



Pandora ITSM API



From:

<https://pandorafms.com/manual/!778/>

Permanent link:

https://pandorafms.com/manual/!778/en/documentation/10_pandora_itsm/21_pandora_itsm_api

2024/12/03 19:30





Pandora ITSM API

Introduction

The external API of Pandora ITSM is used by making remote calls via HTTP or HTTPS over the file:

```
.../include/api.php
```

With this method it is possible to integrate third party applications with Pandora ITSM. The use of the API is done through a call with defined parameters that returns a value or a list of values.

The prototype API call is as follows:

```
http://< Console_web_URL >/integria/include/api.php?< parameters >
```

Where < URL >' is the IP address or public URL where Pandora ITSM is installed. The API method only accepts the following parameters (< parameters >), any other parameter will be ignored:

- **user** (required): Name of the Pandora ITSM user that accesses the data, it is used to verify [the ACL rules](#).
- **user_pass** (obligatorio): User password.
- **op** (required): Operation to be performed. The operations are detailed below.
- **pass** (*required if enabled*): API specific password, configured in the [Pandora ITSM Console](#).
- **params**: List of parameters to be passed to the function separated by a token.
- **token**: Character that separates the different parameters (by default the comma , is used).
- **return_type**: Format of the call result, can be XML or CSV (CSV by default).
- **info**: [Special parameter](#) with which the version can be obtained before authentication.

Security

Authentication by valid IP address

Security is based on a list of IP addresses that will have access to the tool. This list can be created in the [Pandora ITSM configuration menu](#).

It is important to note that this authentication method is always active, so if you wish to omit it, it will be necessary to place the asterisk * character to indicate that any IP address is authorized.

It is recommended to always use a specific list of IP addresses.

Password Authentication

Within the Pandora ITSM configuration menu it is possible to establish a specific password for the API. This password must be sent not encrypted in the API call, so you should always use HTTPS if you want to use this authentication method.

If this password is empty, it means that password authentication is disabled and it is unnecessary to put the pass token in the API call.

Authentication by user and user password

It is necessary in all calls (*except information calls*) to specify the user (`user`) and the user's password (`user_pass`).

There is no way to disable this step of authentication.

Authentication by bearer token

Version 5.0.99 OUM 99 MR 93: You can authenticate with API Token by means of sending the **HTTP headers** of a **bearer token** generated **by each user and for their own private and personal use**.

Bearer token-based authentication allows each of the users registered in Pandora FMS to generate their own identifier in unequivocal correspondence. Said 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 is up to the application making the connection to the PITSM API.

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

- Check the version of curl with `curl -V`.
- If you need more help using curl, invoke `curl -h`.
- If you need to debug the connection use the `-v` parameter (not to be confused with the `-V` used to display the version).
- The `-k` and `-H` parameters are used to accept connections with self-signed certificates for HTTPS and send header information, respectively.
- Query 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_incidents` (replace with your Console web URL and your bearer token previously queried):

```
BEARER="Authorization: Bearer 9ea069a7240411ee95580242ac100002"  
URL="http://192.168.7.117/integria/include/api.php"  
API_QUERY="?return_type=xml&user=admin&op=get_incidents"  
curl -k -H "$BEARER" $URL$API_QUERY && echo ""
```

Note that bearer tokens from different users, while quite similar, are actually unique values.

API Functions

Return of information about Pandora ITSM

This is a special function that can be used without authentication, since the information it returns can also be obtained from the login screen publicly (if Hide version token is disabled).

(syntax)

```
.../api.php?info=version
```

Output:

```
Pandora ITSM Enterprise Edition 5.0.104 Build 240802 MR97
```

get_incidents

Result: Returns a list of tickets to which the calling user has access.

Serialized Parameters:

- String filter: Show only tickets that contain that string in the title or description (not required). Empty for all tickets. The use of upper and lower case for the search is indifferent.
- Status filter: Identify the status (not required). It can take the following values:
 - 0 → All (default).
 - 1 → New.
 - 2 → Not confirmed.
 - 3 → Assigned.
 - 4 → Re-opened.
 - 5 → Pending to be closed.
 - 6 → Pending from a third person.

- 7 → Closed.
- -10 → For all except closed ones.
- Group filter: Group identifier (required). One (1) for all groups.
- Priority filter: Priority identifier (optional). It can have any of the following values:
 - 10 → Maintenance.
 - 0 → Informational.
 - 1 → Low.
 - 2 → Medium.
 - 3 → Serious.
 - 4 → Very serious.
 - -1 → Any.
- Inventory filter: Identifier of the related inventory (optional). Zero (0) for all or none.
- Owner filter: Identifier of the user to whom the incident is assigned (optional).
- Creator filter: Identifier of the user who created the incident (optional).
- Incident filter: Incident identifier (optional).

Example of call with result in CSV format:

(syntax)

```
.../api.php?user=user&user_pass=userpass&pass=api_pass&op=get_incidents&params=day,-10,1,-1,0,admin,user1
```

Example output with CSV format:

```
5,"2016-01-19 11:19:42","0000-00-00
00:00:00",incidence_title,admin,1,1,2,"2014-01-19
11:19:42",,1,0,0,,1,0,0,1,0,mail(at)example.com,,0,"2014-01-20
14:45:01",,,,0,0,0,0,0,0,0
```

Example of call with result in XML:

(syntax)

```
.../api.php?user=user&user_pass=userpass&pass=api_pass&op=get_incidents&params=day,-10,1,-1,0,admin,user1&return_type=xml
```

