds an agent to a policy using the agent ID

Call syntax:

- op=set (required)
- op2=add_agent_policy_by_id (mandatory)

- id=<id_policy> (required). It must be a policy Id.
- other=<serialized parameters> (required). These are agent configuration and data, serialized in the following order:
    - <id_agent>
    - <id_node> (required when using the Command Center (Metaconsole)). ID of the node the agent you wish to add to the policy belongs to.

Examples

- Node:

```
http://127.0.0.1/pandora_console/include/api.php?op=set&op2=add_agent_policy_by_id&id=2&other=1&other_mode=url_encode_separator_%7C&apipass=1234&user=admin&pass=pandora
```

- Command Center (Metaconsole):

```
http://127.0.0.1/pandora_console/include/api.php?op=set&op2=add_agent_policy_by_id&id=2&other=1%7C1&other_mode=url_encode_separator_%7C&apipass=1234&user=admin&pass=pandora
```

set add_agent_policy_by_name

It adds an agent to a policy using the agent name.

Call syntax:

- op=set (required)
- op2=add_agent_policy_by_name (required)
- id=<id_policy> (required). It must be a policy Id.
- other=<serialized parameters> (required). These are agent configuration and data, serialized in the following order:
    - <agent_name>

Examples

```
http://127.0.0.1/pandora_console/include/api.php?op=set&op2=add_agent_policy_by_name&id=4&other=e76774025b24057cc71df514f27027c43484c3af766ed40f259a86a4fd568f9d&other_mode=url_encode_separator_%7C&apipass=1234&user=admin&pass=pandora
```

set remove_agent_from_policy_by_id

It deletes an agent from a policy using the agent ID.

Call syntax:

- op=set (required)
- op2=remove_agent_from_policy_by_id (required)

- id=<id_política> (required). It must be a policy Id.
- other=<serialized parameters> (required). These are the agent's configuration and data, serialized in the following order:
    - <id_agent>
    - <id_node> (required when using the Command Center (Metaconsole)). ID of the node the agent you wish to remove from the policy belongs to.

Examples

- Node:

```
http://127.0.0.1/pandora_console/include/api.php?op=set&op2=remove_agent_from_po
licy_by_id&id=2&other=2&other_mode=url_encode_separator_%7C&apipass=1234&user=ad
min&pass=pandora
```

- Command Center (Metaconsole):

```
http://127.0.0.1/pandora_console/include/api.php?op=set&op2=remove_agent_from_po
licy_by_id&id=4&other=1%7C1&other_mode=url_encode_separator_%7C&apipass=1234&use
r=admin&pass=pandora
```

set remove_agent_from_policy_by_name

It deletes an agent from a policy using the agent name.

Call syntax:

- op=set (required)
- op2=remove_agent_from_policy_by_name (required)
- id=<id_policy> (required). It must be a policy Id.
- other=<serialized parameters> (required). These are the agent's configuration and data, serialized in the following order:
    - <agent_name>

Examples

```
http://127.0.0.1/pandora_console/include/api.php?op=set&op2=remove_agent_from_po
licy_by_name&id=4&other=e76774025b24057cc71df514f27027c43484c3af766ed40f259a86a4
fd568f9d&other_mode=url_encode_separator_%7C&apipass=1234&user=admin&pass=pandor
a
```

set new_network_component

It creates a new network component.

Call syntax:

- op=set (required)
- op2=new_network_component (required)

- id=<network_component_name> (required). It should be the network component name.
- other=<serialized parameters> (required). They are agent configuration and data of the network component, serialized in the following order:
    - <network_component_type>
    - <description>
    - <module_interval>
    - <max_value>
    - <min_value>
    - <snmp_community>
    - <id_module_group>
    - <max_timeout>
    - <history_data>
    - <min_warning>
    - <max_warning>
    - <str_warning>
    - <min_critical>
    - <max_critical>
    - <str_critical>
    - <ff_threshold>
    - <post_process>
    - <network_component_group>
    - <enable_unknown_events> (only in version 5)
    - <each_ff> (only in version 5.1)
    - <ff_threshold_normal> (only in version 5.1)
    - <ff_threshold_warning> (only in version 5.1)
    - <ff_threshold_critical> (only in version 5.1)
    - <ff_type> (only in version 734)

Examples

```
http://127.0.0.1/pandora_console/include/api.php?op=set&op2=new_network_componen
t&id=example_network_component_name&other=7|network%20component%20created%20by%2
0Api|300|30|10|public|3||1|10|20|str|21|30|str1|10|50.00|12&other_mode=url_encod
e_separator_|&apipass=1234&user=admin&pass=pandora
```

set new_plugin_component

It creates a new plugin component.

Call syntax:

- op=set (required)
- op2=new_plugin_component (required)
- id=<plugin_component_name> (required). It should be the plugin component name.
- other=<serialized parameters> (required). They are agent configuration and data of the plugin component, serialized in the following order:
    - <plugin_component_type>
    - <description>
    - <module_interval>
    - <max_value>

- <min_value>
- <module_port>
- <id_module_group>
- <id_plugin>
- <max_timeout>
- <history_data>
- <min_warning>
- <max_warning>
- <str_warning>
- <min_critical>
- <max_critical>
- <str_critical>
- <ff_threshold>
- <post_process>
- <plugin_component_group>
- <enable_unknown_events> (only in version 5)
- <each_ff> (only in version 5.1)
- <ff_threshold_normal> (only in version 5.1)
- <ff_threshold_warning> (only in version 5.1)
- <ff_threshold_critical> (only in version 5.1)
- <ff_type> (only in version 734)

Examples

```
http://127.0.0.1/pandora_console/include/api.php?op|http://127.0.0.1/pandora_con
sole/include/api.php?op=set&op2=
new_plugin_component&id=example_plugin_component_name&other
=2|plugin%20component%20created%20by%20Api|300|30|10|66|3|2|example_user|example
_pass|-
p%20max||1|10|20|str|21|30|str1|10|50.00|12&other_mode=url_encode_separator_|&ap
ipass =1234&user=admin&pass=pandora

=== set new_snmp_component ===

It creates a new SNMP component.

Call syntax:

  * op=set (required)
  * op2=new_snmp_component (required)
  * id=<snmp_component_name> (required). It should be the SNMP component name.
  * other=<serialized parameters> (required). These are the configuration and
data of the snmp component, serialized in the following order:
      * <snmp_component_type>
      * <description>
      * <module_interval>
      * <max_value>
      * <min_value>
      * <id_module_group>
      * <max_timeout>
      * <history_data>
      * <min_warning>
```

```
        * <max_warning>
        * <str_warning>
        * <min_critical>
        * <max_critical>
        * <str_critical>
        * <ff_threshold>
        * <post_process>
        * <snmp_version>
        * <snmp_oid>
        * <snmp_community>
        * <snmp3_auth_user>
        * <snmp3_auth_pass>
        * <module_port>
        * <snmp3_privacy_method>
        * <snmp3_privacy_pass>
        * <snmp3_auth_method>
        * <snmp3_security_level>
        * <snmp_component_group>
        * <enable_unknown_events> (only in version 5)
        * <each_ff> (only in version 5.1)
        * <ff_threshold_normal> (only in version 5.1)
        * <ff_threshold_warning> (only in version 5.1)
        * <ff_threshold_critical> (only in version 5.1)
        * <ff_type> (only in version 734)

**Examples**
<code>

http://127.0.0.1/pandora_console/include/api.php?op=set&op2=new_snmp_component&id=example_snmp_component_name&other=16|SNMP%20component%20created%20by%20Api|300|30|10|3||1|10|20|str|21|30|str1|15|50.00|3|.1.3.6.1.2.1.2.2.1.8.2|public|example_auth_user|example_auth_pass|66|AES|example_priv_pass|MD5|authNoPriv|12&other_mode=url_encode_separator_|&apipass=1234&user=admin&pass=pandora
```

set new_local_component

It creates a new local component.

Call syntax:

- op=set (required)
- op2=new_local_component (required)
- id=<local_component_name> (required). It should be a local component name.
- other=<serialized parameters> (required). They are configuration and data of the local component, serialized in the following order:
  - <description>
  - <id_os>
  - <local_component_group>
  - <configuration_data>. This is the module configuration block.
  - <enable_unknown_events> (only in version 5)

  - <ff_threshold> (only in version 5.1)
  - <each_ff> (only in version 5.1)
  - <ff_threshold_normal> (only in version 5.1)
  - <ff_threshold_warning> (only in version 5.1)
  - <ff_threshold_critical> (only in version 5.1)
  - <ff_timeout> (only in version 5.1)
  - <ff_type> (only in version 734)

Examples

```
http://127.0.0.1/pandora_console/include/api.php?op=set&op2=new_local_component&id=example_local_component_name&other=local%20component%20created%20by%20Api~5~12~module_begin%0dmodule_name%20example_local_component_name%0dmodule_type%20generic_data%0dmodule_exec%20ps%20|%20grep%20pid%20|%20wc%20-l%0dmodule_interval%202%0dmodule_end&other_mode=url_encode_separator_~&apipass=1234&user=admin&pass=pandora
```

set create_alert_template

It creates an alert template.

Call Syntax:

- op=set (required)
- op2=create_alert_template (required)
- id=<template_name> (required). It is the template name.
- other=<serialized parameters> (required). They are template configuration and data, serialized in the following order:
  - <type [regex|max_min|max|min|equal|not_equal|warning|critical|onchange|unknown|always|not_normal]>
  - <description>
  - <id_alert_action>
  - <field1>
  - <field2>
  - <field3>
  - <value>
  - <matches_value>
  - <max_value>
  - <min_value>
  - <time_threshold>
  - <max_alerts>
  - <min_alerts>
  - <time_from>
  - <time_to>
  - <monday>
  - <tuesday>
  - <wednesday>
  - <thursday>

- <friday>
- <saturday>
- <sunday>
- <recovery_notify>
- <field2_recovery>
- <field3_recovery>
- <priority>
- <id_group>
- <special_day>
- <min_alerts_reset_counter>
- <field1_recovery>
- <field4>
- <field5>
- <field6>
- <field7>
- <field8>
- <field9>
- <field10>
- <field11>
- <field12>
- <field13>
- <field14>
- <field15>
- <field4_recovery>
- <field5_recovery>
- <field6_recovery>
- <field7_recovery>
- <field8_recovery>
- <field9_recovery>
- <field10_recovery>
- <field11_recovery>
- <field12_recovery>
- <field13_recovery>
- <field14_recovery>
- <field15_recovery>

Examples

Example 1 (condition: regexp =~ /pp/, action: Mail to XXX, max_alert: 10, min_alert: 0, priority: WARNING, group: databases):

```
http://127.0.0.1/pandora_console/include/api.php?op=set&op2=create_alert_templat
e&id=example&other=regex|template%20based%20in%20regexp|1||||pp|1||||10|0|||||||
||||||3|8&other_mode=url_encode_separator_|&apipass=1234&user=admin&pass=pandora
```

Example 2 (condition: value is not between 5 and 10, max_value: 10.00, min_value: 5.00, time_from: 00:00:00, time_to: 15:00:00, priority: CRITICAL, group: Servers):

```
http://127.0.0.1/pandora_console/include/api.php?op=set&op2=create_alert_templat
```

```
e&id=template_min_max&other=max_min|template%20based%20in%20range|1||||||10|5|||
|00:00:00|15:00:00||||||||||4|2&other_mode=url_encode_separator_|&apipass=1234&
user=admin&pass=pandora
```

set update_alert_template

It updates the template alert.

Call Syntax:

- op=set (required)
- op2=update_alert_template (required)
- id=<id_template> (required). It should be a template id.
- other=<serialized parameters> (required). They are template configuration and data, serialized in the following order:
  - <template_name>
  - <type [regex|max_min|max|min|equal|not_equal|warning|critical|onchange|unknown|always]>
  - <description>
  - <id_alert_action>
  - <field1>
  - <field2>
  - <field3>
  - <value>
  - <matches_value>
  - <max_value>
  - <min_value>
  - <time_threshold>
  - <max_alerts>
  - <min_alerts>
  - <time_from>
  - <time_to>
  - <monday>
  - <tuesday>
  - <wednesday>
  - <thursday>
  - <friday>
  - <saturday>
  - <sunday>
  - <recovery_notify>
  - <field2_recovery>
  - <field3_recovery>
  - <priority>
  - <id_group>
  - <special_day>
  - <min_alerts_reset_counter>
  - <field1_recovery>
  - <field4>
  - <field5>
  - <field6>
  - <field7>

  - <field8>
  - <field9>
  - <field10>
  - <field11>
  - <field12>
  - <field13>
  - <field14>
  - <field15>
  - <field4_recovery>
  - <field5_recovery>
  - <field6_recovery>
  - <field7_recovery>
  - <field8_recovery>
  - <field9_recovery>
  - <field10_recovery>
  - <field11_recovery>
  - <field12_recovery>
  - <field13_recovery>
  - <field14_recovery>
  - <field15_recovery>

Examples

```
http://127.0.0.1/pandora_console/include/api.php?op=set&op2=update_alert_template&id=18&other=example_template_with_changed_name|onchange|changing%20from%20min_max%20to%20onchange||||||1||||5|1|||1|1|0|1|1|0|0|1|field%20recovery%20example%202|field%20recovery%20example%203|1|8&other_mode=url_encode_separator_|&apipass=1234&user=admin&pass=pandora
```

set delete_alert_template

It deletes a alert template and deletes all the alerts it defines.

Call Syntax:

- op=set (required)
- op2=delete_alert_template (required)
- id=<id_template> (required). It should be a template id.

Examples

```
http://127.0.0.1/pandora_console/include/api.php?op=set&op2=delete_alert_template&id=38&apipass=1234&user=admin&pass=pandora
```

set delete_module_template

It deletes a module template.

Call Syntax:

- op=set (required)
- op2=delete_module_template (required)
- id=<id_alert_template_module> (required). It should be an alert_template_module id.

Examples

```
http://127.0.0.1/pandora_console/include/api.php?op=set&op2=delete_module_templa
te&id=38&apipass=1234&user=admin&pass=pandora
```

set delete_module_template_by_names

It deletes a module template.

Call Syntax:

- op=set (required)
- op2=delete_module_template_by_names (required)
- id=<agent name> (required)
- id2=<alert template name> (required)
- other=<serialized parameter> (required). They are the following in this order:
    - <module name> (required)
    - <use_agent_alias> (Values 0 and 1 supported)

Examples

```
http://127.0.0.1/pandora_console/include/api.php?op=set&op2=delete_module_templa
te_by_names&id=sample-
agent&id2=test&other=memfree&apipass=1234&user=admin&pass=pandora&other_mode=url
_encode_separator_|
```

set stop_downtime

It stops a downtime.

Call Syntax:

- op=set (required)
- op2=stop_downtime (required)
- id=<id_downtime> (required). It should be an id downtime.

Examples

```

http://127.0.0.1/pandora_console/include/api.php?op=set&op2=stop_downtime&id=1&a
pipass=1234&user=admin&pass=pandora
```

set new_user

It creates a new user in Pandora FMS.

Call Syntax:

- op=set (required).
- op2=new_user (required).
- id=< identification_user > (required).
- other=< serialized_parameters > (all and each one are required) they are user configuration and data, serialized in the following order:
  1. < full_name > (replace spaces with %20)
  2. < first_name >
  3. < surname >
  4. < middle_name >
  5. < password >
  6. < e_mail >
  7. < phone_number >
  8. < language >
  9. < comments >
  10. < time_autorefresh >
  11. < default_event_filter >
  12. < console_section >
  13. < session_time >

If you lack any of the above thirteen fields simply type the separator (see call syntax) but in any case you must always place 13 separators.

Example (see call syntax):

```
.../include/api.php?op=set&op2=new_user&id=id_nu&other=John%20J.%20Doe|John|Doe|Jay|1234|johndoe@example.com|5555555|en||30|||-1|&other_mode=url_encode_separator_|&apipass=1234&user=admin&pass=pandora
```

set update_user

It updates a user selected by the id in the id parameter.