Example output in XML:

```
<xml version='1.0' encoding='UTF-8'>
  <incident>
    <id_incidencia><![CDATA[5]]></id_incidencia>
    <inicio><![CDATA[2016-01-19 11:19:42]]></inicio>
    <cierre><![CDATA[0000-00-00 00:00:00]]></cierre>
    <titulo><![CDATA[Titulo incidencia]]></titulo>
    <descripcion><![CDATA[Descripcion]]></descripcion>
    <id_usuario><![CDATA[admin]]></id_usuario>
    <estado><![CDATA[1]]></estado>
    <prioridad><![CDATA[1]]></prioridad>
    <id_grupo><![CDATA[2]]></id_grupo>
```

```

<actualizacion><![CDATA[2016-01-19 11:19:42]]></actualizacion>
<id_creator><![CDATA[]]></id_creator>
<notify_email><![CDATA[1]]></notify_email>
<id_task><![CDATA[0]]></id_task>
<resolution><![CDATA[0]]></resolution>
<epilog><![CDATA[]]></epilog>
<id_parent><![CDATA[1]]></id_parent>
<sla_disabled><![CDATA[0]]></sla_disabled>
<affected_sla_id><![CDATA[0]]></affected_sla_id>
<id_incident_type><![CDATA[1]]></id_incident_type>
<score><![CDATA[0]]></score>
<email_copy><![CDATA[copyto@example.com]]></email_copy>
<editor><![CDATA[]]></editor>
<id_group_creator><![CDATA[0]]></id_group_creator>
<last_stat_check><![CDATA[2016-01-20 14:55:01]]></last_stat_check>
<closed_by><![CDATA[]]></closed_by>
<extra_data><![CDATA[]]></extra_data>
<extra_data2><![CDATA[]]></extra_data2>
<blocked><![CDATA[0]]></blocked>
<old_status><![CDATA[0]]></old_status>
<old_resolution><![CDATA[0]]></old_resolution>
<old_status2><![CDATA[0]]></old_status2>
<old_resolution2><![CDATA[0]]></old_resolution2>
<workunits_hours><![CDATA[0]]></workunits_hours>
<workunits_count><![CDATA[0]]></workunits_count>
</incident>
</xml>

```

get_incident_details

Result: Returns the details of a ticket.

Parameters:

- Ticket identifier (required).

Example of call with result in CSV format:

(syntax)

```

.../api.php?user=user&user_pass=userpass&pass=api_pass&op=get_incident_details&params=2

```

Example output in format CSV:

```

1,"2010-02-17 01:35:35","0000-00-00 00:00:00",Myfddfe,,demo,1,1,3,3,"2010-02-17
01:50:05",user,1,0,0,,0,0,0,0

```

Example of call with result in XML:

(syntax)

```
.../api.php?user=user&user_pass=userpass&pass=api_pass&op=get_incident_details&params=2&return_type=xml
```

Example output in XML:

```
<xml>
  <id_issue>1</id_issue>
  <start>2010-02-17 01:35:35</start>
  <close>0000-00-00 00:00:00</close>
  <title>Myfddfe</title>
  <description></description>
  <user_id>demo</user_id>
  <source>1</source>
  <state>1</state>
  <priority>3</priority>
  <group_id>3</group_id>
  <update>2010-02-17 01:50:05</update>
  <id_creator>admin</id_creator>
  <notify_email>1</notify_email>
  <id_task>0</id_task>
  <resolution>0</resolution>
  <epilg></epilg>
  <id_parent></id_parent>
  <sla_disabled>0</sla_disabled>
  <affected_sla_id>0</affected_sla_id>
  <id_incident_type>0</id_incident_type>
  <score>0</score>
</xml>
```

create_incident

Result: A ticket is created.

User: It is the creator of the ticket.

Parameters:

- Title (required). Empty titles are not allowed. To introduce commas in this parameter, you must replace them with (: : :), to introduce a space you must use %20 . For example, to enter Doe, John you would insert in the URL: Doe : : :%20John .
- Group identifier (required).
- Priority (not required). It can have these values:
 - 10 → 0 Maintenance.
 - 0 → 1 Informative (informative, by default).
 - 1 → 2 Low (low priority).
 - 2 → 3 Medium (medium priority).
 - 3 → 4 Serious (serious priority).

- 4 → 5 Very serious (very serious priority).
- Description (not required). Empty by default. To introduce commas in this parameter, you must replace them with (: : :), to introduce a space you must use %20 . For example, to enter Doe, John you would insert in the URL: Doe : : :%20John .
- Inventory identifier (not required). Issues can be created with several inventory objects and they will have to be separated from each other by : .
- Incident type identifier (not required). Zero 0 (none) by default.
- Email in copy (not required): Empty default value.
- Owner (not required). By default admin.
- Parent ticket identifier (not required). Default zero 0 (no parent).
- Status (required). It can take the following values:
 - 1 → New (default).
 - 2 → Not confirmed.
 - 3 → Assigned.
 - 4 → Re-opened.
 - 5 → Pending to be closed.
 - 6 → Earring from a third person.
 - 7 → Closed.
- Info_extra (not required).
- Resolution:
 - 0 → None (none, by default).
 - 1 → Fixed.
 - 2 → Invalid.
 - 3 → Won't fix.
 - 4 → Duplicate.
 - 5 → Works for me.
 - 6 → Incomplete.
 - 7 → Expired.
 - 8 → Moved.
 - 9 → In process (in process).
- Info_extra2 (not required).
- Info_extra3 (not required).
- Info_extra4 (not required).
- CampoTipoIncidencia1.
- CampoTipoIncidencia2
- CampoTipoIncidencia3
- ...
- CampoTipoIncidenciaN.

Example call with output in CSV format:

(syntax)

```
.../api.php?user=user&user_pass=userpass&pass=api_pass&op=create_incident&params=Title,2,1,Description%20de%20la%20incidencia,1:2:3,1,copyto(at)example.com,admin,0,1
```

Example of result in CSV format:

```
0
```

Example call with result in XML:

(syntax)

```
.../api.php?user=user&user_pass=userpass&pass=api_pass&op=create_incident&params=Titulo%20de%20la%20incidencia,2,1,Descripcion%20de%20la%20incidencia,1:2:3,1,copyto(at)example.com,admin,0&return_type=xml
```

Example output in XML:

```
<xml>
0
</xml>
```

update_incident

Result: Updates the data of a ticket.

Parameters:

- Ticket identifier (required).
- Title (required). Empty titles are not allowed.
- Description (not required). Empty by default.
- Epilogue (not required). Empty by default.
- Group identifier (required).
- Priority (not required). It can have these values:
 - 10 → 0 Maintenance.
 - 0 → 1 Informative (informative, by default).
 - 1 → 2 Low (low priority).
 - 2 → 3 Medium (medium priority).
 - 3 → 4 Serious (serious priority).
 - 4 → 5 Veryserious (very serious priority).
- Resolution:
 - 0 → None (none, by default).
 - 1 → Fixed.
 - 2 → Invalid.
 - 3 → Won't fix.
 - 4 → Duplicate.
 - 5 → Works for me.
 - 6 → Incomplete.
 - 7 → Expired.
 - 8 → Moved.
 - 9 → In process (in process).
- Status (required). It can take the following values:
 - 1 → New (default).
 - 2 → Not confirmed.
 - 3 → Assigned.
 - 4 → Re-opened.
 - 5 → Pending to be closed.
 - 6 → Earning from a third person.
 - 7 → Closed.
- Owner (not required). By default admin.

- Parent ticket identifier (not required). Default zero 0 (no parent).
- Incident type identifier (not required). Zero 0 (none) by default.
- Info_extra (not required).
- Info_extra2 (not required).
- Info_extra3 (not required).
- Info_extra4 (not required).
- CampoTipoIncidencia1.
- CampoTipoIncidencia2
- CampoTipoIncidencia3
- ...
- CampoTipoIncidenciaN.

If you do not want to modify the current value of the ticket type fields, you must fill in those fields with their corresponding value. If they are empty, they will be updated with no value.

Example of call with result in CSV format:

(syntax)

```
.../api.php?user=user&pass=api_pass&op=update_incident&params=4,nuevo%20titulo,nueva%20descripcion,nuevo%20epilogo,4,2,1,3,admin,0,6,,Soporte,Hardware
```

Example of call with result in XML::

(syntax)

```
.../api.php?user=user&pass=api_pass&op=update_incident&params=4,nuevo%20titulo,nueva%20descripcion,nuevo%20epilogo,4,2,1,1,3,admin,0,6,Soporte,Hardware&return_type=xml
```

Example result in XML:

```
<xml>
0
</xml>
```

add_inventory_to_ticket

Result: Allows you to associate an inventory object to a ticket.

Parameters:

- Ticket identifier (required).
- Inventory item identifier (required).

Example of call with result in CSV format:

(syntax)

```
.../api.php?user=user&user_pass=userpass&pass=api_pass&op=add_inventory_to_ticket&params=1,2
```

Example output in CSV:

```
1
```

Example of call with result in XML:

(syntax)

```
.../api.php?user=user&user_pass=userpass&pass=api_pass&op=add_inventory_to_ticket&params=1,2&return_type=xml
```

Example output in XML:

```
<xml>
1
</xml>
```

delete_incident

Result: A ticket is deleted.

Parameters:

- Ticket identifier (required).

Example of call with result in CSV format:

(syntax)

```
.../api.php?user=user&pass=api_pass&op=delete_incident&params=3
```

Example of output in CSV format:

```
0
```

Example of call with result in XML:

(syntax)

```
.../api.php?user=user&pass=api_pass&op=delete_incident&params=3&return_type=xml
```

Example result in XML:

```
<xml>
0
</xml>
```

get_incident_workunits

Result: Returns a list with the Workunits associated with a ticket.

Parameters:

- Ticket identifier (required).

Example of call with result in CSV format:

([syntax](#))

```
.../api.php?user=user&user_pass=user_pass&pass=api_pass&op=get_incident_workunits&params=3
```

Example output with CSV format:

```
3,"2010-02-17 01:48:48",0.25,admin,"Workunit description",0,0,,1
2,"2010-02-17 01:47:41",0.25,admin,"Another workunit description",0,0,,1
```

Example of call with result in XML:

([syntax](#))

```
.../api.php?user=user&user_pass=user_pass&pass=api_pass&op=get_incident_workunits&params=3&return_type=xml
```

Example output in XML:

```
<xml>
  <workunit>
    <id>3</id>
    <timestamp>2010-02-17 01:48:48</timestamp>
    <duration>0.25</duration>
    <id_user>admin</id_user>
    <description>Workunit description</description>
    <have_cost>0</have_cost>
    <id_profile>0</id_profile>
    <locked></locked>
```

```

        <public>1</public>
    </workunit>
    <workunit>
        <id>2</id>
        <timestamp>2010-02-17 01:47:41</timestamp>
        <duration>0.25</duration>
        <id_user>admin</id_user>
        <description>Another workunit description</description>
        <have_cost>0</have_cost>
        <id_profile>0</id_profile>
        <locked></locked>
        <public>1</public>
    </workunit>
</xml>

```

create_workunit

Result: Create a Workunit in a ticket.

Parameters:

- Ticket identifier (required).
- Description (not required). By default, empty.
- Duration in hours (not required). By default 0.00 hours.
- If there has been a cost (not required). By default, no.
- If it is public or not (not required). By default, no.
- Associated profile identifier (not required). By default zero 0, with no associated profile.

Example of call with result in CSV format:

(syntax)

```

.../api.php?user=user&user_pass=userpass&pass=api_pass&op=create_workunit&params=1
,Descripcion,0.5,0,1,1

```

Example output with CSV format:

```
0
```

Example of call with result in XML:

(syntax)

```

.../api.php?user=user&user_pass=userpass&pass=api_pass&op=create_workunit&params=1
,Descripcion,0.5,0,1,1&return_type=xml

```

Example output in XML:

```
<xml>
0
</xml>
```

get_incident_files

Result: Returns a list with the files associated with a ticket.

Parameters:

- Ticket identifier (required).

Example of call with result in CSV format:

(syntax)

```
.../api.php?user=user&user_pass=userpass=&pass=api_pass&op=get_incident_files&params=1
```

Example output with CSV format:

```
1,1,0,0,admin,file1.php,"php example",764026
2,1,0,0,admin,file2.png,"png example",34026
```

Example of call with result in XML:

(syntax)

```
.../api.php?user=user&user_pass=userpass=&pass=api_pass&op=get_incident_files&params=1&return_type=xml
```

Example output in XML:

```
<xml>
  <file>
    <id_attachment>1</id_attachment>
    <id_incidencia>1</id_incidencia>
    <id_task>0</id_task>
    <id_kb>0</id_kb>
    <id_usuario>admin</id_usuario>
    <filename>file1.php</filename>
    <description>php example</description>
    <size>764026</size>
  </file>
  <file>
    <id_attachment>2</id_attachment>
    <id_incidencia>1</id_incidencia>
    <id_task>0</id_task>
```

```
<id_kb>0</id_kb>
<id_usuario>admin</id_usuario>
<filename>file2.png</filename>
<description>png example</description>
<size>34026</size>
</file>
</xml>
```

download_file

Result: Returns the content of a file as text, in Base64.