Call Syntax:

- op=set (required)
- op2=update_user (required)
- id=<user_name> (required). It should be a user name.
- other=<serialized parameters> (required). They are module configuration and data, serialized in the following order:
  - <fullname>

- <firstname>
- <lastname>
- <middlename>
- <password>
- <email>
- <phone>
- <languages>
- <comments>
- <is_admin>
- <block_size>
- <flash_chart>

Examples

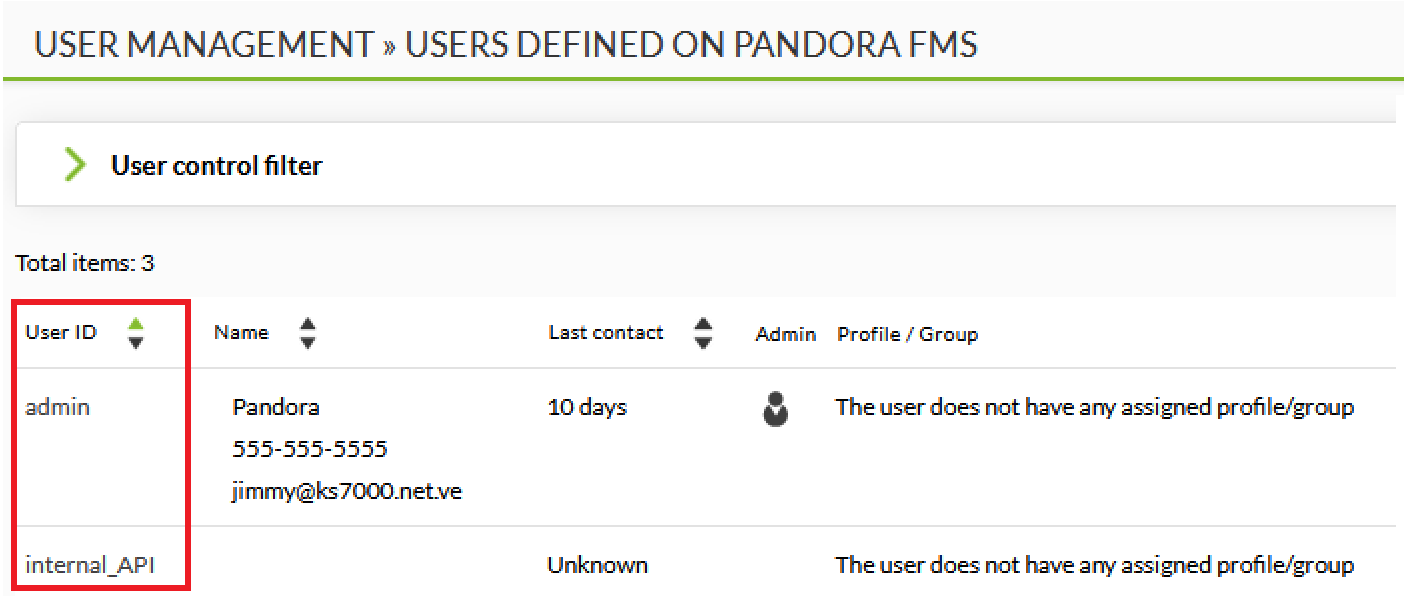
```
http://127.0.0.1/pandora_console/include/api.php?op=set&op2=update_user&id=example_user_name&other=example_fullname||example_lastname||example_new_passwd|example_email||example_language|example%20comment|1|30|&other_mode=url_encode_separator_|&apipass=1234&user=admin&pass=pandora
```

set delete_user

Deletes a user by means of its identifier.

Call syntax:

- op=set (required).
- op2=delete_user (required).
- id=< id_user > (required) the identifier of the user to delete (User ID).



USER MANAGEMENT » USERS DEFINED ON PANDORA FMS

User control filter

Total items: 3

User ID	Name	Last contact	Admin	Profile / Group
admin	Pandora 555-555-5555 jimmy@ks7000.net.ve	10 days	●	The user does not have any assigned profile/group
internal_API		Unknown		The user does not have any assigned profile/group

Example (see call syntax):

```
../include/api.php?op=set&op2=delete_user&id=md&apipass=1234&user=admin&pass=pan
```

```
dora
```

set delete_user_permissions

This feature is in Command Center (Metaconsole).

It deletes user permissions.

Call syntax:

- op=set(requiered)
- Op2=delete_user_permission(requiered)
- Return_type=csv|json(requiered)
- Other=id profile(requiered)

Example

It deletes user permissions. It can return Json or CSV, this return is remarked through the URL.

```
http://localhost/pandora_console/include/api.php?op=set&op2=delete_user_permission&return_type=json&other=2&other_mode=url_encode_separator_|&apipass=1234&user=admin&pass=pandora
```

set add_permission_user_group

This feature is in Command Center (Metaconsole).

It adds permissions to a user group.

Call syntax:

- Op=set(requiered)
- Op2=add_permission_user_to_group(requiered)
- Return_type=csv|json(requiered)
- Other=user id(requiered)|group_id(requiered)|profile id(requiered)|No hierarchy(Optional)| profile id(Optional)

Examples

It can return Json or CSV, this return is remarked through the URL.

Be careful, no_hierarchy may be empty. If that is the case,
it takes value 0.

Be careful, permission id (id_up) in the table can be used if an existing permission must be modified.

```
http://localhost/pandora_console/include/api.php?op=set&op2=add_permission_user_
to_group&return_type=json&other=admin|0|1|1|20&other_mode=url_encode_separator_|
&apipass=1234&user=admin&pass=pandora
```

set enable_disable_user

It enables a disabled user.

Call syntax:

- op=set (required)
- op2=enable_disable_user (required)
- id=<user_name> (required). It should be a username.

Examples

Example 1 (Disable user 'example_name')

```
http://127.0.0.1/pandora_console/include/api.php?op=set&op2=enable_disable_user&
id=example_name&other=0&other_mode=url_encode_separator_|&apipass=1234&user=admi
n&pass=pandora
```

Example 2 (Enable user 'example_name')

```
http://127.0.0.1/pandora_console/include/api.php?op=set&op2=enable_disable_user&
id=example_name&other=1&other_mode=url_encode_separator_|&apipass=1234&user=admi
n&pass=pandora
```

set meta_synch_user

This feature is in Command Center (Metaconsole).

It syncronizes Command Center (Metaconsole) users to the node

Call syntax:

- op=set (required)
- op2=meta_synch_user (required)
- other=<serialized parameters> (opcional):
  - <user_name_1,user_name_2,user_name_3..> (Usernames separated by comma and without spaces at the beginning or the end, required)

  - <server_name> (required)
  - <profile_mode(1-0)> (optional)
  - <group_name> (optional)
  - <profile_1,profile_2,profile_3> (Profiles separated by comma and without spaces at the beginning or at the end, optional)
  - <create_groups(1-0)> (optional)

Example

```
http://localhost/pandora_console/include/api.php?op=set&op2=meta_synch_user&apip
ass=1234&user=admin&pass=pandora&other=name1,name2|nodo1|0|||&other_mode=url_enc
ode_separator_|
```

set create_group

It creates a group.

Call syntax:

- op=set (required)
- op2=create_group (required)
- id=<group_name> (required). It should be a group name.
- other=<serialized_parameters> (required). They are the following in this order:
  - <icon name>
  - <parent group id> (optional)
  - <description> (optional)
  - <propagate acl> (optional)
  - <disable alerts> (optional)
  - <custom id> (optional)
  - <contact info> (optional)
  - <other info> (optional)
  - <Maximum number of agents in the group> (required, zero means no limit)
  - <Create Agent group with password> (optional)

Examples

Example 1 (with parent group: Servers)

```
http://127.0.0.1/pandora_console/include/api.php?op=set&op2=create_group&id=exam
ple_group_name&other=applications|2&other_mode=url_encode_separator_|&apipass=12
34&user=admin&pass=pandora
```

Example 2 (without parent group)

```
http://127.0.0.1/pandora_console/include/api.php?op=set&op2=create_group&id=exam
ple_group_name2&other=computer|&other_mode=url_encode_separator_|&apipass=1234&u
ser=admin&pass=pandora
```

Example 3 (create a group of agents with password 1234, and with a limit of 3 Agents in the group)

```
http://127.0.0.1/pandora_console/include/api.php?op=set&op2=create_group&id=example_group_name&other=applications|2||||||3|1234&other_mode=url_encode_separator_|&apipass=1234&user=admin&pass=pandora
```

set update_group

It updates a group with past data as parameters.

Call syntax:

- op=set (required)
- op2=update_group (required)
- id=<group_id> (required). It should be a group id
- other=<serialized_parameters> (required). They are the following in this order:
  - <group name>
  - <icon name>
  - <parent group id>
  - <description>
  - <propagate acl>
  - <disable alerts>
  - <custom id>
  - <contact info>
  - <other info>

Example

```
http://127.0.0.1/pandora_console/include/api.php?op=set&op2=update_group&id=example_group_id&other=New%20Name|application|2|new%20description|1|0|custom%20id||&other_mode=url_encode_separator_|&apipass=1234&user=admin&pass=pandora
```

set delete_group

It deletes a group.

Call syntax:

- op=set (required)
- op2=delete_group (required)
- id=<group_id> (required) it should be a group id

Example

```
http://127.0.0.1/pandora_console/include/api.php?op=set&op2=delete_group&id=303&apipass=1234&user=admin&pass=pandora
```

set add_user_profile

It adds a profile into user.

Call syntax:

- op=set (required)
- op2=add_user_profile (required)
- id=<user_name> (required). It should be a user name.
- other=<serialized parameters> (required). They are group and profile configuration and data, serialized in the following order:
  - <group>
  - <profile>

Examples

```
http://127.0.0.1/pandora_console/include/api.php?op=set&op2=add_user_profile&id=md&other=12|4&other_mode=url_encode_separator_|&apipass=1234&user=admin&pass=pandora
```

set delete_user_profile

It deletes a profile from a user.

Call syntax:

- op=set (required)
- op2=delete_user_profile (required)
- id=<user_name> (required). It should be a user name.
- other=<serialized parameters> (required). They are the group configuration, data and profile, serialized in the following order:
  - <group>
  - <profile>

Examples

```
http://127.0.0.1/pandora_console/include/api.php?op=set&op2=delete_user_profile&id=md&other=12|4&other_mode=url_encode_separator_|&apipass=1234&user=admin&pass=pandora
```

set new_incident

It creates a new incident.

Call syntax:

- op=set (required)
- op2=new_incident (required)
- other=<serialized parameters> (required). They are incident configuration and data, serialized in the following order:
    - <title>
    - <description>
    - <origin>
    - <priority>
    - <status>
    - <group>

Examples

```
http://127.0.0.1/pandora_console/include/api.php?op=set&op2=new_incident&other=titulo|descripcion%20texto|Logfiles|2|10|12&other_mode=url_encode_separator_|&apipass=1234&user=admin&pass=pandora
```

set new_note_incident

It adds a note within an incident.

Call syntax:

- op=set (required)
- op2=new_note_incident (required)
- id=<id_incident> (required). It is the incident id.
- id2=<user_name> (required). Username.
- other=<note> (required). It is the note codified in url encode.

Examples

```
http://127.0.0.1/pandora_console/include/api.php?op=set&op2=new_note_incident&id=5&id2=miguel&other=una%20nota%20para%20la%20incidencia&apipass=1234&user=admin&pass=pandora
```

set validate_all_alerts

It validates all alerts.

Call syntax:

- op=set (required)
- op2=validate_all_alerts (required)

Examples

```
http://127.0.0.1/pandora_console/include/api.php?op=set&op2=validate_all_alerts&
apipass=1234&user=admin&pass=pandora
```

set validate_all_policy_alerts

It validates the alerts created from a policy.

Call syntax:

- op=set (required)
- op2=validate_all_policy_alerts (required)

Examples

```
http://127.0.0.1/pandora_console/include/api.php?op=set&op2=validate_all_policy_
alerts&apipass=1234&user=admin&pass=pandora
```

set event_validate_filter

It validates all events that pass the past filter as parameters.

This feature is in Command Center (Metaconsole).

Call syntax:

- op=set (required)
- op2=event_validate_filter (required)
- other_mode=url_encode_separator_|(optional)
- other=<serialized_parameters> (optional). They are the following in this order:
    - <separator>
    - <criticity> From 0 to 4
    - <agent name>
    - <module name>
    - <alert template name>
    - <user>
    - < numeric interval minimum level> en unix timestamp
    - < numeric interval maximum level> en unix timestamp
    - <use_agent_name> (Values 0 and 1 supported)

Examples

```

http://127.0.0.1/pandora_console/include/api.php?op=set&op2=event_validate_filte
r&other_mode=url_encode_separator_|&other=;|2&apipass=1234&user=admin&pass=pando
ra
```

set event_validate_filter_pro

It is similar to previous call.

This feature is in Command Center (Metaconsole).

Call syntax:

- op=set (required))
- op2=event_validate_filter_pro (required)
- other_mode=url_encode_separator_| (optional)
- other=<serialized parameters> (optional). They are the following in this order:
    - <separator>
    - <criticity> From 0 to 4
    - <id agent>
    - <id module>
    - <id agent module alert>
    - <user>
    - <numeric interval minimum level> in unix timestamp
    - <numeric interval maximum level> in unix timestamp

Examples

```
http://127.0.0.1/pandora_console/include/api.php?op=set&op2=event_validate_filter_pro&other_mode=url_encode_separator_|&other=;|2&apipass=1234&user=admin&pass=pandora
```

set validate_event_by_id

Validates an event given its id.

Call syntax:

- op=set (required)
- op2=validate_event_by_id (required)
- id=<event_id> (required) event id.

Examples

```
http://localhost/pandora_console/include/api.php?op=set&op2=validate_event_by_id&id=23&apipass=1234&user=admin&pass=pandora
```

set new_alert_template

It applies a new alert from a template and module which has gone through id agent and module name.

Call syntax:

- op=set (ob)
- op2=new_alert_template (required)
- id=<agent name> (required)
- id2=<alert template name> (required)
- other_mode=url_encode_separator_| (optional)
- other=<serialized parameter> (required). They are the following in this order:
    - <module name> (required)
    - <use_agent_alias> (Values 0 and 1 supported)

Examples

http://127.0.0.1/pandora_console/include/api.php?op=set&op2=new_alert_template&id=miguel-portatil&id2=test&other_mode=url_encode_separator_|&other=memfree&apipass=1234&user=admin&pass=pandora

set alert_actions

It adds actions within an alert.

Call syntax:

- op=set (required)
- op2=alert_actions (required)
- id=<agent name> (required)
- id2=<alert template name> (required)
- other_mode=url_encode_separator_| (required)
- other=<serialized parameters> (required). They are the following in this order:
    - <module name> (required)
    - <action name> (required)
    - <fires min > (optional)
    - <fires max > (optional)
    - <use_agent_alias> (Values 0 and 1 supported)

Examples

```
http://127.0.0.1/pandora_console/include/api.php?op=set&op2=alert_actions&id=mig
uel-
portatil&id2=test&other_mode=url_encode_separator_|&other=memfree|test&apipass=1
234&user=admin&pass=pandora
```

```
http://127.0.0.1/pandora_console/include/api.php?op=set&op2=alert_actions&id=mig
uel-
```

```
portatil&id2=test&other_mode=url_encode_separator_|&other=memfree|test|1|3&apipass=1234&user=admin&pass=pandora
```

set alert_commands

It adds commands within an alert.

Call syntax:

- op=set (required)
- op2=alert_commands (required)
- id=<name of the command> (required)
- other_mode=url_encode_separator_| (optional)
- other=<serialized parameters> (required). They are the following in this order:
  - <command> (required)
  - <id_group> (required)
  - <description > (required)
  - <internal > (optional)
  - <field_description_1><field_value_1><field_description_2><field_value_2>...<field_description_n><field_value_n>

Examples

```
http://localhost/pandora_console/include/api.php?op=set&op2=alert_commands&id=PRUEBA1&other=command|0|Desc|1|des1|val1|des2|val2|des3|val3||val4|des5&other_mode=url_encode_separator_|&apipass=1234&user=admin&pass=pandora
```

set new_module

It creates a new module.