Parameters:

- File identifier (required).

Example of call with result in CSV format:

(syntax)

```
.../api.php?user=user&user_pass=userpass&pass=api_pass&op=download_file&params=32
```

Example output with CSV format:

```
[base64 file content]
```

Example of call with result in XML:

(syntax)

```
.../api.php?user=user&user_pass=userpass&pass=api_pass&op=get_incidents&params=day
,-10,1&return_type=xml
```

Example output in XML:

```
<xml>
[base64 file content]
</xml>
```

attach_file

Result: Adds a file to the file list of a ticket.

Parameters:

- Ticket identifier: (not required), zero 0 by default.
- File name: (not required).

- Size of the file in Bytes: (not required), by default zero 0 .
- File description: (not required), empty by default.
- Content of the file in Base64 (not required) by default it is created with empty content.

Example of call with result in CSV format:

(syntax)

```
.../api.php?user=user&user_pass=userpass&pass=api_pass&op=attach_file&params=1,example.txt,234,File  
%20of%20example,aw50ZWdyawEgcm9ja3M=
```

Example output with CSV format:

```
0
```

Example of call with result in XML:

(syntax)

```
.../api.php?user=user&user_pass=userpass&pass=api_pass&op=attach_file&params=1,example.txt,234,File  
%20of%20example,aw50ZWdyawEgcm9ja3M=&return_type=xml
```

Example output in XML:

```
<xml>  
0  
</xml>
```

delete_file

Result: Removes a file from a ticket's file list.

Parameters:

- File identifier (required).

Example of call with result in CSV format:

(syntax)

```
.../api.php?user=user&user_pass=userpass&pass=api_pass&op=delete_file&params=32
```

Example output with CSV format:

```
0
```

Example of call with result in XML:

(syntax)

```
.../api.php?user=user&user_pass=userpass&pass=api_pass&op=delete_file&params=32&return_type=xml
```

Example output in XML:

```
<xml>
0
</xml>
```

create_user

Result: Call used to create a user and assign them to a role or group. To assign more profiles you have to use a different call. Returns one 1 if created successfully, zero 0 otherwise.

Parameters:

- Username (required). To introduce commas in this parameter, it must be replaced by (: : :), to introduce a space, use %20 . For example, to enter Doe , John you would insert in the URL: Doe : : :%20John .
- Group identifier (required).
- Profile identifier (required).
- Email (not required). By default empty.
- Password (not required). User password, by default is the username.
- Description (not required). By default empty. To introduce commas in this parameter, it must be replaced by (: : :), to introduce a space, use %20 . For example, to enter Doe , John you would insert in the URL: Doe : : :%20John .
- Real name (not required). By default empty.
- Externaluser (not required). By default zero 0.

Example of call with result in CSV format:

(syntax)

```
.../api.php?user=user&user_pass=userpass&pass=api_pass&op=create_user&params=jorge,3,4,jorge(at)lux.com,userpass,description,Jorge%20Surname,1
```

Example output with CSV format:

```
1
```

Example of call with result in XML:

(syntax)

```
.../api.php?user=user&user_pass=userpass&pass=api_pass&op=create_user&params=jorge,3,4,jorge(at)lux.com,userpass,description,Jorge%20LastName,1&return_type=xml
```

Example output in XML:

```
<xml>
1
</xml>
```

validate_user

Result: Returns one 1 if authentication is successful, zero 0 otherwise.

Parameters:

- User: Name of the registered user to be validated (required).
- Pass: User password (required).

Example of call with result in CSV format:

(syntax)

```
.../api.php?&user=user&user_pass=userpass&pass=api_pass&op=validate_user&params=jorge,jorge_pass
```

Example output with CSV format:

```
1
```

Example of call with result in XML:

(syntax)

```
.../api.php?&user=user&user_pass=userpass&pass=api_pass&op=validate_user&params=jorge,jorge_pass&return_type=xml
```

Example output in XML:

```
<xml>
1
</xml>
```

delete_user

Result: Call used to delete a user. Returns one 1 if cleared successfully, zero 0 otherwise.

admin cannot be deleted via the API.

parameters

- User: (required).

Example of call with result in CSV format:

(syntax)

```
.../api.php?user=user&user_pass=userpass&pass=api_pass&op=delete_user&params=pepe
```

Example output with CSV format:

```
1
```

Example of call with result in XML:

(syntax)

```
.../api.php?user=user&user_pass=userpass&pass=api_pass&op=delete_user&params=pepe&return_type=xml
```

Example output in XML:

```
<xml>
1
</xml>
```

user_exists

Result: Call used to check if a user exists. Returns one 1 if it exists, zero 0 if it does not exist.

Parameters:

- User: (required).

Example of call with result in CSV format:

(syntax)

```
.../api.php?user=user&user_pass=userpass&pass=api_pass&op=user_exists&params=pepe
```

Example output with CSV format:

```
1
```

Example of call with result in XML:

(syntax)

```
.../api.php?user=user&user_pass=userpass&pass=api_pass&op=user_exists&params=pepe&return_type=xml
```

Example output in XML:

```
<xml>
1
</xml>
```

create_lead

Result: Call used to create a **lead** . Returns one 1 if created successfully, returns zero 0 otherwise.

Parameters:

- Full name (required). To introduce commas in this parameter, you must replace them with (: : :), to introduce a space you must use %20 . For example, to enter Doe , John you would insert in the URL: Doe : : :%20John .
- Company (required).
- Email (required).
- Country (not required). By default empty.
- Estimated sale (not required). Zero 0 by default.
- Progress (in percent, not required). By default 0%.
- Telephone (not required). By default, empty.
- Mobile phone (not required). By default, empty.
- Position (not required). By default empty.
- Owner (required). User name.
- Language: Language code (required).
 - de → Deutch (German).
 - en_GB → English.
 - es → Spanish.
 - fr → Français (French).
 - pl → Polski (Polish).
 - ru → Русский (Russian).
 - zh_CN → 简化字 (Chinese).
- Comments (not required). To introduce commas in this parameter, you must replace them with (: : :), to introduce a space you must use %20 . For example, to enter Doe , John you would insert in the URL: Doe : : :%20John .

- Category identifier (not required). By default zero 0 (none).
- Directed by (not required). By default zero 0.
- Campaign (not required). By default zero 0.
- Label or tag separator (required if several tags are going to be added).
- Tag names: Split with the previously declared separator. If the tags do not exist, they will not be stored.

Example of call with result in CSV format:

```
.../api.php?user=user&user_pass=userpass&pass=api_pass&op=create_lead&params=Dario%Rodriguez%Perez,Artica+ST,drp(at)example.com,Spain,230,0,55566677,6665577,Developer,admin,es,This%20comment%20is%20very%20important,1,0,7,-%7C-,Software-%7C-Hardware-%7C-Tag%20con%20espacios
```

Example output with CSV format:

```
1
```

Example of call with result in XML:

(syntax)

```
.../api.php?user=user&user_pass=userpass&pass=api_pass&op=create_lead&params=Dario%Rodriguez%Perez,Artica+ST,drp(at)example.com,Spain,230,0,55566677,6665577,Developer,admin,es,This%20comment%20is%20very%20important,1,0,7,-%7C-,Software-%7C-Hardware-%7C-Tag%20con%20espacios&return_type=xml
```

Example output in XML:

```
<xml>
1
</xml>
```

get_last_invoice_id

Result: Call used to get the last invoice identifier. Returns the last invoice identifier, empty if unsuccessful.

Parameters: none.

Example of call with result in CSV format:

(syntax)

```
.../api.php?user=user&user_pass=userpass&pass=api_pass&op=get_last_invoice_id
```

Example output with CSV format:

```
140601_0082
```

Example of call with result in XML:

(syntax)

```
.../api.php?user=user&user_pass=userpass&pass=api_pass&op=get_last_invoice_id&return_type=xml
```

Example output in XML:

```
<xml>
  <last_invoice_id>
    <id><![CDATA[140601_0082]]></id>
  </last_invoice_id>
</xml>
```

get_invoice

Result: Call to obtain the information of an invoice. Returns the invoice information, empty otherwise.

Parameters:

- Invoice identifier: It is the value of the Bill ID field (required).

Example of call with result in CSV format:

(syntax)

```
.../api.php?user=user&user_pass=userpass&pass=api_pass&op=get_invoice&params=140601_0071
```

Example output with CSV format:

```
2,admin,,1,140601_0071,ads,,,,,222.00,0.00,0.00,0.00,0.00,0.00,EUR,,0,0,,2014-01-24,0000-00-00,pending,,,Submitted,en_GB
```

Example of call with result in XML:

(syntax)

```
.../api.php?user=user&user_pass=userpass&pass=api_pass&op=get_invoice&params=140601_0071&return_type=xml
```

Example output in XML:

```
<xml>
  <invoice>
    <id><![CDATA[2]]></id>
    <id_user><![CDATA[admin]]></id_user>
    <id_task><![CDATA[]]></id_task>
    <id_company><![CDATA[1]]></id_company>
    <bill_id><![CDATA[140601_0071]]></bill_id>
    <concept1><![CDATA[ads]]></concept1>
    <concept2><![CDATA[]]></concept2>
    <concept3><![CDATA[]]></concept3>
    <concept4><![CDATA[]]></concept4>
    <concept5><![CDATA[]]></concept5>
    <amount1><![CDATA[222.00]]></amount1>
    <amount2><![CDATA[0.00]]></amount2>
    <amount3><![CDATA[0.00]]></amount3>
    <amount4><![CDATA[0.00]]></amount4>
    <amount5><![CDATA[0.00]]></amount5>
    <tax><![CDATA[0.00]]></tax>
    <currency><![CDATA[EUR]]></currency>
    <description><![CDATA[]]></description>
    <id_attachment><![CDATA[0]]></id_attachment>
  <locked><![CDATA[0]]></locked>
  <locked_id_user><![CDATA[]]></locked_id_user>
  <invoice_create_date><![CDATA[2014-01-24]]></invoice_create_date>
  <invoice_payment_date><![CDATA[0000-00-00]]></invoice_payment_date>
  <status><![CDATA[pending]]></status>
  <reference><![CDATA[]]></reference>
  <internal_note><![CDATA[]]></internal_note>
  <invoice_type><![CDATA[Submitted]]></invoice_type>
  <id_language><![CDATA[en_GB]]></id_language>
</invoice>
</xml>
```

create_invoice

Result: Call to create an invoice. It returns two values, the status code (one 1 if the operation was performed, zero 0 if a failure was detected) and a message with the result of the operation.

Parameters:

- User identifier (must exist or empty).
- Task identifier (not required). Zero 0 by default.
- Company identifier (required).
- Invoice identifier (required).
- Concept (required). To introduce commas in this parameter, you must replace them with (: : :), to introduce a space you must use %20 . For example, to enter Doe, John will have to be inserted in the URL: Doe : : :%20John .
- Quantity (required).

- Taxes: Tax percentage (not required). By default zero 0.
- Currency: Text string (required).
- Description: (not required). By default empty. To introduce commas in this parameter, you must replace them with (: : :), to introduce a space you must use %20 . For example, to enter Doe, John you would insert in the URL: Doe : : :%20John .
- Blocked flag: Boolean (not required). Default zero 0 (not locked).
- Blocker user identifier (required if the blocked flag is active). By default empty.
- Creation date (not required). By default the current day.
- Date of payment: (not required). By default the current day.
- State (not required). By default pending.
- Reference (not required). By default empty.
- Internal note (not required). By default empty.
- Type of invoice (required: Submitted or Received).
- Language identifier (not required).
 - de → Deutch (German).
 - en_GB → English.
 - es → Spanish.
 - fr → Français (French).
 - pl → Polski (Polish).
 - ru → Русский (Russian).
 - zh_CN → 简化字 (Chinese).