Call Syntax:

- op=set (required)
- op2=new_module (required)
- id=<agent_name> (required)
- id2=<new module name> (required)
- other_mode=url_encode_separator_| (optional)
- other=<serialized parameters> (required). They are the following in this order:
  - <network module kind > (required)
  - <action name> (required)
  - <ip o url > (required)
  - <port > (optional)
  - <description > (optional)
  - <min > (optional)
  - <max > (optional)
  - <post process > (optional)
  - <module interval > (optional)

  - <min warning > (optional)
  - <max warning > (optional)
  - <min critical > (optional)
  - <max critical > (optional)
  - <history data > (optional)
  - <enable_unknown_events> (only in version 5)
  - <use_agent_alias> (Values 0 and 1 supported)

Examples

```
http://127.0.0.1/pandora_console/include/api.php?op=set&op2=new_module&id=miguel
-
portatil&id2=juanito&other_mode=url_encode_separator_|&other=remote_tcp_string|l
ocalhost|33|descripcion%20larga&apipass=1234&user=admin&pass=pandora
```

set delete_module

It deletes a module.

From version 768 onwards, if the module to be deleted (e.g. *Host Alive*) is the *parent* of another module ("Cascading Protection Services" functionality), the *child* modules will also be deleted.

Call syntax:

- op=set (required).
- op2=delete_module (obligatorio)
- id=< agent name > (obligatorio).
- id2=< module name > (obligatorio).

Example (see call syntax):

```
.../include/api.php?op=set&op2=delete_module&id=example&id2=example2&apipass=123
4&user=admin&pass=pandora&other_mode=url_encode_separator_|
```

simulate parameter

It is absolutely essential to use

`other_mode=url_encode_separator_< separator >`

with this parameter!

If you want to confirm if the agent exists and its module also exists, before executing the final deletion you can use the `simulate` parameter. If your test call is incorrect the PFMS API 1.0 will return the following message: `Parameter error.`.

Example (see call syntax):

```
.../include/api.php?op=set&op2=delete_module&id=agent_name_example&id2=module_name_example&other=simulate&other_mode=url_encode_separator_|&apipass=1234&user=admin&pass=pandora
```

set enable_alert

It enables an agent alert.

Call syntax

- op=set (required)
- op2=enable_alert
- id=<Agent name> (required)
- id2=<Module name> (required)
- other:Alert template name (p.e: Warning event) (required)

Examples

```
http://localhost/pandora_console/include/api.php?op=set&op2=enable_alert&id=garfio&id2=Status&other=Warning%20condition&apipass=1234&user=admin&pass=pandora
```

set enable_alert_alias

It enables agent alert by alias.

Call syntax:

- op=set (required)
- op2=enable_alert_alias
- id=<Agent alias> (required)
- id2=<Module name> (required)
- other:Alert template name (p.e: Warning event) (required)

Examples

```
http://localhost/pandora_console/include/api.php?op=set&op2=enable_alert_alias&id=nova&id2=CPU%20Load&other=critical%20condition&apipass=1234&user=admin&pass=pandora
```

set disable_alert

It disables an agent alert.

Call syntax:

- op=set (required)
- op2=disable_alert
- id=<Agent name> (required)
- id2=<Module name> (required)
- other:Alert template name (p.e: Warning event) (required)

Examples

```
http://localhost/pandora_console/include/api.php?op=set&op2=disable_alert&id=gar
fio&id2=Status&other=Warning%20condition&apipass=1234&user=admin&pass=pandora
```

set disable_alert_alias

It disables an agent alert.

Call syntax:

- op=set (required)
- op2=disable_alert_alias
- id=<Agent alias> (required)
- id2=<Module name> (required)
- other:Alert template name (p.e: Warning event) (required)

Examples

```
http://localhost/pandora_console/include/api.php?op=set&op2=disable_alert_alias&
id=nova&id2=CPU%20Load&other=critical%20condition&apipass=1234&user=admin&pass=p
andora
```

set enable_module_alerts

Same as enable_alert api call.

Call syntax:

- op=set (required)
- op2=enable_module_alerts
- id=<Name of the agent> (required)
- id2=<Name of the module> (required)
- other (optional)=<use_agent_alias> (Values 0 and 1 supported)

Examples

```
http://localhost/pandora_console/include/api.php?op=set&op2=enable_module_alerts
&id=garfio&id2=Status&apipass=1234&user=admin&pass=pandora
```

set disable_module_alerts

Same as api disable_alert.

Call syntax:

- op=set (required)
- op2=disable_module_alerts
- id=<Name of the agent> (required)
- id2=<Name of the module> (required)
- other (optional)=<use_agent_alias> (Values 0 and 1 supported)

Examples

```
http://localhost/pandora_console/include/api.php?op=set&op2=disable_module_alert
s&id=garfio&id2=Status&apipass=1234&user=admin&pass=pandora
```

set enable_module

It enables the module.

Call syntax

- op=set (required)
- op2=enable_module
- id=<Agent name> (required)
- id2=<Module name> (required)
- other (optional)=<use_agent_alias> (Values 0 and 1 supported)

Examples

```
http://localhost/pandora_console/include/api.php?op=set&op2=enable_module&id=gar
fio&id2=Status&apipass=1234&user=admin&pass=pandora
```

set disable_module

It disables the module.

Call syntax:

- op=set (required)

- op2=disable_module
- id=<Agent name> (required)
- id2=<Module name> (required)
- other (optional)=<use_agent_alias> (Values 0 and 1 supported)

Examples

```
http://localhost/pandora_console/include/api.php?op=set&op2=disable_module&id=ga
rfio&id2=Status&apipass=1234&user=admin&pass=pandora
```

set create_network_module_from_component

It creates a new network module from a component.

Call syntax:

- op=set (required)
- op2=create_network_module_from_component (required)
- id=<Agent name> (required)
- id2=<Component name> (required)
- other (optional)=<use_agent_alias> (Values 0 and 1 supported)

Examples

```
http://localhost/pandora_console/include/api.php?op=set&op2=create_network_modul
e_from_component&id=garfio&id2=OS%20Total%20process&apipass=1234&user=admin&pass
=pandora
```

set module_data

It adds a module value. This function generates an XML with the data that will be sent to the server, which will be the one to update the database.

This API call just controls the XML file generation, displaying in a message the status of that process.

Call syntax:

- op=set (required)
- op2=module_data (required)
- id=<id module agent> (required)
- other:module data and timestamp serialized.
    - dato: data which must belong to any Pandora FMS data type.
    - tiempo: it could be a specified timestamp of the string now.

Example

```
http://127.0.0.1/pandora_console/include/api.php?op=set&op2=module_data&id=14&ot
her_mode=url_encode_separator_|&other=123|now&apipass=1234&user=admin&pass=pando
ra
```

The successful generation of the XML file doesn't
guarantee that the data has been stored in the database.

set new_module_group

It creates a new module group.

Call syntax:

- op=set (required)
- op2=new_module_group (required)
- id=<name of the module group> (required)

Examples

```
http://localhost/pandora_console/include/api.php?op=set&op2=new_module_group&id=
Module_group_name&apipass=1234&user=admin&pass=pandora
```

set module_group_synch

This feature is in Command Center (Metaconsole).

It syncronizes a module group from the Command Center (Metaconsole).

Call syntax:

- op=set (required)
- op2=module_group_synch (required)
- other=<serialized parameters> (required). In this case, the name of the server(s)that are required to sync with the meta.
  - <server_name_1><server_name_2>...<server_name_n> (required)

Examples

```
http://localhost/pandora_console/include/api.php?op=set&op2=module_group_synch&o
ther=server_name1|server_name2|server_name3&other_mode=url_encode_separator_|&ap
ipass=1234&user=admin&pass=pandora
```

set add_module_in_conf

It adds the configuration to a local module.

Call syntax:

- op=set (required)
- op2=add_module_in_conf (required)
- id=<agent id> (required)
- id2=<module name> (required)
- other:The module data that will be placed in the conf file encoded in base64 (required) (required)

Examples

```
http://127.0.0.1/pandora_console/include/api.php?op=set&op2=add_module_in_conf&a
pipass=1234&user=admin&pass=pandora&id=9043&id2=example_name&other=bW9kdWxlX2JlZ
2luCm1vZHVsZV9uYW1lIGV4YW1wbGVfbmFtZQptb2R1bGVfdHlwZSBnZW5lcmljX2RhdGEKbW9kdWxlX
2V4ZWMgZWNobyAxOwptb2R1bGVfZW5k
```

It will return '0' when it is successful, '-1' when there is an error and '-2' if it already exists.

set delete_module_in_conf

It deletes a local module configuration.

Call syntax:

- op=set (required)
- op2=add_module_in_conf (required)
- id=<agent id> (required)
- id2=<module name> (required)

Examples

```
http://127.0.0.1/pandora_console/include/api.php?op=set&op2=add_module_in_conf&u
ser=admin&pass=pandora&id=9043&id2=example_nameInsert non-formatted text here
```

It will return '0' when it is successful or '-1' when there is an error

set update_module_in_conf

It updates a local module configuration.

Call syntax:

- op=set (required)
- op2=update_module_in_conf (required)
- id=<agent id> (required)
- id2=<module name> (required)
- other:The new module data that will be placed in the conf file encoded in base64 (required)

Examples

```
http://127.0.0.1/pandora_console/include/api.php?op=set&op2=update_module_in_conf&apipass=1234&user=admin&pass=pandora&id=9043&id2=example_name&other=bW9kdWxlX2JlZ2luCm1vZHVsZV9uYW1lIGV4YW1wbGVfbmFtZQptb2R1bGVfdHlwZSBnZW5lcmljX2RhdGEKbW9kdWxlX2V4ZWMgZWNobyAxOwptb2R1bGVfZW5k
```

It will return '1' when there are no changes, '0' when it is successful, '-1' when there is an error and '-2' if does not exist.

set create_event

It creates a new event in Pandora FMS.

This feature also works in Command Center (Metaconsole).

Call syntax:

- `op=set` (required).
- `op2=create_event` (required).
- `other=<` serialized_parameters > (required), they are the configuration and event data, serialized in the following order:
    - < event_text > (required, string type).
    - < id_group > (required, string type).
    - < id_agent> (required, numeric type).
    - < status > `0` *New*, 1 *Validated*, 2 *In process*.
    - < id_user >
    - < event_type > Any of the following values:
        - `unknown`
        - `alert_fired`
        - `alert_recovered`
        - `alert_ceased`
        - `alert_manual_validation`
        - `system`
        - `error`
        - `new_agent`
        - `configuration_change`
        - `going_unknown`
        - `going_down_critical`
        - `going_down_warning`

        - going_up_normal
    - < severity > Any of the following values:
        - 0 *Maintenance*.
        - 1 *Informative*.
        - 2 *Normal*.
        - 3 *Warning*.
        - 4 *Critical*.
        - 5 *Minor*.
        - 6 *Major*.
    - < id_agent_module > (numeric type).
    - < id_alert_am> (ID Alert Module linked to event, numeric type).
    - < critical_instructions > (string type).
    - < warning_instructions > (string type).
    - < unknown_instructions > (string type).
    - < comment > (string type).
    - < owner_user_name > (string type).
    - < event_source > (string type).
    - < tags > (string type).
    - < custom_data> Custom data should be a base64 encoded JSON document.
    - < server_id > (*only for Command Center (Metaconsole)*) The id of the child node.
    - <id_extra> (alphanumeric type).

Example (see call syntax):

The event to be created has the following custom JSON data:

```
{"Answer to the Ultimate Question of Life, the Universe, and Everything": 42}
```

The above is encoded in base64 and inserted in the call:

```
../include/api.php?op=set&op2=create_event&other_mode=url_encode_separator_|&apipass=1234&user=admin&pass=pandora&other=Event_name|0|1|0|admin|alert_fired|4|1|||||comment|admin|||tags|eyJBbnN3ZXIgdG8gdGhlIFVsdGltYXRlIFF1ZXN0aW9uIG9mIExpZmUsIHRoZSBVbml2ZXJzZSwgYW5kIEV2ZXJ5dGhpbmciOiA0Mn0=||12
```

set add_event_comment

It adds an event comment.

This feature also works in Command Center (Metaconsole).

Call syntax:

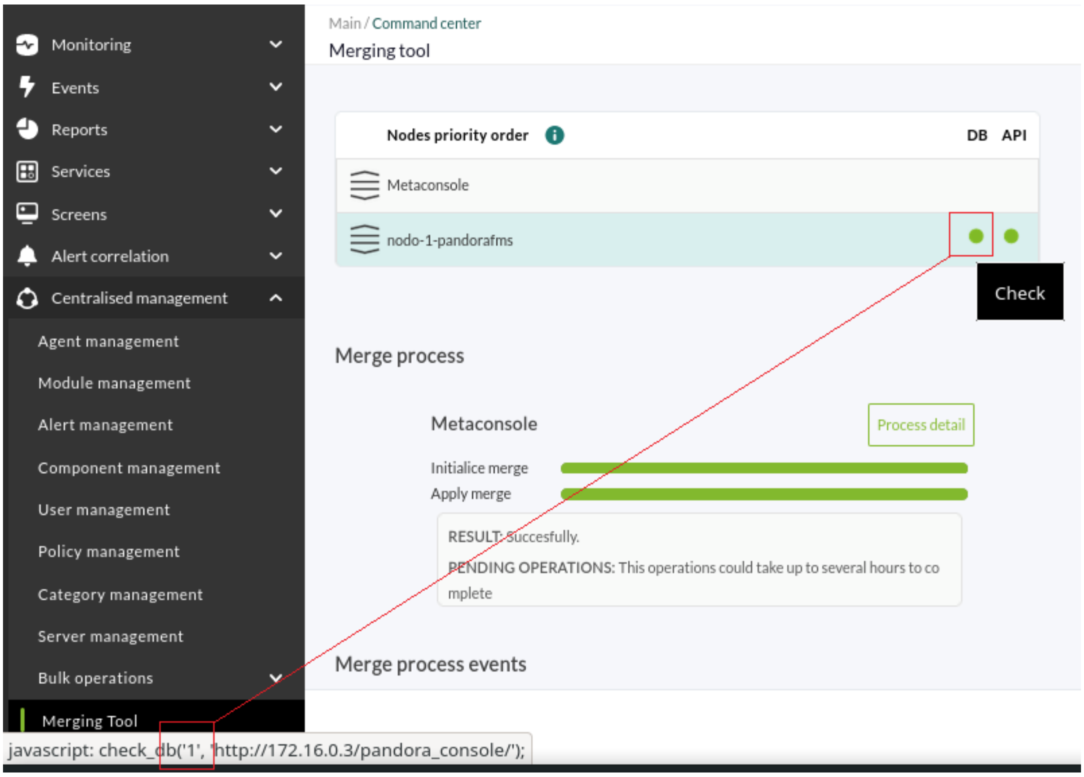
- op=set (required).
- op2=add_event_comment (required).
- id=< id_event > (required, numeric type).
- other=< serialized_parameters > (required) are the comment to be added and, if used in Command

Center (Metaconsole), the numerical identifier of the node, data serialized as follows:
  - < comment > (required, string type).
  - < separator > (*optional, for Command Center -Metaconsole- use*).
  - < id_node > (*optional, for Command Center -Metaconsole- use*).
- `url_encode_separator_`< separator >:See call syntax.

Command Center (Metaconsole) usage:

- NG 762 version an earlier: After the comment and the separator you must enter the parameter `true`.
- NG 766 version and later: After the comment and the separator you must enter the numerical identifier of the node.

This node identifier can be obtained by accessing the Command Center:



RESULT: Succesfully.
PENDING OPERATIONS: This operations could take up to several hours to co mplete

javascript: check_db('1', 'http://172.16.0.3/pandora_console/');

Examples (see call syntax)

- Node (event id is 7 and comment is `comment`):

```
../include/api.php?op=set&op2=add_event_comment&id=7&other=comment&other_mode=ur
l_encode_separator_|&apipass=1234&user=admin&pass=pandora
```

- Command Center (Metaconsole) (NG 766 and later, the event identifier is 1, the comment is `comment` and the node identifier is 3):

```
../include/api.php?op=set&op2=add_event_comment&id=1&other=comment|3&other_mode=
url_encode_separator_|&apipass=1234&user=admin&pass=pandora
```

set update_event

Updates events through API.