Example of call without errors with result in CSV format:

(syntax)

```
.../api.php?user=user&user_pass=userpass&pass=api_pass&pass=1234&op=create_invoice
&params=admin,,1,140601_0082,paso,3.00,,EUR,,,,,paid,,,Submitted,
```

Example output without errors with CSV format:

```
1,"invoice created"
```

Example output with errors with CSV format:

```
0,"invalid billing id"
```

Example of call with result in XML:

(syntax)

```
.../api.php?user=user&user_pass=userpass&pass=api_pass&op=create_invoice&params=ad
min,,1,140601_0082,paso,3.00,,EUR,,,,,paid,,,Submitted,&return_type=xml
```

Example output without errors in XML:

```
<xml>
  <invoice>
    <status><![CDATA[1]]></status>
```

```
<error><![CDATA[invoice created]]></error>
</invoice>
</xml>
```

Example output with errors in XML:

```
<xml>
<invoice>
  <status><![CDATA[0]]></status>
  <error><![CDATA[invalid billing id]]></error>
</invoice>
</xml>
```

create_company

Result: Returns the identifier of the company created.

Parameters:

- Name (required). To introduce commas in this parameter, you must replace them with (: : :), to introduce a space you must use %20 . For example, to enter Doe, John you would insert in the URL: Doe : : :%20John .
- Address (not required). By default empty. To introduce commas in this parameter, you must replace them with (: : :), to introduce a space you must use %20 . For example, to enter Doe, John you would insert in the URL: Doe : : :%20John .
- Tax identification (not required). By default empty.
- Company role identifier (not required). By default zero 0 (no role).
- Country (not required). By default, empty.
- Manager: Name of the user who is in charge of the company (it must exist or put it empty).
- Identifier of the parent company (not required). By default zero 0.

Example of call with result in CSV format:

(syntax)

```
.../api.php?user=user&user_pass=userpass&pass=api_pass&op=create_company&params=Ar
tica,AlbertoAguilera,12312312,1,Country,admin,0
```

Example output with CSV format:

6

Example of call with result in XML:

(syntax)

```
.../api.php?user=user&user_pass=userpass&pass=api_pass&op=create_company&params=Ar
tica,AlbertoAguilera,12312312,1,
```

```
Country,admin,0&return_type=xml
```

Example output in XML:

```
<xml>
  <company>
    <id><![CDATA[6]]></id>
  </company>
</xml>
```

add_address_to_newsletter

Result: Returns the identifier of the email address created. If it fails, it returns a zero value 0.

Parameters:

- Id newsletter (required and must exist).
- Name (required).
- Email (required).

Example of call with result in CSV format:

(syntax)

```
.../api.php?user=user&user_pass=userpass&pass=yourpassword&op=add_address_to_newsletter&params=2,juan,juan(at)mail.com
```

Result

```
10
```

Example of call with result in XML:

(syntax)

```
.../api.php?user=user&user_pass=userpass&pass=yourpassword&op=add_address_to_newsletter&params=2,juan,juan(at)mail.com&return_type=xml
```

Example output in XML:

```
<xml>
10
</xml>
```

get_incident_tracking

Result: Returns a list with the tracking of a ticket.

Parameters:

- Ticket identifier (required).

Example of call with result in CSV format:

(syntax)

```
.../api.php?user=user&user_pass=userpass&pass=api_pass&op=get_incident_tracking&params=2
```

Example output with CSV format:

```
9,2,0,"2010-02-17 01:56:33",demo,0,Created
10,2,2,"2010-02-17 02:06:58",demo,0,"Workunit added"
```

Example of call with result in XML:

(syntax)

```
.../api.php?user=user&user_pass=userpass&pass=api_pass&op=get_incident_tracking&params=2&return_type=xml
```

Example output in XML:

```
<xml>
  <tracking>
    <id_it>9</id_it>
    <id_incident>2</id_incident>
    <state>0</state>
    <timestamp>2010-02-17 01:56:33</timestamp>
    <id_user>demo</id_user>
    <id_additional>0</id_additional>
    <description>Created</description>
  </tracking>
  <tracking>
    <id_it>10</id_it>
    <id_incident>2</id_incident>
    <state>2</state>
    <timestamp>2010-02-17 02:06:58</timestamp>
    <id_user>demo</id_user>
    <id_additional>0</id_additional>
    <description>Workunit added</description>
  </tracking>
</xml>
```

get_incidents_resolutions

Result: Returns a list with the identifiers and names of the Pandora ITSM ticket resolutions.

Parameters: None.

Example of call with result in CSV format:

(syntax)

```
.../api.php?user=user&user_pass=userpass&pass=api_pass&op=get_incidents_resolutions
```

Example output with CSV format:

```
1,Fixed  
2,Invalid  
3,"Won't fix"
```

Example of call with result in XML:

(syntax)

```
.../api.php?user=user&user_pass=userpass&pass=api_pass&op=get_incidents_resolutions&return_type=xml
```

Example output in XML:

```
<xml>  
  <resolution>  
    <id>1</id>  
    <name>Fixed</name>  
  </resolution>  
  <resolution>  
    <id>2</id>  
    <name>Invalid</name>  
  </resolution>  
  <resolution>  
    <id>3</id>  
    <name>Wont fix</name>  
  </resolution>  
</xml>
```

get_incidents_status

Result: Returns a list with the identifiers and names of the Pandora ITSM ticket statuses.

Parameters: None.

Example of call with result in CSV format:

(syntax)

```
.../api.php?user=user&user_pass=userpass&pass=api_pass&op=get_incidents_status
```

Example output with CSV format:

```
1,New  
2,Unconfirmed  
3,Assigned
```

Example of call with result in XML:

(syntax)

```
.../api.php?user=user&user_pass=userpass&pass=api_pass&op=get_incidents_status&return_type=xml
```

Example output in XML:

```
<xml>  
  <status>  
    <id>1</id>  
    <name>New</name>  
  </status>  
  <status>  
    <id>2</id>  
    <name>Unconfirmed</name>  
  </status>  
  <status>  
    <id>3</id>  
    <name>Assigned</name>  
  </status>  
</xml>
```

get_incident_priorities

Result: Returns a list with the identifiers and names of the Pandora ITSM ticket priorities.

Parameters: None.