This feature can be also used in the Command Center (Metaconsole).

Call syntax:

- op=set (required)
- op2=event (required)
- id=event_id (required)
- other=<serialized_parameters> (required). Event fields to be updated, formatted as *field,value* and separated by the chosen url_encode_separator (e.g. |).

= Available fields =

- estado: Numeric identifier of the event status. It can be:
  - 0 (New)
  - 1 (Validated)
  - 2 (In process)
- evento: Event name.
- event_type: Event type. It can be:
  - going_unknown
  - unknown
  - alert_fired
  - alert_recovered
  - alert_ceased
  - alert_manual_validation
  - recon_host_detected
  - system
  - error
  - new_agent
  - going_up_warning
  - going_up_critical
  - going_down_warning
  - going_down_normal
  - going_down_critical
  - going_up_normal
  - configuration_change
- criticity: Numeric identifier of the event's criticity. It can be:
  - 0 (Maintenance)
  - 1 (Informational)
  - 2 (Normal)
  - 3 (Minor)
  - 4 (Warning)
  - 5 (Major)
  - 6 (Critical)
  - 20 (Warning o Critical)

    - 21 (Distinto a Normal)
    - 34 (Critical o normal)
- tags: Tags associated to the event.
- source: Source of the event's data.
- id_extra: When using this feature, older events with the same Extra ID as the new one will be automatically validated.
- critical_instructions: Instructions for operators on the actions to be done when a module goes to Critical status. Only shown in the event if module is on Critical status.
- warning_instructions: Instructions for operators on the actions to be done when a module goes to Warning status. Only shown in the event if module is on Warning status.
- unknown_instructions: Instructions for operators on the actions to be done when a module goes to Unknown status. Only shown in the event if module is on Unknown status.
- owner_user: User assigned to the event.
- custom_data: It allows to add custom information to the event. It must be formated as a base64 encoded json (e.g.

```
{"field1":"value1","field2":"value2"}
```

). See example below.

- module_status: Numeric identifier for the status of the module that triggered the event. It can be:
    - 0 (Normal)
    - 1 (Critical)
    - 2 (Warning)
    - 3 (Unknown)
    - 4 (Not init)

Examples

- Node:

```
http://192.168.80.190/pandora_console/include/api.php?op=set&op2=event&apipass=pandora&user=admin&pass=pandora&id=175&other_mode=url_encode_separator_|&other=estado,2|evento,Updated
event|custom_data,eyJmaWVsZDEiOiJ2YWx1ZTEiLCJmaWVsZDIiOiJ2YWx1ZTIifQo=
```

- Command Center (Metaconsole):

```
http://192.168.80.35/pandora_console/include/api.php?op=set&op2=event&apipass=1234&user=admin&pass=pandora&id=315132&other_mode=url_encode_separator_|&other=estado,0|owner_user,operator|evento,Updated event
```

set create_netflow_filter

(>=5.0)

It creates a new netflow filter.

Call syntax:

- op=set (required)
- op2=create_netflow_filter (required)
- other=<serialized parameters> (required). It filters data in this order:
    - <filter_name> (required)
    - <group_id> (required)
    - <filter> (required)
    - <aggregate_by> (Possible values: dstip,dstport,none,proto,srcip,srcport) (required)
    - <output_format> (Possible values: kilobytes,kilobytespersecond,megabytes,megabytespersecond) (required)

Examples

```
http://127.0.0.1/pandora/include/api.php?op=set&op2=create_netflow_filter&apipass=1234&user=admin&pass=pandora&other=Filter%20name|9|host%20192.168.50.3%20OR%20host%20192.168.50.4%20or%20HOST%20192.168.50.6|dstport|kilobytes&other_mode=url_encode_separator_|
```

set create_custom_field

It creates a new custom field for agents. *Custom fields* are useful in custom field macros for remote monitoring.

Call syntax:

- `op=set` (required).
- `op2=create_custom_field` (required).
- `other=`< serialized parameters > (required) Parameters to configure the custom field.
    - < custom_field_name > (required, *string* type).
    - < show_in_agent_operation_view > (required) `0` will not be displayed in the agent operation view, 1 will be displayed, (as long as it has a saved value).
    - < password_field > (required) `0` normal custom field, 1 its content will be hidden with asterisks in the Web Console.

If the execution of the command is successful, it will return a numeric identifier corresponding to the custom field created.

Example (see call syntax) :

```
../include/api.php?op=set&op2=create_custom_field&other=mycustomfield|0|0&other_mode=url_encode_separator_|&apipass=1234&user=admin&pass=pandora
```

Make sure that the user used has the appropriate rights to create custom fields. In the example the credentials of a *superadmin* are used, if you use the default user `internal_API` you will not be able to create such custom fields.

set create_tag

> = 5.0

It creates a new tag.

Call syntax:

- op=set (required)
- op2=create_tag (required)
- other=<serialized parameters> (required). Parameters to configure the tag.
  - <name> Tag name (required)
  - <description> Tag description
  - <eurl> Tag URL
  - <email> Tag email

Example

```
http://127.0.0.1/pandora_console/include/api.php?op=set&op2=create_tag&other=tag_name|tag_description|tag_url|tag_email&other_mode=url_encode_separator_|&apipass=1234&user=admin&pass=pandora
```

set enable_disable_agent

It enables a disabled agent.

Call syntax:

- op=set (required)
- op2=enable_disable_agent (required)
- id=<agent_id> (required). It should be an agent id.

Examples

Example 1 (Disable agent 'example_id')

```
http://127.0.0.1/pandora_console/include/api.php?op=set&op2=enable_disable_agent&id=example_id&other=0&other_mode=url_encode_separator_|&apipass=1234&user=admin&pass=pandora
```

Example 2 (Enable agent 'example_id')

```
http://127.0.0.1/pandora_console/include/api.php?op=set&op2=enable_disable_agent&id=example_id&other=1&other_mode=url_encode_separator_|&apipass=1234&user=admin&pass=pandora
```

set gis_agent_only_position

It adds a new GIS position within any agent.

Call syntax:

- `op=set` (required).
- `op2=gis_agent_only_position` (required).
- `id=`<index> (required). Agent index.
- `other=`<serialized parameters> (required). Parameters to set the GIS using the `url_encode_separator_` to differentiate them (use as decimal separator the dot `.`):
  - <latitude> Latitude.
  - <longitude> Longitude.
  - <altitude> Altitude.

Example

```
http://127.0.0.1/pandora_console/include/api.php?apipass=1234&user=admin&pass=pandora&op=set&op2=gis_agent_only_position&id=582&other_mode=url_encode_separator_|&other=2%7C1%7C0
```

set gis_agent

> = 5.0

It adds a gis data agent.

Call syntax:

- op=set (required)
- op2=gis_agent_only_position (required)
- id=<índice> (compolsory). Agent index.
- other=<serialized parameters> (required). Gis data.
  - <latitude>
  - <longitude>
  - <altitude>
  - <ignore_new_gis_data>
  - <manual_placement>
  - <start_timestamp>
  - <end_timestamp>
  - <number_of_packages>
  - <description_save_history>
  - <description_update_gis>
  - <description_first_insert>

Ejemplo

```
http://127.0.0.1/pandora5/include/api.php?apipass=1234&user=admin&pass=pandora&o
```

```
p=set&op2=gis_agent&id=582&other_mode=url_encode_separator_|&other=2%7C2%7C0%7C0
%7C0%7C2000-01-01+01%3A01%3A01%7C0%7C666%7Caaa%7Cbbb%7Cccc
```

set reset_agent_counts

It updates agent alert and module counting.

Call syntax:

- op=set (required)
- op2=reset_agent_counts (required)
- id=<id_agent> (required). It must be an agent id or `All`.

Example

```
http://127.0.0.1/pandora_console/include/api.php?op=set&op2=reset_agent_counts&a
pipass=1234&user=admin&pass=pandora&id=All
```

set create_special_day

> = 5.1

It adds a new special day.

Call syntax:

- op=set (required)
- op2=create_special_day (required)
- other=<serialized parameters> (required)
  - <special day> Special day.
  - <same day> Same day.
  - <description> Description.
  - <id_group> Group ID.

Example

```
http://127.0.0.1/pandora_console/include/api.php?apipass=1234&user=admin&pass=pa
ndora&op=set&op2=create_special_day&other_mode=url_encode_separator_|&other=2014
-05-03|Sunday|desc|0
```

set update_special_day

> = 5.1

It updates an already defined special day configuration.

Call syntax:

- op=set (required)
- op2=update_special_day (required)
- id=<special day's id> (required)
- other=<serialized parameters> (required)
    - <special day> Special day.
    - <same day> Same day.
    - <description> Description.
    - <id_group> Group ID.

Example

```
http://127.0.0.1/pandora_console/include/api.php?apipass=1234&user=admin&pass=pandora&op=set&op2=update_special_day&id=1&other_mode=url_encode_separator_|&other=2014-05-03|Sunday|desc|0
```

set delete_special_day

> = 5.1

It deletes a special day.

Call syntax:

- op=set (required)
- op2=delete_special_day (required)
- id=<special day's id> (required)

Example

```
http://127.0.0.1/pandora_console/include/api.php?apipass=1234&user=admin&pass=pandora&op=set&op2=delete_special_day&id=1
```

set pagerduty_webhook

> = 5.1

It connects PagerDuty notifications with Pandora FMS alerts. This call is set in the Webhook option in PagerDuty's service to validate Pandora FMS alerts which have been previously linked to Pager Duty when they were validated from PagerDuty.

Call syntax:

- op=set (required)
- op2=pagerduty_webhook (required)
- id=alert (required)

Example

```
http://127.0.0.1/pandora_console/include/api.php?op=set&op2=pagerduty_webhook&apipass=1234&user=admin&pass=pandora&id=alert
```

set tag_user_profile

> = 6

It adds a tag into a user profile.

Call syntax:

- op=set (required)
- op2=tag_user_profile (required)
- id=id_user (required)
- id2=id_tag (required)
- other_mode=url_encode_separator_| (required)
- other=<id_group>|<id_profile> (required)

Example

```
http://127.0.0.1/pandora_console/include/api.php?op=set&op2=tag_user_profile&apipass=1234&user=admin&pass=pandora&id=1&id2=2&other_mode=url_encode_separator_|&other=122|3
```

set tag

> = 6

It adds a tag into Pandora FMS.

Call syntax:

- op=set (required)
- op2=tag (required)
- id=name (required)
- other_mode=url_encode_separator_| (required)
- other=<description>|<url>|<email>|<phone>

Example

```
http://127.0.0.1/pandora_console/include/api.php?op=set&op2=tag&apipass=1234&user=admin&pass=pandora&id=test&other_mode=url_encode_separator_|&other="a test"|http://www.artica.es|test@artica.es|01189998819991197253
```

set add_tag_module

> = 6

It adds a tag to a module.

Call syntax:

- op=set (required)
- op2=add_tag_module (required)
- id=id_module (required)
- id2=id_tag (required)

Example

```
http://127.0.0.1/pandora_console/include/api.php?op=set&op2=add_tag_module&apipa
ss=1234&user=admin&pass=pandora&id=1&id2=2
```

set remove_tag_module

> = 6

It removes a tag from a module.

Call syntax:

- op=set (required)
- op2=add_tag_module (required)
- id=id_module (required)
- id2=id_tag (required)

Example

```
http://127.0.0.1/pandora_console/include/api.php?op=set&op2=remove_tag_module&ap
ipass=1234&user=admin&pass=pandora&id=1&id2=2
```

set planned_downtimes_created

> = 5.1

It adds new planned downtime.

- op=set
- op2=planned_downtimes_created
- other=<description>;<date_from>;<date_to>;<id_group>;<monday>;
- <tuesday>;<wednesday>;<thursday>;<friday>;<saturday>;<sunday>;<periodically_time_from>;<periodically_time_to>;
- <periodically_day_from>;<periodically_day_to>;<type_downtime>;<type_execution>;<type_periodi

city>;<id_user>;

Date format needs to be MM/DD/YYYY for this call to work properly.

Examples

Once:

```
http://localhost/pandora_console/include/api.php?op=set&op2=planned_downtimes_created&apipass=1234&user=admin&pass=pandora&id=testing&other=testing|11/05/2018|11/16/2018|0|1|1|1|1|1|1|1|12:06:00|19:06:00|1|31|quiet|once|weekly|admin&other_mode=url_encode_separator_|
```

Periodically:

```
http://localhost/pandora_console/include/api.php?op=set&op2=planned_downtimes_created&apipass=1234&user=admin&pass=pandora&id=testing&other=testing|11/05/2018|11/16/2018|0|1|1|1|1|1|1|1|12:06:00|19:06:00|1|31|quiet|periodically|weekly|admin&other_mode=url_encode_separator_|
```

set planned_downtimes_edit

> = 754

Edit a planned stop.

- `op=set` .
- `op2=planned_downtimes_edit` .
- `id=` planned shutdown identifier.
- `other=` all parameters are optional:
  - <name>
  - <description>
  - <date from>
  - <date to>
  - <time from>
  - <time to>
  - <id group>
  - <monday>
  - <tuesday>
  - <wednesday>
  - <thursday>
  - <friday>
  - <saturday>
  - <sunday>
  - <periodically_day_from>
  - <periodically_day_to>

- <stop type>
- <ejecution type>
- <periodicity type>

The date format must be YYYYY/MM/DD (year/month/day)
for this call to work properly.

Example:

```
http://localhost/pandora_console/include/api.php?op=set&op2=planned_downtimes_edit&apipass=1234&user=admin&pass=pandora&id=2&other=testing2|test2|2022/05/10|2022/06/12|19:03:03|19:55:00|0|0|0|0|0|0|0|1|31|quiet|once|weekly&other_mode=url_encode_separator_|
```

set planned_downtimes_add_agents

> = 754

Add Agents to the planned stop.

- op=set .
- op2=planned_downtimes_agents .
- id=planned stop identifier.
- other=identifier of each Agent separated by semicolons:
    - <id_agent1;id_agent2;id_agent3;....id_agentn;>

Example:

```
http://127.0.0.1/pandora_console/include/api.php?op=set&op2=planned_downtimes_add_agents&apipass=1234&user=admin&pass=pandora&id=4&other=1;2;3&other_mode=url_encode_separator_|
```

set planned_downtimes_delete_agents

> = 754

Removes Agents (and the Modules of those agents) from the planned shutdown.

- op=set .
- op2=planned_downtimes_delete_agents .
- id=planned stop identifier.
- other=identifier of each Agent separated by semicolons:
    - <id_agent1;id_agent2;id_agent3;....id_agentn;>

Example:

```
http://127.0.0.1/pandora_console/include/api.php?op=set&op2=planned_downtimes_de
lete_agents&apipass=1234&user=admin&pass=pandora&id=4&other=1;2;3&other_mode=url
_encode_separator_|
```

set planned_downtimes_additem

> = 5.1 It adds new items of a planned downtime.

- op=set
- op2=planned_downtimes_additem
- id=planned_downtime_id
- Other=<id_agent1;id_agent2;id_agent3;....id_agentn;>;<name_module1;name_module2;name_module3;......name_modulen;>

Example

```
http://127.0.0.1/pandora_console/include/api.php?op=set&op2=planned_downtimes_ad
ditem&apipass=1234&user=admin&pass=pandora&id=123&other=1;2;3;4%7CStatus;Unkown_
modules%20&other_mode=url_encode_separator_|
```

set planned_downtimes_deleted

> =5.1

It deletes a planned downtime

- op=set
- op2=planned_downtimes_deleted
- id=planned_downtime_id

Example

```
http://127.0.0.1/pandora_console/include/api.php?op=set&op2=planned_downtimes_de
leted&apipass=1234&user=admin&pass=pandora&id=10
```

set create_synthetic_module

> =5.1SP4

It adds a new synthetic module:

- op=set
- op2=create_synthetic_module
- id=Agent name to add module
- id2=<use_agent_alias>
- Other=<name_module><synthetic_type><AgentName;Operation;NameModule> OR <AgentName;NameModule> OR <Operation;Value>

- Data of module:

```
 In arithmetic creations the first piece of data is Agent without operator:
AgentName;NameModule or if you type in a value: <Operation;Value>. The rest of
values follow this one: <AgentName;Operation;NameModule> or <Operation;Value>
 average: <AgentName;Operation;NameModule> or <Operation;Value>
```

- Operation:

```
  arithmetic: ADD, SUB, MUL, DIV
  average: Only AVG
```

Example

```
http://127.0.0.1/pandora_console/include/api.php?op=set&op2=create_synthetic_module&apipass=1234&user=admin&pass=pandora&id=test&other=Test|arithmetic|Agent%20Name;Module%20Name|Agent%20Name2;ADD;Module%20Name2&other_mode=url_encode_separator_|
```

```

http://127.0.0.1/pandora_console/include/api.php?op=set&op2=create_synthetic_module&apipass=1234&user=admin&pass=pandora&id=pepito&other=prueba|average|Agent%20Name;AVG;Name%20Module|Agent%20Name2;AVG;Name%20Module2&other_mode=url_encode_separator_|
```

set create_service

> = 7

It creates a new service.

- op=set
- op2=create_service
- other=<name>;<description>;<id_group>;<critical>;<warning>;<id_agent>;<sla_interval>;<sla_limit>;

<id_warning_module_template_alert>;<id_critical_module_template_alert>;<id_critical_module_sla_template_alert>;

Example

```

http://127.0.0.1/pandora_console/include/api.php?op=set&op2=create_service&return_type=json&other=test1%7CDescripcion%7C12%7C1%7C0.5%7C1&other_mode=url_encode_separator_%7C&apipass=1234&user=admin&pass=pandora
```

set update_service

> = 7

It modifies a service.

- op=set
- op2=update_service
- id=service id
- other=<name>;<description>;<id_group>;<critical>;<warning>;<id_agent>;<sla_interval>;<sla_limit>;

<id_warning_module_template_alert>;<id_critical_module_template_alert>;<id_critical_module_sla_template_alert>;

Example

```
http://127.0.0.1/pandora_console/include/api.php?op=set&op2=update_service&return_type=json&id=1&other=test2%7CDescripcion2%7C%7C%7C0.6%7C&other_mode=url_encode_separator_%7C&apipass=1234&user=admin&pass=pandora
```

set add_element_service

> = 7

It adds elements to a service.

- op=set
- op2=add_element_service
- id=service id
- Other=Json with elements in base64

The structure of the json should be as follows:

```
[
  {
    "type": "agent",
    "id": 2,
    "description": "Test1",
    "weight_critical": 0,
    "weight_warning": 0,
    "weight_unknown": 0,
    "weight_ok": 0
  },
  {
    "type": "module",
    "id": 1,
    "description": "Test2",
    "weight_critical": 0,
    "weight_warning": 0,
    "weight_unknown": 0,
    "weight_ok": 0
  },
  {
    "type": "service",
    "id": 3,
    "description": "Test3",
    "weight_critical": 0,
    "weight_warning": 0,
    "weight_unknown": 0,
    "weight_ok": 0
  }
]
```

In each type, the id field refers to different things:

- If it belongs to agent type, it is agent id

- If it belongs to module type, it is agent module id

- If it belongs to service type, it is the service id to be added.

Example

```
http://127.0.0.1/pandora_console/include/api.php?op=set&op2=add_element_service&
return_type=json&id=6&other=W3sidHlwZSI6ImFnZW50IiwiaWQiOjIsImRlc2NyaXB0aW9uIjoi
amlqaWppIiwid2VpZ2h0X2NyaXRpY2FsIjowLCJ3ZWlnaHRfd2FybmluZyI6MCwid2VpZ2h0X3Vua25v
d24iOjAsIndlaWdodF9vayI6MH0seyJ0eXBlIjoibW9kdWxlIiwiaWQiOjEsImRlc2NyaXB0aW9uIjoi
SG9sYSBxdWUgdGFsIiwid2VpZ2h0X2NyaXRpY2FsIjowLCJ3ZWlnaHRfd2FybmluZyI6MCwid2VpZ2h0
X3Vua25vd24iOjAsIndlaWdodF9vayI6MH0seyJ0eXBlIjoic2VydmljZSIsImlkIjozLCJkZXNjcmlw
dGlvbiI6ImplamVqZWplIiwid2VpZ2h0X2NyaXRpY2FsIjowLCJ3ZWlnaHRfd2FybmluZyI6MCwid2Vp
Z2h0X3Vua25vd24iOjAsIndlaWdodF9vayI6MH1d&other_mode=url_encode_separator_%7C&api
pass=1234&user=admin&pass=pandora
```

set metaconsole_synch

> = 7

It adds license key to the Command Center (Metaconsole) and performs the synchronization with nodes.

- op=set
- op2=metaconsole_synch
- id=License key

Example

```
http://127.0.0.1/pandora_console/enterprise/meta/include/api.php?op=set&op2=meta
console_synch&id=LICENSEKEY&apipass=1234&user=admin&pass=pandora
```

set migrate_agent

> = 7.21 ONLY METACONSOLE

It adds a selected agent to the agent migration queue.

- op=set
- op2=migrate_agent
- id=id_agent to migrate
- Other=origin node name| destination node name | (true|false) not to migrate historical database
- other_mode=url_encode_separator_|
- Return_type=string, json, etc

Example

```
http://127.0.0.1/pandora_console/include/api.php?op=set&op2=migrate_agent&apipass=1234&user=admin&pass=pandora&id=2&other=nova|fringe|0&other_mode=url_encode_separator_|&return_type=string
```

set new_cluster

> = 7.0

It creates an agent cluster

It creates a monitoring cluster with agents and items to monitor different nodes.

- op=set
- op2=new_cluster
- other=cluster_name| cluster_type| description| group_id
- other_mode=url_encode_separator_

Example

```
http://127.0.0.1/pandora_console/include/api.php?op=set&op2=new_cluster&other=nombre_cluster%7CAA%7Cdescripcion%7C12&other_mode=url_encode_separator_%7C&apipass=1234&user=admin&pass=pandora
```

set add_cluster_agent

> = 7.0

It adds an agent to a cluster.

- op=set
- op2=add_cluster_agent
- Other=json with elements in base64
- other_mode=url_encode_separator_

The json structure should be:

```
[
  {
    "id": 5,
    "id_agent": 2
  },
  {
    "id": 5,
    "id_agent": 3
  }
]
```

```
http://127.0.0.1/pandora_console/include/api.php?op=set&op2=add_cluster_agent&ot
her=WwogIHsKICAgICJpZCI6IDUsCiAgICAiaWRfYWdlbnQiOiAyCiAgfSwKICB7CiAgICAiaWQiOiA1
LAogICAgImlkX2FnZW50IjogMwogIH0KXQ==&other_mode=url_encode_separator_%7C&apipass
=1234&user=admin&pass=pandora
```

set add_cluster_item (active/active)

> = 7.0

It adds an active/active item to a cluster

- op=set
- op2=add_cluster_item
- Other=json with elements in base64
- other_mode=url_encode_separator_

The json structure should be:

```
[
  {
    "name": "Swap_Used",
    "id_cluster": 5,
    "type": "AA",
    "critical_limit": 80,
    "warning_limit": 60
  },
  {
    "name": "TCP_Connections",
    "id_cluster": 5,
    "type": "AA",
    "critical_limit": 80,
    "warning_limit": 60
  }
]
```

```
http://127.0.0.1/pandora_console/include/api.php?op=set&op2=add_cluster_item&oth
er=WwogIHsKICAgICJuYW1lIjogIlN3YXBfVXNlZCIsCiAgICAiaWRfY2x1c3RlciI6IDUsCiAgICAid
HlwZSI6ICJBQSIsCiAgICAiY3JpdGljYWxfbGltaXQiOiA4MCwKICAgICJ3YXJuaW5nX2xpbWl0IjogN
jAKICB9LAogIHsKICAgICJuYW1lIjogIlRDUF9Db25uZWN0aW9ucyIsCiAgICAiaWRfY2x1c3RlciI6I
```

```
DUsCiAgICAidHlwZSI6ICJBQSIsCiAgICAiY3JpdGljYWxfbGltaXQiOiA4MCwKICAgICJ3YXJuaW5nX
2xpbWl0IjogNjAKICB9Cl0=&other_mode=url_encode_separator_%7C&apipass=1234&user=ad
min&pass=pandora
```

set add_cluster_item (active/passive)

> = 7.0

It adds a passive/actve item to a cluster.

- op=set
- op2=add_cluster_item
- Other=json with elements in base64
- other_mode=url_encode_separator_

The json structure should be:

```
[
  {
    "name": "DiskUsed_/proc/kcore",
    "id_cluster": 5,
    "type": "AP",
    "is_critical": 1
  },
  {
    "name": "DiskUsed_/proc/sched_debug",
    "id_cluster": 5,
    "type": "AP",
    "is_critical": 1
  }
]
```

```
http://127.0.0.1/pandora_console/include/api.php?op=set&op2=add_cluster_item&oth
er=WwogIHsKICAgICJuYW1lIjogIkRpc2tVc2VkXy9wcm9jL2tjb3JlIiwKICAgICJpZF9jbHVzdGVyI
jogNSwKICAgICJ0eXBlIjogIkFQIiwKICAgICJpc19jcml0aWNhbCI6IDEKICB9LAogIHsKICAgICJuY
W1lIjogIkRpc2tVc2VkXy9wcm9jL3NjaGVkX2RlYnVnIiwKICAgICJpZF9jbHVzdGVyIjogNSwKICAgI
CJ0eXBlIjogIkFQIiwKICAgICJpc19jcml0aWNhbCI6IDEKICB9Cl0=&other_mode=url_encode_se
parator_%7C&apipass=1234&user=admin&pass=pandora
```

set delete_cluster

> = 7.0

It deletes a cluster.

- op=set
- op2=delete_cluster
- id=id of the cluster to delete

```
http://127.0.0.1/pandora_console/include/api.php?op=set&op2=delete_cluster&id=7&
```

```
apipass=1234&user=admin&pass=pandora
```

set delete_cluster_agents

> = 7.0

It unpairs an agent from a cluster

- op=set
- op2=delete_cluster_agents
- Other=JSON format with elements (see structure below)
- other_mode=url_encode_separator_

The json structure should be:

```
[
  {
    "id": 5,
    "id_agent": 2
  },
  {
    "id": 5,
    "id_agent": 3
  }
]
```

Example

```
http://127.0.0.1/pandora_console/include/api.php?op=set&op2=delete_cluster_agent
s&other=WwogIHsKICAgICJpZCI6IDUsCiAgICAiaWRfYWdlbnQiOiAyCiAgfSwKICB7CiAgICAiaWQi
OiA1LAogICAgImlkX2FnZW50IjogMwogIH0KXQ==&other_mode=url_encode_separator_%7C&api
pass=1234&user=admin&pass=pandora
```

set delete_cluster_item

> = 7.0

It deletes an item from a cluster.

- op=set
- op2=delete_cluster_item
- id=id of item to eliminate

```
http://127.0.0.1/pandora_console/include/api.php?op=set&op2=delete_cluster_item&
id=9&apipass=1234&user=admin&pass=pandora
```

set create_policy

> = 7.0. 725

It creates a policy. Both the policy name, which cannot be repeated, and the id_group, which must exist in the database, are required.

- op=set
- op2=create_policy
- Other=policy name (required) | id_del grupo (required) | description;
- other_mode=url_encode_separator_|
- Return_type=(string, csv, json).

Example

```
http://localhost/pandora_console/include/api.php?op=set&op2=create_policy&apipas
s=1234&user=admin&pass=pandora&other=name%20Policy|11|this%20description&other_m
ode=url_encode_separator_|&return_type=json
```

set update_policy

> = 7.0. 725

It updates a policy. The policy name cannot be repeated and the id_group must exist in the database. It returns 0 (false) or the updated policy id (true).

- op=set
- op2=update_policy
- id=policy id
- Other=policy name | group_id | description;
- other_mode=url_encode_separator_|
- Return_type=(string, csv, json).

Example

```
http://localhost/pandora_console/include/api.php?op=set&op2=update_policy&apipas
s=1234&user=admin&pass=pandora&id=17&other=policy2|11|this%20description&other_m
ode=url_encode_separator_|&return_type=json
```

set delete_policy

> = 7.0. 725

It deletes a policy. It is required to enter the policy id, which must exist in order to delete it. It returns 0 (false) or 1 (true).

- op=set
- op2=delete_policy

- id=policy id
- Return_type=(string, csv, json).

Example

```
http://localhost/pandora_console/include/api.php?op=set&op2=delete_policy&apipas
s=1234&user=admin&pass=pandora&id=10&return_type=json
```

set add_collections_policy

> = 7.0. 725

It adds a collection to a policy. It is necessary to enter the policy id and for such a policy to exist, as an id, name or short name of the collection you wish to add, which should exist too. Both fields are required. It returns 0 (false) or the id of the collection added to a policy (true).

- op=set
- op2=add_collections_policy
- id=policy id
- id2=id, name or short name
- Return_type=(string, csv, json).

Example

```
http://localhost/pandora_console/include/api.php?op=set&op2=add_collections_poli
cy&apipass=1234&user=admin&pass=pandora&return_type=string&id=4&id2=apache_plugi
n
```

set remove_collections_policy

> = 7.0. 725

To mark a policy collection that is yet to be deleted, it is necessary:

- A policy id, and for such a policy to exist. Mandatory.
- An id, name or short name of the collection you wish to remove, which must exist. Mandatory.
- And 0 or 1: 1 marks it as yet to be deleted and 0 removes said state. If this field is empty, it will try to check it so that it can be deleted by default.

It returns 0 (false) or 1 (true).

- op=set
- op2=remove_collections_policy
- id= policy id
- id2=id, name or short name
- Other=(bool) 0 or 1
- Return_type=(string, csv, json).

Example

```
http://localhost/pandora_console/include/api.php?op=set&op2=remove_collections_p
olicy&apipass=1234&user=admin&pass=pandora&return_type=string&id=4&id2=2&other=1
.
```

set create_plugins_policy

> = 7.0. 725

To add a plugin to a policy the following are necessary:

- A policy id and for such a policy to exist. Mandatory.
- The plugin string to run. Mandatory.

It returns 0 (false) or the plugin id that has been added to a policy (true).

- op=set
- op2=create_plugins_policy
- id=policy id
- id2=text string
- Return_type=(string, csv, json).

Example

```
http://localhost/pandora_console/include/api.php?op=set&op2=create_plugins_polic
y&apipass=1234&user=admin&pass=pandora&return_type=json&id=2&id2=echo%201
```

set delete_plugins_policy

> = 7.0. 725

These are the requirements to remove a plugin from a policy:

- A policy id and for such a policy to exist. Mandatory.
- A plugin id to be removed. Mandatory.
- 0 or 1: 1 means the plugin is yet to be deleted and 0 removes this state. If not added, this field is marked as pending to be deleted by default.

It returns 0 (false) or 1 (true).

- op=set
- Op2=delete_plugins_policy
- id=policy id
- id2=plugin id
- Other=0 or 1
- Return_type=(string, csv, json).

Example

```
http://localhost/pandora_console/include/api.php?op=set&op2=delete_plugins_polic
y&apipass=1234&user=admin&pass=pandora&return_type=string&id=2&id2=1&other=1
```

set linking_policy

> = 7.0. 725

To link an unlinked policy module it is necessary:

- An agent id module that is unlinked.

It returns 0 (false) or 1 (true).

- op=set
- op2=linking_policy
- id=module agent id
- Return_type=(string, csv, json).