Example of call with result in CSV format:

(syntax)

```
.../api.php?user=user&user_pass=userpass&pass=api_pass&op=get_incident_priorities
```

Example output with CSV format:

```
10,Maintenance
0,Informative
1,Low
2 medium
3, Serious
4, very serious
```

Example of call with result in XML:

(syntax)

```
.../api.php?user=user&user_pass=userpass&pass=api_pass&op=get_incident_priorities&
return_type=xml
```

Example output in XML:

```
<xml>
  <priority>
    <id>10</id>
    <name>Maintenance</name>
  </priority>
  <priority>
    <id>0</id>
    <name>Informational</name>
  </priority>
  <priority>
    <id>1</id>
    <name>Low</name>
  </priority>
  <priority>
    <id>2</id>
    <name>Middle</name>
  </priority>
  <priority>
    <id>3</id>
    <name>Severe</name>
  </priority>
  <priority>
    <id>4</id>
    <name>Very serious</name>
  </priority>
</xml>
```

get_incident_fields

Result: Returns the custom fields for a given ticket type. If a ticket identifier is also specified, their values will be returned.

Parameters:

- Ticket type identifier (required).
- Ticket identifier (optional).

Example of call with result in CSV format:

(syntax)

```
.../api.php?user=user&user_pass=userpass&pass=api_pass&op=get_incident_fields&params=4.32
```

Example output with shape to CSV:

```
chJvYmFuZG8mI3gyMDtsaXN0YQ==,8,8,4,4,"testing list","testing list",text,text,,,1,1,,,0,0,,,1,1,"content test", 0
```

Example of call with result in XML:

(syntax)

```
.../api.php?user=user&user_pass=userpass&pass=api_pass&op=get_incident_fields&params=4.32&return_type=xml
```

Example output in XML:

```
<xml>
  <field>
    <label_enco>chJvYmFuZG8mI3gyMDtsaXN0YQ==</label_enco>
    <id>8</id>
    <id_incident_type>4</id_incident_type>
    <label>testing list</label>
    <type>text</type>
    <combo_value></combo_value>
    <show_in_list>1</show_in_list>
    <parent>0</parent>
    <linked_value></linked_value>
    <order>1</order>
    <data>test content</data>
    <blocked>0</blocked>
  </field>
</xml>
```

get_groups

Result: Returns a list with the identifiers and names of the Pandora ITSM groups.

Parameters: None.

Example of call with result in CSV format:

(syntax)

```
.../api.php?user=user&user_pass=userpass&pass=api_pass&op=get_groups
```

Example output with CSV format:

```
2,"Customer #A"  
3,"Customer #B"  
4,Engineering
```

Example of call with result in XML:

(syntax)

```
.../api.php?user=user&user_pass=userpass&pass=api_pass&op=get_groups&return_type=x  
ml
```

Example output in XML:

```
<xml>  
  <group>  
    <id>1</id>  
    <name>"Customer #A"</name>  
  </group>  
  <group>  
    <id>2</id>  
    <name>"Customer #B"</name>  
  </group>  
  <group>  
    <id>3</id>  
    <name>Engineering</name>  
  </group>  
</xml>
```

get_users

Result: Returns a list with the users over which the user making the call has visibility.

Parameters: None.

Example of call with result in CSV format:

(syntax)

```
.../api.php?user=user&user_pass=userpass&pass=api_pass&op=get_users
```

Example output with CSV format:

```
user1  
user2  
user3
```

Example of call with result in XML:

(syntax)

```
.../api.php?user=user&user_pass=userpass&pass=api_pass&op=get_incidents&params=day  
, -10, 1&return_type=xml
```

Example output in XML:

```
<xml>  
  <id_user>user1</id_user>  
  <id_user>user2</id_user>  
  <id_user>user3</id_user>  
</xml>
```

get_stats

Result: Returns a value corresponding to the query performed.

Parameters:

- **metric:** Metric of statistical values, can be the following:
 - **total_incidents:** Total number of tickets (limited by the maximum number of Pandora ITSM tickets).
 - **opened:** Total number of open tickets.
 - **closed:** Total number of closed tickets.
 - **avg_life:** Average life of the tickets in seconds.
 - **sla_compliance:** SLA compliance percentage.
 - **avg_scoring:** Average ticket score from 0 to 10.
 - **avg_worktime:** Average work time on the ticket in seconds.
 - **string:** String to filter the ticket search.
 - **status:** String with values separated by commas, can take values between 1 and 7 (default 1,2,3,4,5,6,7).
 - **id_user:** Filtered by user (default "").
 - **id_group:** Filtered by group (by default one 1).
 - **id_company:** Filtered by company (zero 0 by default).
 - **id_inventory:** Filtered by inventory (zero 0 by default).
 - **priority:** String with values separated by commas, it can take the values 0 (Informative), 1

(Low), 2 (Medium), 3 (Serious), 4 (Very serious) and 10 (Maintenance). By default, all.

Example call to get SLA percentage with result in CSV format:

(syntax)

```
.../api.php?user=user&user_pass=userpass&pass=api_pass&op=get_stats&params=sla_compliance
```

Obtain the total number of open tickets with the NEW status (code 1), in any group:

(syntax)

```
.../api.php?user=user&user_pass=userpass&pass=api_pass&op=get_stats&params=opened,1
```

Get total open tickets (any status, in any group).

(syntax)

```
.../api.php?user=user&user_pass=userpass&pass=api_pass&op=get_stats&params=opened
```

Example output with CSV format:

```
1,"Sample inventory object"  
2,"Another inventory"  
3,"Thirdventory"
```

Example of call with result in XML:

(syntax)

```
.../api.php?user=user&user_pass=userpass&pass=api_pass&op=get_stats&params=sla_compliance&return_type=xml
```

Example output in XML:

```
<xml>  
97  
</xml>
```

get_incidents_summary

Result: Returns the ticket status summary.

Parameters: Field, if the ca is addedmpo will only display that value. The allowed fields are:

- new .
- unconfirmed .
- assigned .
- reopened .
- verified .
- pending_third_person .
- closed .
- not_closed .
- total .