Example

```
http://localhost/pandora_console/include/api.php?op=set&op2=linking_policy&apipa
ss=1234&user=admin&pass=pandora&return_type=json&id=28
```

set create_alerts_policy

> = 7.0. 725

To create policy alert, these are necessary:

- A policy id that must exist (required).
- A policy module id, if it belongs to the normal type it is mandatory.
- A template id that must exist (required).
- Whether it is external or not: 0 (normal), 1 (external). It is 0 (normal) by default.
- If it belongs to the external type (1) a module name that must match it.

It returns 0 (false) or the id of the alert added in the policy (true).

- op=set
- op2=create_alerts_policy
- id=policy id
- id2=policy module id
- Other=policy module id|id template|0 or 1|module name.
- other_mode=url_encode_separator_|
- Return_type=(string, csv, json).

Example

```

http://localhost/pandora_console/include/api.php?op=set&op2=create_alerts_policy
```

```
&apipass=1234&user=admin&pass=pandora&return_type=string&id=2&other=0|2|1|cpu%20
load&other_mode=url_encode_separator_|
```

set update_alerts_policy

> = 7.0. 725

To update policy alerts, it is necessary:

- A policy id that must exist (required).
- Active 0 | Waiting 1
- enable 0 | disable 1

It returns 0 (false) or the id of the updated alert in the policy (true).

- op=set
- op2=update_alerts_policy
- id=policy id
- Other=0 or 1 (activate/deactivate) | 0 or 1 (enable/disable).
- other_mode=url_encode_separator_|
- Return_type=(string, csv, json).

Example

```
http://localhost/pandora_console/include/api.php?op=set&op2=update_alerts_policy
&apipass=1234&user=admin&pass=pandora&return_type=string&id=7&other=1|1&other_mo
de=url_encode_separator_|
```

set delete_alerts_policy

> = 7.0. 725

To remove an alert from a policy it is necessary:

- A policy alert id that must exist (required).
- 0 | 1 mark it as yet to bet removed or not. If this field is empty, 1 is set by default.
- Active 0 | Standby 1

It returns 0 (false) or 1 (true).

- op=set
- Op2=delete_alerts_policy
- id=policy id
- id2=0 or 1.
- Return_type=(string, csv, json).

Example

```
http://localhost/pandora_console/include/api.php?op=set&op2=delete_alerts_policy
&apipass=1234&user=admin&pass=pandora&return_type=string&id=7&id2=1
```

set create_alerts_actions_policy

> = 7.0. 725

To add an action to a policy alert it is necessary:

- A policy alert id that must exist (required).
- A policy action id that must exist (required).
- You may add a minimum number of shots or a maximum number of shots by default: 0.

It returns 0 (false) or the id of the action added to the policy alert (true).

- op=set
- op2=create_alerts_action_policy
- id=policy id
- id2=action id
- Other=min fires | max fires
- other_mode=url_encode_separator_|
- Return_type=(string, csv, json).

Example

```
http://172.16.0.2/pandora_console/include/api.php?op=set&op2=create_alerts_actio
ns_policy&apipass=1234&user=admin&pass=pandora&return_type=string&id=1&id2=4&oth
er=5|2&other_mode=url_encode_separator_|
```

set delete_alerts_actions_policy

> = 7.0. 725

To remove an action from a policy alert, it is necessary:

- A policy alert id that must exist (required).
- A policy action id that must exist (required).
- op=set
- op2=delete_alerts_action_policy
- id=policy id
- id2=action id
- Return_type=(string, csv, json).

Example

```
http://localhost/pandora_console/include/api.php?op=set&op2=delete_alerts_action
s_policy&apipass=1234&user=admin&pass=pandora&return_type=string&id=1&id2=4
```

set remove_agents_policy

> = 7.0. 725

To remove an agent from a policy, it is necessary:

- A policy id that must exist (required).
- An agent id that must exist (required).
- op=set
- op2=remove_agents_policy
- id=policy id
- id2=group id
- Return_type=(string, csv, json).

Example

```
http://localhost/pandora_console/include/api.php?op=set&op2=remove_agents_policy
&apipass=1234&user=admin&pass=pandora&return_type=string&id=5&id2=2
```

set add_groups_policy

> = 7.0. 725

To add a policy group, it is necessary:

- An existing policy id (required).
- An existing group id (required).
- op=set
- op2=add_groups_policy
- id=policy id
- id2=group id
- Return_type=(string, csv, json).

Example

```
http://localhost/pandora_console/include/api.php?op=set&op2=add_groups_policy&ap
ipass=1234&user=admin&pass=pandora&return_type=string&id=7&id2=12
```

set remove_groups_policy

> = 7.0. 725

To remove an agent from a policy, it is necessary:

- An existing policy id (required).
- An existing group id (required).
- op=set
- op2=add_groups_policy
- id=policy id

- id2=group id
- Return_type=(string, csv, json).

Example

```
http://localhost/pandora_console/include/api.php?op=set&op2=remove_groups_policy
&apipass=1234&user=admin&pass=pandora&return_type=string&id=5&id2=2
```

set create_os

> = 7.0. 727

It creates a new operating system with the data as parameters.

Call syntax:

- op=set (required)
- op2=create_os (required)
- other=<serializead parameters> (required). They are the system's data, serializead in this order:
  - <name>
  - <description>
  - <icon>

Example

```
http://127.0.0.1/pandora_console/include/api.php?op=set&op2=create_os&other=os_n
ame%7Cos_description%7Cos_icon.png&other_mode=url_encode_separator_%7C&apipass=1
234&user=admin&pass=pandora
```

set update_os

> = 7.0. 727

It updates the operating system with data as parameters.

Call syntax:

- op=set (required)
- op2=create_os (required)
- other=<serializead parameters> (required). They are the operating system's data, serialized in this order:
  - <name>
  - <description>
  - <icon>

Example

```
http://127.0.0.1/pandora_console/include/api.php?id=107&op=set&op2=update_os&oth
er=os_name_to_update%7Cos_description_to_update%7Cos_icon_to_update.png&other_mo
```

```
de=url_encode_separator_%7C&apipass=1234&user=admin&pass=pandora
```

set disabled_and_standby

> = 7.0. 728

It disables an agent and, if it has remote configuration, it makes it go into standby mode.

Call Syntax:

- op=set (required)
- op2=disabled_and_standby (required)
- id=<agent id> (required). In the Command Center (Metaconsole), it is the id of the agent, that is to say, the id of the node.
- id2=<id del nodo> (required in the Command Center (Metaconsole), not needed in the node)
- other=<valor> (optional) Value (0 to enable and 1 to disable) of the new status. If no value is specified, it disables the agent.

Example

```
http://127.0.0.1/pandora_console/include/api.php?id=2&op=set&op2=disabled_and_st
andby&other=1&other_mode=url_encode_separator_%7C&apipass=1234&user=admin&pass=p
andora
```

```
http://127.0.0.1/pandora_console/include/api.php?id=2&id2=1&op=set&op2=disabled_
and_standby&other=1&other_mode=url_encode_separator_%7C&apipass=1234&user=admin&
pass=pandora
```

set validate_traps

> = 7.0. 728

It allows to validate traps.

- op=set
- op2=validate_traps
- id=trap id

Example

```
https://127.0.0.1/pandora_console/include/api.php?op=set&op2=validate_traps&id=1
&other_mode=url_encode_separator_%7C&apipass=1234&user=admin&pass=pandora
```

set delete_traps

> = 7.0. 728

It allows to delete a trap:

- op=set
- Op2=delete_traps
- id=trap id

Example

```
http://127.0.0.1/pandora_console/include/api.php?id=2&op=set&op2=delete_traps&id
=1&other_mode=url_encode_separator_%7C&apipass=1234&user=admin&pass=pandora
```

set access_process

> = 7.0NG. 728

This function allows to manage the access to a third application from records in the system audit log of Pandora FMS. It uses different parameters to process one of these registry actions in your application and avoids concurrent access by administrator users. Optionally, it can ban access to non-administrator users.

Call syntax:

- op=set
- op2=access_process
- other =<required parameters»
  - <user_id>

The id of the user trying to access the application, this data is recorded and checked in the audit to filter user access, exit, exploration or navigation through the application.

- <action> - (login,logout,exclude,browse)

login: It is used to request access to the application. It deletes the text string `free` and registers your access in Pandora FMS audit or returns the text string `denied` if there is another user in the audit records who has previously logged in the system and has not yet logged out. It will also return `denied` if the administrator-only access parameter is enabled and the user is not an admin.

logout: Logs a user's application logout so that other users are again allowed to access.

browse: It must be used on all pages of the external application, as it checks whether this user is still the last one to access or be expelled from the application.

exclude: It expels the currently registered user and registers the access of another one sent in the user_id parameter

- <app ip address>

IP address of the application from which you access it. It is registered and checked in the Pandora FMS audit to check user access status in the application.

- <app name>

Name of the application you are accessing from. It is registered and checked in the Pandora FMS audit to check the user access status in the application.

- <only admin access>

It forbids the access to non-administrator users.

Example (requires adapted environment)

```
http://127.0.0.1/pandora_console/include/api.php?op=get&op2=access_process&other
=1%7Clogin%7C192.168.50.25%7Cexternal_app%7C0&other_mode=url_encode_separator_%7
C&apipass=1234&user=admin&pass=pandora
```

set create_event_response

It creates a new event response.

Syntax:

- op=set (required).
- op2=create_event_response (required).
- return_type=csv or return_type=json (one of two required)
- other=< serialized parameters > (all required). In the following order:
  - < name >
  - < description >
  - < target > : If the forth parameter (< type >) is command, here it is necessary to indicate the desired command to execute the response to the event. On the other hand, if it is url, indicate the URL you wish to access as a response to the event.
  - < type > : command or url.
  - < group id > The user must have permissions over the indicated group.
  - < modal window width > In pixels.
  - < modal window height > In pixels.
  - < new window > 1 or 0. It indicates whether the URL is shown on a new window (1) or on a modal window (0).
  - < command timeout > Time for wait a response in seconds.
  - < parameters > Parameters that complement the execution of the command.
  - < server id > Id of the server in which the command is to be executed. For the local console, the value is 0.

Example (see call syntax):

```
…/include/api.php?op=set&op2=create_event_response&other=response|description%20
response|touch|command|0|650|400|0|response|0|90&other_mode=url_encode_separator
_|&apipass=1234&user=admin&pass=pandora
```

set update_event_response

It edits an existing event response. The user must have permissions on the group to which the response belongs, in order to edit it.

Call syntax:

- op=set (required)
- op2=update_event_response (required)
- Return_type=csv|json (required)
- id=<id of event response> (required)
- other=<serialized parameters> (optional). In the following order:
    - <name>
    - <description>
    - <target> If the fourth parameter (<type>) is *command*, indicate here the command that you wish for the event response to execute. On the other hand, if it is *url*, indicate the URL you wish to access as a response to the event.
    - <type> *command* or *url*.
    - <group id> The user must have permissions on the indicated group.
    - <modal window width> In pixels.
    - <modal window height> In pixels.
    - <new window> 1 or 0. It indicates whether the URL is displayed on a new window (1) or on a modal window (0).
    - <command timeout> Response in seconds.
    - <parameters> Parameters that complement the command's execution.
    - <server> Id of the server where the command is to be executed. For the local console the value is 0.

Example

```
http://localhost/pandora_console/include/api.php?op=set&op2=update_event_response&id=5&other=response|description%20response|touch|command|0|650|400|0|response|0|90&other_mode=url_encode_separator_|&apipass=1234&user=admin&pass=pandora
```

set delete_event_response

It deletes an event response. The user must have permissions on the group to which the response belongs in order to delete it.

Call syntax:

- op=set (required)
- op2=delete_event_reponse (required)
- Return_type=csv|json
- id=<event response id> (required)

Example

```
http://localhost/pandora_console/include/api.php?op=set&op2=delete_event_respons
e&id=7&apipass=1234&user=admin&pass=pandora
```

set create_user_profile_info

It creates a new user profile.

Call syntax:

- op=set (required)
- op2=create_user_profile_info (required)
- Return_type=csv|json (required)
- other=<serialized parameters> In the following order:
    - <name> (required)
    - <access bits» 1 if bit is to be set and 0 if not. They are not required but if no value is specified, the bit will be marked as not active. In this order:

IR|IW|IM|AR|AW|AD|LW|LM|UM|DM|ER|EW|EM|RR|RW|RM|MR|MW|MM|VR|VW|VM|PM.

To know which permissions should be activated, click on this link.

Example

This example creates a profile that gives read-only access to Pandora, that is, it is the same as the predefined profile *Operator (read)*.

```
http://localhost/pandora_console/include/api.php?op=set&op2=create_user_profile_
info&return_type=json&other=API_profile%7C1%7C0%7C0%7C1%7C0%7C0%7C0%7C0%7C0%7C0%
7C1%7C0%7C0%7C1%7C0%7C0%7C1%7C0%7C0%7C1%7C0%7C0%7C0&other_mode=url_encode_separa
tor_%7C&apipass=1234&user=admin&pass=pandora
```

set update_user_profile_info

It updates a new existing user profile.

Call syntax:

- op=set (required)
- op2=update_user_profile_info (required)
- other=<serialized parameters> (all optional). In the following order:
    - <name>
    - <access bits» 1 if you want to set the bit and 0 if you want to disable it. If no value is specified, it will not change. In this order:

IR|IW|IM|AR|AW|AD|LW|LM|UM|DM|ER|EW|EM|RR|RW|RM|MR|MW|MM|VR|VW|VM|PM.

To find out which permissions should be activated, click on this link.

Example

In this example the name of the profile with ID 6 is replaced by *API_profile_updated* and it is granted all permissions (read, write and management) related to agents and permission to read events is withdrawn.

```
http://localhost/pandora_console/include/api.php?op=set&op2=update_user_profile_info&return_type=json&id=6&other=API_profile_updated%7C%7C%7C%7C1%7C1%7C1%7C%7C%7C%7C%7C%7C0%7C%7C%7C%7C%7C%7C%7C%7C%7C%7C%7C%7C&other_mode=url_encode_separator_%7C&apipass=1234&user=admin&pass=pandora
```

set delete_user_profile_info

It deletes a user profile. It also deletes all that profile assignments to any user.

Call syntax:

- op=set (required)
- op2=delete_user_profile_info (required)
- Return_type=csv|json
- id=<id del perfil> (required)

Example

```
http://localhost/pandora_console/include/api.php?op=set&op2=delete_user_profile_info&return_type=json&id=8&other_mode=url_encode_separator_%7C&apipass=1234&user=admin&pass=pandora
```

set create_event_filter

It creates an event filter.

Call syntax:

- op=set (required)
- op2=create_event_filter (required)
- other=<serialized parameters>. In this order:
  - <id_group_filter>

  - <id_group>
  - <event_type [new_agent|alert_recovered|alert_ceased|alert_fired|configuration_change|error|alert_manual_valdation|critical|normal|warning|not_normal|recon_host_detected|system|unknown|going_unknown]>
  - <severity [0|1|2|3|4|5|6|20|21|34]> (0: Maintenance, 1: Informative, 2: Normal, 3: Warning, 4: Critical, 5: Minor, 6: Principal, 20: Not normal, 21: Critical/Normal, 34: Warning/Critical)
  - <event_status [2|3|0|1]> (0: Only new, 1: only validated, 2: only in process, 3: only not validated)
  - <free_search>
  - <agent_id>
  - <pagination_size [25|50|100|200|500]>
  - <max_hours_old>
  - <id_user_ack>
  - <duplicate [0|1]> (0: All the events, 1: grouped events)
  - <date_from> (format: AAAA/MM/DD)
  - <date_to> (format: AAAA/MM/DD)
  - <events_with_tags> (format:

```
["tag_id_1", "tag_id_2", "tag_id_3", "…"]
```

)

- <events_without_tags> (format:

```
["tag_id_1", "tag_id_2", "tag_id_3", "…"]
```

)

- <alert_events [0|1]> (0: filter by event aletrs, 1: Alert events only)
- <module_id>
- <source>
- <id_extra>
- <user_comment>

Examples

```
http://127.0.0.1/pandora_console/include/api.php?op=set&op2=create_event_filter&id=test&other=%7C%7Cerror%7C4%7C%7C%7C1%7C%7C12%7C%7C%7C2018-12-09%7C2018-12-13%7C[%226%22]%7C[%2210%22,%226%22,%223%22]%7C1%7C10%7C%7C%7C&other_mode=url_encode_separator_%7C&apipass=1234&user=admin&pass=pandora
```

set update_event_filter

It updates an event filter.

Call syntax:

- op=set (required)
- op2=update_event_filter (required)
- id=<event_filter_id> (required). It must be an existing event filter id.
- other=<serialized parameters> (optional). In the following order:
    - <id_group_filter>
    - <id_group>
    - <event_type [new_agent|alert_recovered|alert_ceased|alert_fired|configuration_change|error|alert_manual_valdation|critical|normal|warning|not_normal|recon_host_detected|system|unknown|going_unknown]>
    - <severity [0|1|2|3|4|5|6|20|21|34]> (0: Maintenance, 1: Informative, 2: Normal, 3: Warning, 4: Critical, 5: Minor, 6: Principal, 20: Not normal, 21: Critical/Normal, 34: Warning/Critical)
    - <event_status [2|3|0|1]> (0: new only, 1: validated only, 2: in process only, 3: not validated only)
    - <free_search>
    - <agent_id>
    - <pagination_size [25|50|100|200|500]>
    - <max_hours_old>
    - <id_user_ack>
    - <duplicate [0|1]> (0: all events, 1: grouped events)
    - <date_from> (format: AAAA/MM/DD)
    - <date_to> (format: AAAA/MM/DD)
    - <events_with_tags> (format:

```
["tag_id_1", "tag_id_2", "tag_id_3", "…"]
```

)

- <events_without_tags> (format:

```
["tag_id_1", "tag_id_2", "tag_id_3", "…"]
```

)

- <alert_events [0|1]> (0: filter by alert events, 1: alert events only)
- <module_id>
- <source>
- <id_extra>
- <user_comment>

Example

```
http://127.0.0.1/pandora_console/include/api.php?op=set&op2=update_event_filter&id=195&other=new_name%7C%7C%7Calert_recovered%7C%7C%7C%7C%7C%7C%7C%7C%7C%7C%7C%7C%7C%7C%7C%7C%7C&other_mode=url_encode_separator_%7C&apipass=1234&user=admin&pass=pandora
```

set delete_event_filter

It deletes an event filter.

Call syntax:

- op=set (required)
- op2=delete_event_filter (required)
- id=<event_filter_id> (required). It must be an existing event filter id.

Example

```
http://127.0.0.1/pandora_console/include/api.php?op=set&op2=delete_event_filter&id=38&apipass=1234&user=admin&pass=pandora
```

get all_event_filters

It returns the list of existing event filters.

Call syntax:

- op=get (required)
- op2=all_event_filters (required)
- other=cvs_separator (optional)

Examples

```
http://127.0.0.1/pandora_console/include/api.php?op=get&op2=all_event_filters&return_type=csv&other=;&apipass=1234&user=admin&pass=pandora
```

set create_inventory_module

It creates a new inventory module.

Call syntax:

- op=set (required)
- op2=create_inventory_module (required)
- Return_type=csv|json
- other=<serialized parameters> In the following order:
  - <name> (required)
  - <description» (required, but can be empty)
  - <operation system id» (required). Numerical Id of the table *tconfig_os*.
  - <interpreter» (required, but can be empty). If the interpreter is empty, it is interpreted as just a local inventory module.
  - < code » (required, but can be empty). *Script* that must be executed to obtain the inventory data. Must be in base64.

  - < data format » (required). Header of the retrieved data (separated by ;).
  - <block mode» (required) 1 or 0.

Example

This example creates an inventory module called OS that collects the *kernel-name* and *nodename* data from a Linux system.

```
http://localhost/pandora_console/include/api.php?op=set&op2=create_inventory_module&return_type=json&other=OS%7COS_name_description%7C1%7C/bin/bash%7CIyEvYmluL2Jhc2gKZWNobyB1bmFtZSAtbiAtcw==%7Ckernelname;nodename%7C0&other_mode=url_encode_separator_%7C&apipass=1234&user=admin&pass=pandora
```

set update_inventory_module

It updates an inventory module.

Call syntax:

- op=set (required)
- op2=update_inventory_module (required)
- id=inventory module ID (required)
- Return_type=csv|json
- other=<serialized parameters>. In the following order:
  - <name> (required)
  - <description>: (required, but it can be empty)
  - <operation system id» (required). Numerical Id of the *tconfig_os* table.
  - <interpreter>: (required, but can be empty). If the interpreter is empty, it is interpreted as just a local inventory module.
  - < code >: (required, but it can be empty). *Script* that must be executed to obtain inventory data. It must be in base64.
  - < data format >: (required) Header of the retrieved data (separated by ;).
  - <block mode>: (required) 1 or 0.

Example

```
http://localhost/pandora_console/include/api.php?op=set&op2=update_inventory_module&return_type=json&id=42&other=OS_easy%7COS_name_description%7C1%7C/bin/bash%7CIyEvYmluL2Jhc2gKZWNobyB1bmFtZSAtbiAtcw==%7Ckernelname;nodename%7C0&other_mode=url_encode_separator_%7C&apipass=1234&user=admin&pass=pandora
```

set delete_inventory_module

It deletes a user profile. It also deletes all assignments from that profile to any user.

Call syntax:

- op=set (required)
- op2=delete_inventory_module (required)
- Return_type=csv|json
- id=<inventory module ID> (required)

Examples

```
http://localhost/pandora_console/include/api.php?op=set&op2=delete_inventory_module&return_type=json&id=42&other_mode=url_encode_separator_%7C&apipass=1234&user=admin&pass=pandora
```

set create_collection

It creates a collection.

Call syntax:

- op=set (required)
- op2=create_collection (required)
- other=<serialized values> (required) Serialized values to create the collection.
- <name>
- <short_name>
- <id_group>
- <description>

It is necessary to pair it with 'other_mode' as follows: other_mode=url_encode_separator_<separator> to pass the separator of the serialized values of other.

Examples

```
http://localhost/pandora_console/include/api.php?op=set&op2=create_collection&other=test_plugin|test_p|0|test&other_mode=url_encode_separator_|&apipass=1234&user=admin&pass=pandora
```

set delete_collection

It deletes a collection.

Call syntax:

- op=set (required)
- op2=delete_collection (required)
- id=id_collection (required)

Examples

```
http://localhost/pandora_console/include/api.php?op=set&op2=delete_collection&id
=15&apipass=1234&user=admin&pass=pandora
```

set enable_disable_discovery_task

Enables or disables a task in the Discovery Task list.

Call syntax:

- op=set (required)
- op2=enable_disable_discovery_task (required)
- id=task identifier (required)
- Other=1 for disable and 0 for enable

Examples

```
http://localhost/pandora_console/include/api.php?op=set&op2=enable_disable_disco
very_task&id=1&other=1&other_mode=url_encode_separator_|&apipass=1234&user=admin
&pass=pandora
```

set create_module_policy_json

Adds a Module to the target policy by means of a definition made in JSON format.

Call syntax:

- op=set (required)
- op2=create_module_policy_json (required)
- id=policy id (required)
- other =JSON encoded base64 with the module definition. Possible fields (JSON pseudo format):

```
{
name: <value>, // mandatory
id_module: <value>, // mandatory
id_tipo_modulo: <value>, // mandatory
configuration_data: <value>,
description: <value>,
unit: <value>,
max: <value>,
min: <value>,
module_interval: <value>,
ip_target: <value>,
tcp_port: <value>,
tcp_send: <value>,
tcp_rcv: <value>,
snmp_community: <value>,
snmp_oid: <value>,
id_module_group: <value>,
flag: <value>,
disabled: <value>,
id_export: <value>,
plugin_user: <value>,
plugin_pass: <value>,
plugin_parameter: <value>,
id_plugin: <value>,
post_process: <value>,
prediction_module: <value>,
max_timeout: <value>,
max_retries: <value>,
custom_id: <value>,
history_data: <value>,
min_warning: <value>,
max_warning: <value>,
str_warning: <value>,
min_critical: <value>,
max_critical: <value>,
str_critical: <value>,
min_ff_event: <value>,
custom_string_1: <value>,
custom_string_2: <value>,
custom_string_3: <value>,
custom_integer_1: <value>,
custom_integer_2: <value>,
pending_delete: <value>,
critical_instructions: <value>,
warning_instructions: <value>,
unknown_instructions: <value>,
critical_inverse: <value>,
warning_inverse: <value>,
id_category: <value>,
module_ff_interval: <value>,
quiet: <value>,
cron_interval: <value>,
macros: <value>,
disabled_types_event: {
"going_unknwon": 1, // Disable going unknown events.
},
module_macros: <value>,
min_ff_event_normal: <value>,
min_ff_event_warning: <value>,
min_ff_event_critical: <value>,
ff_type: <value>,
each_ff: <value>,
ff_timeout: <value>,
dynamic_interval: <value>,
dynamic_max: <value>,
dynamic_min: <value>,
dynamic_next: <value>,
dynamic_two_tailed: <value>,
prediction_sample_window: <value>,
prediction_samples: <value>,
prediction_threshold: <value>,
cps: <value>,
}
```

Examples

In JSON format before converting to base64:

```
{
  "id_tipo_modulo": 1,
  "id_modulo": 10,
  "name": "sample module",
  "description": "Module created by API",
  "configuration_data": "module_begin\nmodule_name sample module\nmodule_type
generic_data\nmodule_exec echo 1\nmodule_end",
  "min_warning": 2,
  "max_warning": 5,
  "min_critical": 5,
  "max_critical": 7
}
```

Once base64 encoded:

```
http://localhost/pandora_console/include/api.php?op=set&op2=create_module_policy
_json&id=1&other=ewogICJpZF90aXBvX21vZHVsbyI6IDEsCiAgImlkX21vZHVsbyI6IDEwLAogICJ
uYW1lIjogInNhbXBsZSBtb2R1bGUiLAogICJkZXNjcmlwdGlvbiI6ICJNb2R1bGUgY3JlYXRlZCBieSB
BUEkiLAogICJjb25maWd1cmF0aW9uX2RhdGEiOiAibW9kdWxlX2JlZ2luXG5tb2R1bGVfbmFtZSBzYW1
wbGUgbW9kdWxlXG5tb2R1bGVfdHlwZSBnZW5lcmljX2RhdGFcbm1vZHVsZV9leGVjIGVjaG8gMVxubW9
kdWxlX2VuZCIsCiAgIm1pbl93YXJuaW5nIjogMiwKICAibWF4X3dhcm5pbmciOiA1LAogICJtaW5fY3J
pdGljYWwiOiA1LAogICJtYXhfY3JpdGljYWwiOiA3Cn0=&apipass=pandora&user=admin&pass=pa
ndora
```

set update_module_policy_json

Update a existing Module to the target policy by means of a definition made in JSON format.

Call syntax:

- op=set (required)
- op2=update_module_policy_json (required)
- id=policy id (required)
- other =JSON encoded base64 with the module definition. Possible fields (JSON pseudo format):

```
{
name: <value>, // mandatory
id_module: <value>, // mandatory
```

```
id_tipo_modulo: <value>, // mandatory
configuration_data: <value>,
description: <value>,
unit: <value>,
max: <value>,
min: <value>,
module_interval: <value>,
ip_target: <value>,
tcp_port: <value>,
tcp_send: <value>,
tcp_rcv: <value>,
snmp_community: <value>,
snmp_oid: <value>,
id_module_group: <value>,
flag: <value>,
disabled: <value>,
id_export: <value>,
plugin_user: <value>,
plugin_pass: <value>,
plugin_parameter: <value>,
id_plugin: <value>,
post_process: <value>,
prediction_module: <value>,
max_timeout: <value>,
max_retries: <value>,
custom_id: <value>,
history_data: <value>,
min_warning: <value>,
max_warning: <value>,
str_warning: <value>,
min_critical: <value>,
max_critical: <value>,
str_critical: <value>,
min_ff_event: <value>,
custom_string_1: <value>,
custom_string_2: <value>,
custom_string_3: <value>,
custom_integer_1: <value>,
custom_integer_2: <value>,
pending_delete: <value>,
critical_instructions: <value>,
warning_instructions: <value>,
unknown_instructions: <value>,
critical_inverse: <value>,
warning_inverse: <value>,
id_category: <value>,
module_ff_interval: <value>,
quiet: <value>,
cron_interval: <value>,
macros: <value>,
disabled_types_event: {
"going_unknwon": 1, // Disable going unknown events.
},
module_macros: <value>,
min_ff_event_normal: <value>,
min_ff_event_warning: <value>,
min_ff_event_critical: <value>,
ff_type: <value>,
each_ff: <value>,
ff_timeout: <value>,
dynamic_interval: <value>,
dynamic_max: <value>,
dynamic_min: <value>,
dynamic_next: <value>,
dynamic_two_tailed: <value>,
prediction_sample_window: <value>,
prediction_samples: <value>,
prediction_threshold: <value>,
cps: <value>,
}
```

Examples

In JSON format before converting to base64:

```
{
  "id_tipo_modulo": 1,
  "id_modulo": 1,
  "name": "name edited",
  "description": "Module created by API",
  "configuration_data": "module_begin\nmodule_name sample module\nmodule_type
generic_data\nmodule_exec echo 1\nmodule_end",
  "min_warning": 3,
  "max_warning": 6,
  "min_critical": 6,
  "max_critical": 8
}
```

Once base64 encoded:

```
http://localhost/pandora_console/include/api.php?op=set&op2=update_module_policy
_json&id=1&id2=43&other=ewogICJpZF90aXBvX21vZHVsbyI6IDEsCiAgImlkX21vZHVsbyI6IDEs
CiAgIm5hbWUiOiAibmFtZSBlZGl0ZWQiLAogICJkZXNjcmlwdGlvbiI6ICJNb2R1bGUgY3JlYXRlZCBi
eSBBUEkiLAogICJjb25maWd1cmF0aW9uX2RhdGEiOiAibW9kdWxlX2JlZ2luXG5tb2R1bGVfbmFtZSBz
YW1wbGUgbW9kdWxlXG5tb2R1bGVfdHlwZSBnZW5lcmljX2RhdGFcbm1vZHVsZV9leGVjIGVjaG8gMVxu
bW9kdWxlX2VuZCIsCiAgIm1pbl93YXJuaW5nIjogMywKICAibWF4X3dhcm5pbmciOiA2LAogICJtaW5f
Y3JpdGljYWwiOiA2LAogICJtYXhfY3JpdGljYWwiOiA4Cn0=&apipass=pandora&user=admin&pass
=pandora
```

set event_custom_id

To set a custom identifier in an event.

Call syntax:

- op=get (required).
- op2=set (required).
- id= < id_event > (required, numerical value).
- id2= < id_custom_event > (required, alphanumeric value).
- other=< id_node > If running in Command Center (Metaconsole), the node identifier. Default value 0 (Command Center).

Example (see call syntax):

```
…/include/api.php?op=set&op2=event_custom_id&id=110&id2=9999&apipass=1234&user=admin&pass=pandora
```

Example for a node in Command Center:

```
…/include/api.php?op=set&op2=event_custom_id&id=3831949&id2=Test123&other=1&apipass=1234&user=admin&pass=pandora
```

Examples

These are several examples in several languages about calling Pandora FMS API.

PHP

```
<?php
$ip = '192.168.70.110';
$pandora_url = '/pandora5';
$apipass = '1234';
$user = 'admin';
$password = 'pandora';
$op = 'get';
$op2 = 'all_agents';
$return_type = 'csv';
$other = '';
$other_mode = '';

$url = "http://" . $ip . $pandora_url . "/include/api.php";

$url .= "?";
$url .= "apipass=" . $apipass;
$url .= "&user=" . $user;
$url .= "&pass=" . $password;
$url .= "&op=" . $op;
$url .= "&op2=" . $op2;
```

```
if ($id !== '') {
    $url .= "&id=" . $id;
}
if ($id2 !== '') {
    $url .= "&id2=" . $id2;
}
if ($return_type !== '') {
    $url .= "&return_type=" . $return_type;
}
if ($other !== '') {
    $url .= "&other_mode=" . $other_mode;
    $url .= "&other=" . $other;
}

$curlObj = curl_init();
curl_setopt($curlObj, CURLOPT_URL, $url);
curl_setopt($curlObj, CURLOPT_RETURNTRANSFER, 1);
$result = curl_exec($curlObj);
curl_close($curlObj);

$agents = array();
if (!empty($result)) {
    $lines = explode("\n", $result);
    foreach ($lines as $line) {
        $fields = explode(";", $line);

        $agent = array();
        $agent['id_agent'] = $fields[0];
        $agent['name'] = $fields[1];
        $agent['ip'] = $fields[2];
        $agent['description'] = $fields[3];
        $agent['os_name'] = $fields[4];
        $agent['url_address'] = $fields[5];

        $agents[] = $agent;
    }
}

print_list_agents($agents);

function print_list_agents($agents) {
    echo "<table border='1' style='empty-cells: show;'>";

    echo "<thead>";
    echo "<tr>";
    echo "<th>" . "ID" . "</th>";
    echo "<th>" . "Name" . "</th>";
    echo "<th>" . "IP" . "</th>";
    echo "<th>" . "Description" . "</th>";
    echo "<th>" . "OS" . "</th>";
    echo "<th>" . "URL" . "</th>";
    echo "</tr>";
    echo "</thead>";

    foreach ($agents as $agent) {
        echo "<tr>";
        echo "<td>" . $agent['id_agent'] . "</td>";
        echo "<td>" . $agent['name'] . "</td>";
        echo "<td>" . $agent['ip'] . "</td>";
        echo "<td>" . $agent['description'] . "</td>";
        echo "<td>" . $agent['os_name'] . "</td>";
        echo "<td>" . $agent['url_address'] . "</td>";
        echo "</tr>";
    }
    echo "</table>";
}
?>
```

Python

```
import pycurl
import cStringIO
import pprint

def main():
    ip = '192.168.70.110'
    pandora_url = '/pandora5'
    apipass = '1234'
    user = 'admin'
    password = 'pandora'
    op = 'get'
    op2 = 'all_agents'
    return_type = 'csv'
    other = ''
    other_mode = ''

    url = "http://" + ip  + pandora_url + "/include/api.php"

    url += "?"
    url += "apipass=" + apipass
    url += "&user="  + user
    url += "&pass=" + password
    url += "&op=" + op
    url += "&op2=" + op2

    buf = cStringIO.StringIO()

    c = pycurl.Curl()
    c.setopt(c.URL, url)
    c.setopt(c.WRITEFUNCTION, buf.write)
    c.perform()

    output = buf.getvalue()
    buf.close()

    lines = output.split("\n")
    agents = []
    for line in lines:
        if not line:
            continue

        fields = line.split(";")
        agent = {}
        agent['id_agent'] = fields[0]
        agent['name'] = fields[1]
        agent['ip'] = fields[2]
        agent['description'] = fields[3]
        agent['os_name'] = fields[4]
        agent['url_address'] = fields[5]

        agents.append(agent)

    for agent in agents:
        print("---- Agent #" + agent['id_agent'] + " ----")
        print("Name: " + agent['name'])
        print("IP: " + agent['ip'])
        print("Description: " + agent['description'])
        print("OS: " + agent['os_name'])
        print("URL: " + agent['url_address'])
        print("")

if __name__ == "__main__":
    main()
```

Perl

```
use strict;
use warnings;
use WWW::Curl::Easy;

sub write_callback {
    my ($chunk,$variable) = @_;

    push @{$variable}, $chunk;
    return length($chunk);
}

my $ip = '192.168.70.110';
my $pandora_url = '/pandora5';
my $apipass = '1234';
my $user = 'admin';
my $password = 'pandora';
my $op = 'get';
my $op2 = 'all_agents';
my $return_type = 'csv';
my $other = '';
my $other_mode = '';

my $url = "http://" . $ip  . $pandora_url . "/include/api.php";
$url .= "?";
$url .= "apipass=" . $apipass;
$url .= "&user=" . $user;
$url .= "&pass=" . $password;
$url .= "&op=" . $op;
$url .= "&op2=" . $op2;

my @body;

my $curl = WWW::Curl::Easy->new;
$curl->setopt(CURLOPT_URL, $url);
$curl->setopt(CURLOPT_WRITEFUNCTION, \&write_callback);
$curl->setopt(CURLOPT_FILE, \@body);
$curl->perform();

my $body=join("",@body);
my @lines = split("\n", $body);

foreach my $line (@lines) {
    my @fields = split(';', $line);

    print("\n---- Agent #" . $fields[0] . " ----");
    print("\nName: " .  $fields[1]);
    print("\nIP: " .  $fields[2]);
    print("\nDescription: " . $fields[3]);
    print("\nOS: "  . $fields[4]);
    print("\n");
}
```

Ruby

```
require 'open-uri'

ip = '192.168.70.110'
pandora_url = '/pandora5'
apipass = '1234'
user = 'admin'
password = 'pandora'
op = 'get'
op2 = 'all_agents'
return_type = 'csv'
other = ''
other_mode = ''

url = "http://" + ip  + pandora_url + "/include/api.php"
url += "?"
url += "apipass=" + apipass
url += "&user="  + user
url += "&pass=" + password
url += "&op=" + op
url += "&op2=" + op2

agents = []

open(url) do |content|

    content.each do |line|
        agent = {}

        tokens = line.split(";")

        agent[:id_agent] = tokens[0]
        agent[:name] = tokens[1]
        agent[:ip] = tokens[2]
        agent[:description] = tokens[3]
        agent[:os_name] = tokens[4]
        agent[:url_address] = tokens[5]

        agents.push agent
    end
end

agents.each do |agent|
    print("---- Agent #" + (agent[:id_agent] || "") + " ----\n")
    print("Name: " + (agent[:name] || "") + "\n")
    print("IP: " + (agent[:ip] || "") + "\n")
    print("Description: " + (agent[:description] || "") + "\n")
    print("OS: " + (agent[:os_name] || "") + "\n")
    print("URL: " + (agent[:url_address] || "") + "\n")
    print("\n")
end
```

Lua

```

require("curl")

local content = ""

function WriteMemoryCallback(s)
    content = content .. s

    return string.len(s)
end

ip = '192.168.70.110'
pandora_url = '/pandora5'
```

```
apipass = '1234'
user = 'admin'
password = 'pandora'
op = 'get'
op2 = 'all_agents'
return_type = 'csv'
other = ''
other_mode = ''

url = "http://" .. ip  .. pandora_url .. "/include/api.php"

url = url .. "?"
url = url .. "apipass=" .. apipass
url = url .. "&user=" .. user
url = url .. "&pass=" .. password
url = url .. "&op=" .. op
url = url .. "&op2=" .. op2

if curl.new then c = curl.new() else c = curl.easy_init() end

c:setopt(curl.OPT_URL, url)
c:setopt(curl.OPT_WRITEFUNCTION, WriteMemoryCallback)

c:perform()

for line in string.gmatch(content, "[^\n]+") do
    line = string.gsub(line, "\n", "")

    count = 0
    for field in string.gmatch(line, "[^\;]+") do
        if count == 0 then
            print("---- Agent #" .. field .. " ----")
        end
        if count == 1 then
            print("Name: " .. field)
        end
        if count == 2 then
            print("IP: " .. field)
        end
        if count == 3 then
            print("Description: " .. field)
        end
        if count == 4 then
            print("OS: " .. field)
        end
        if count == 5 then
            print("URL: " .. field)
        end

        count = count + 1
    end
    print("")
```

```
end
```

Brainfuck

```
[-]>[-]<>+++++++++[<+++++++++>-]<-.>+++++[<+++++>-]<----.>++++[<++++>-]<---.
>++++[<---->-]<++.
>+++[<+++>-]<++.

-.>++++++++[<-------->-]<--.>+++[<--->-]<---.>++++++++[<++++++++>-]<++++.

+.>++++++++[<-------->-]<-----.>+++++++++[<+++++++++>-]<----.

++.
--.>+++[<--->-]<+.>+++[<+++>-]<.>++[<++>-]<++.
>++[<-->-]<-.
>+++++++++[<--------->-]<++.
>+++++++++[<+++++++++>-]<---.

+.>+++++++++[<--------->-]<++.>+++++++++[<+++++++++>-]<+++.>++++[<---->-]<+.
>+++[<+++>-]<.
>+++[<--->-]<++.
>+++[<+++>-]<-.
>+++++++++[<--------->-]<++.
>+++++++++[<+++++++++>-]<+++.
>+++[<--->-]<--.

----.>+++[<+++>-]<-.

+++.
-.>+++++++++[<--------->-]<++.>+++++++++[<+++++++++>-]<-.>++++[<---->-]<+.
>++++[<++++>-]<+.
>++++[<---->-]<-.
>++++++++[<-------->-]<-.
>++++++++[<++++++++>-]<++++++++.
>+++[<--->-]<++.

++.
++.>++++[<++++>-]<---.>++[<-->-]<--.

+++.>++++++++[<-------->-]<---.>+++[<--->-]<---.>+++++++++[<+++++++++>-]<-.
>+++[<--->-]<--.
>++++[<++++>-]<---.

---.>+++++++++[<--------->-]<++.>+++++++++[<+++++++++>-]<+++++.>+++++[<----->-
]<++++.
>+++[<+++>-]<++.
>+++[<--->-]<++.
>++++++++[<-------->-]<-----.
>+++++++++[<+++++++++>-]<----.
>+++[<+++>-]<-.
>++++[<---->-]<--.
```

```
>++[<++>-]<+.
>+++[<+++>-]<--.

++++.>+++++++++[<--------->-]<--.>++++++++[<++++++++>-]<++++++.>+++[<+++>-]<+++.
>+++[<--->-]<.

++.
--.>+++[<+++>-]<--.>++[<++>-]<+.>+++[<--->-]<++.
>++[<++>-]<++.
>++[<-->-]<-.

++++.>++++++++[<-------->-]<-----.
```

Java (Android)

See our project (Pandora FMS Event Viewer) in Pandroid FMS Event Viewer source code in SourceForge SVN repository but this is the most important piece of code, which calls the API to get event data.

```

/**
     * Performs an http get petition.
     *
     * @param context
     *            Application context.
     * @param additionalParameters
     *            Petition additional parameters
     * @return Petition result.
     * @throws IOException
     *             If there is any problem with the connection.
     */
    public static String httpGet(Context context,
            List<NameValuePair> additionalParameters) throws IOException {
        SharedPreferences preferences = context.getSharedPreferences(
                context.getString(R.string.const_string_preferences),
                Activity.MODE_PRIVATE);

        String url = preferences.getString("url", "") + "/include/api.php";
        String user = preferences.getString("user", "");
        String password = preferences.getString("password", "");
        String apiPassword = preferences.getString("api_password", "");
        if (url.length() == 0 || user.length() == 0) {
            return "";
        }
        ArrayList<NameValuePair> parameters = new ArrayList<NameValuePair>();
        parameters.add(new BasicNameValuePair("user", user));
        parameters.add(new BasicNameValuePair("pass", password));
        if (apiPassword.length()> 0) {
            parameters.add(new BasicNameValuePair("apipass", apiPassword));
        }
        parameters.addAll(additionalParameters);
        Log.i(TAG, "sent: " + url);
        if (url.toLowerCase().contains("https")) {
            // Secure connection
            return Core.httpsGet(url, parameters);
        } else {
            HttpParams params = new BasicHttpParams();
            HttpConnectionParams.setConnectionTimeout(params,
                    CONNECTION_TIMEOUT);
            HttpConnectionParams.setSoTimeout(params, CONNECTION_TIMEOUT);
            DefaultHttpClient httpClient = new DefaultHttpClient(params);
            UrlEncodedFormEntity entity;
            HttpPost httpPost;
            HttpResponse response;
            HttpEntity entityResponse;
            String return_api;
            httpPost = new HttpPost(url);
            entity = new UrlEncodedFormEntity(parameters);
            httpPost.setEntity(entity);
            response = httpClient.execute(httpPost);
            entityResponse = response.getEntity();
            return_api = Core
                    .convertStreamToString(entityResponse.getContent());
            Log.i(TAG, "received: " + return_api);
            return return_api;
        }
    }
```

New calls extension in the API

To develop new calls for the API, keep in mind that:

- The call has to be inscribed as a function in the file <installation Pandora FMS Console>/include/functions_api.php .
- The function must have this structure: The prefix `api`, the kind of operation `get`, `set` or `help` (depending on whether it is a data read, data write or retrieve help operation) and the name of the call, trying to be coherent with the operation, as for example: function api_get_[call_name](parameters) .
- The function can have no parameters, but if it has them, the parameters received will be the following in the same order:
- id: first operator or parameter, it contains a string.
- id2: second operator or parameter, it contains a string.
- other: rest of operators or parameters, it contains as an array of two positions:
    - $other['type']: That might be be a string or an array.
    - $other['data']: That will be a string with the parameter or an array of numeric index with the past parameters.
- returnType: string that specifies the kind of return that the call will have. It is usually visible for you, but you may use or modify it if necessary.

New Calls in the API from Pandora FMS extensions

It is possible to create new API calls without using …`/include/functions_api.php`. The process consists of adding into a Pandora FMS extension directory a file with the following name:

```
<extension_name>.api.php
```

and into this file create the desired functions with the same considerations of the standard API but with `apiextension` prefix instead of `api`.

For example, having an extension called `module_groups` with the path

<Pandora installation>/extensions/module_groups

You must create a file called `module_groups.api.php` into this directory.

The desired functions will be within this file, for example a function to get the number of modules in a group. This function must have a name like: `apiextension_get_groupmodules`.

Function example

In this function, made up functions have been used.

```
function apiextension_get_groupmodules($group_name) {
    $group_id = group_id_from_name($group_name);
    if($group_id == false) {
       echo 'Group doesnt exist';
       return;
    }

    $number_of_modules = group_modules($group_id);
    echo $number_of_modules;
}
```

Call example

This call example gets the number of modules of the `Servers` group.

```
http://127.0.0.1/pandora_console/include/api.php?op=get&op2=extension&ext_name=module_groups&ext_function=groupmodules&id=Servers&apipass=1234&user=admin&pass=pandora
```

API Functions

The following functions could be used in your call function code:

- returnError(typeError, returnType): It gives back an error in an standardized way for all calls.
    - typeError: By now 'id_not_found' or null.
    - returnType: By now 'string' or error message.
- returnData(returnType, data, separator): It is the function that returns the API call data.
    - returnType: That could be 'string', 'csv', 'csv_head'
    - data: It is an array that contains the data, as well as its format. It has the following fields:
        - 'type' (required): It could be 'string' and 'array'.
        - 'list_index' (optional): It contains a numeric index array containing the alphanumeric index to be taken out through exit.
        - 'data' (required): It contains a string with the data or an array of alphanumeric index or numeric index with data.

Example

```
function api_get_module_last_value($idAgentModule, $trash1, $other = ';', $returnType)
{
    $sql = sprintf('SELECT datos FROM tagente_estado WHERE id_agente_modulo = %d', $idAgentModule);
    $value = get_db_value_sql($sql);
    if ($value === false) {
        switch ($other['type']) {
            case 'string':
                switch ($other['data']) {
                    case 'error_message':
                    default:
                        returnError('id_not_found', $returnType);
                    break;
                }
                break;
            case 'array':
                switch ($other['data'][0]) {
                    case 'error_value':
                        returnData($returnType, array('type' => 'string', 'data' => $other['data'][1]));
                        break;
                }
                break;
        }
    }
    else {
        $data = array('type' => 'string', 'data' => $value);
        returnData($returnType, $data);
    }
}
```

Future of API.php

Some ideas for the future of api.php are:

- Broading the API call ensemble.
- Returning and getting values in xml, JSON...
- Increasing security call for insecure environments.
- Integration with third tool standards.

Go back to Pandora FMS documentation index