Example of call with result in CSV format:

(syntax)

```
.../api.php?user=user&user_pass=userpass&pass=api_pass&op=get_incidents_summary&return_type=csv
```

Example output with CSV format:

```
NEW, 1
UNCONFIRMED, 0
ASSIGNED, 70
REOPENED, 3
VERIFIED, 12
PENDING_THIRD_PERSON, 308
CLOSED, 3382
NOT_CLOSED, 393
TOTAL, 3776
```

Example of call with result in XML:

(syntax)

```
.../api.php?user=user&user_pass=userpass&pass=api_pass&op=get_incidents_summary&return_type=xml
```

Example output in XML:

```
<xml>
  <new>1</new>
  <unconfirmed>0</unconfirmed>
  <assigned>70</assigned>
  <reopened>3</reopened>
  <verified>12</verified>
  <pending_third_person>308</pending_third_person>
  <closed>3382</closed>
  <not_closed>393</not_closed>
  <total>3776</total>
</xml>
```

Example call with a specific field and result in CSV format:

(syntax)

```
.../api.php?user=user&user_pass=userpass&pass=api_pass&op=get_incidents_summary&return_type=csv&params=not_closed
```

Example output with CSV format:

```
393
```

Example of call with result in XML:

(syntax)

```
.../api.php?user=user&user_pass=userpass&pass=api_pass&op=get_incidents_summary&return_type=xml&params=not_closed
```

Example output in XML:

```
<xml>  
  <data>393</data>  
</xml>
```

get_inventories

Result: Returns a list with the identifiers and names of the Pandora ITSM inventories.

Parameters: None.

Example of call with result in CSV format:

(syntax)

```
.../api.php?user=user&user_pass=userpass&pass=api_pass&op=get_inventories
```

Example output with CSV format:

```
1,"Sample inventory object"  
2,"Another inventory"  
3,"Thirdventory"
```

Example of call with result in XML:

(syntax)

```
.../api.php?user=user&user_pass=userpass&pass=api_pass&op=get_inventories&return_type=xml
```

Example output in XML:

```
<xml>
  <inventory>
    <id>1</id>
    <name>Sample inventory object</name>
  </inventory>
  <inventory>
    <id>2</id>
    <name>Another inventory</name>
  </inventory>
  <inventory>
    <id>3</id>
    <name>Thirdventory</name>
  </inventory>
</xml>
```

get_user_exists_in_group

Result: Returns a 1 if the given user exists in any of the given groups. Zero 0 otherwise.

Serialized parameters:

- user_id.
- group_id: They can be several groups separated by a token.

Call example:

(syntax)

```
.../api.php?user=user&user_pass=userpass&pass=api_pass&op=user_exists_in_group&params=test,4,5
```

get_incident_inventories

Result: Returns a list with inventory objects associated with a ticket.

Serialized parameters:

- Ticket filter: Ticket identifier (required).

Example of call with result in CSV format:

(syntax)

```
.../api.php?user=user&user_pass=userpass&pass=api_pass&op=get_incident_inventories
&params=10
```

Example output with CSV format:

```
2,1,admin,Sanchez,0,Sancho's
computer,0,0,0,0,0000-00-00,new,2013-08-19,0000-00-005,3,admin,InventarioCSV,1,d
escription ,0,0,0,1,2017-08-02,new,2016-10-09,2017-10-09
```

Example of call with result in XML:

(syntax)

```
.../api.php?user=user&user_pass=userpass&pass=api_pass&op=get_incident_inventories
&params=10&return_type=xml
```

Example output in XML:

```
<xml version='1.0' encoding='UTF-8'>
  <inventory>
    <id><![CDATA[2]]></id>
    <id_object_type><![CDATA[1]]></id_object_type>
    <owner><![CDATA[admin]]></owner>
    <name><![CDATA[Sanchez]]></name>
    <public><![CDATA[0]]></public>
    <description><![CDATA[Sancho's computer]]></description>
    <id_contract><![CDATA[0]]></id_contract>
    <id_manufacturer><![CDATA[0]]></id_manufacturer>
    <id_parent><![CDATA[0]]></id_parent>
    <show_list><![CDATA[0]]></show_list>
    <last_update><![CDATA[0000-00-00]]></last_update>
    <status><![CDATA[new]]></status>
    <receipt_date><![CDATA[2013-08-19]]></receipt_date>
    <issue_date><![CDATA[0000-00-00]]></issue_date>
  </inventory>
</xml>
```

get_last_cron_execution

Result: Returns the time that has passed in minutes since the last execution of Crontask.

Parameters: None.

Example call:

(syntax)

```
.../api.php?user=user&user_pass=userpass&pass=api_pass&op=get_last_cron_execution
```

Result:

```
3
```

example call con result in XML:

(syntax)

```
.../api.php?user=user&user_pass=userpass&pass=api_pass&op=get_last_cron_execution&return_type=xml
```

Example output in XML:

```
<xml>
  <cronjob>
    <last_exec>3</last_exec>
  </cronjob>
</xml>
```

get_num_queued_emails

Result: Returns the number of emails that are in the queue waiting to be sent.

Parameters: None.

Example call:

(syntax)

```
.../api.php?user=user&user_pass=userpass&pass=api_pass&op=get_num_queued_emails
```

Result:

```
20
```

Example of call with result in XML:

(syntax)

```
.../api.php?user=user&user_pass=userpass&pass=api_pass&op=get_num_queued_emails&return_type=xml
```

Example output in XML:

```
<xml>
```



```
<pending_email>
  <num>20</num>
</pending_email>
</xml>
```

API Extension

To develop new calls for the API you have to take into account that:

- The call must be registered as a function in the file <PITSM installation>/include/functions_api.php, in addition to being considered among the valid functions in the file <PITSM installation>/include/api.php.
- The function must return a single value or an array of pairs, eg name-value.
- The formatting of the result of the call to XML or CSV that is made in the file api.php must be done automatically.

Function Examples for the API

Function example that returns a single value

Definition in the functions_api.php file:

functions_api.php

```
function host_risk($host_name) {

    $host_id = get_hosts_id($host_name);

    If there is not a host with this name return false
    if (!$host_id) {
        return false;
    }

    $risk = return_last_risk_agent($host_id);

    return $risk;
}
```

Call in the api.php file:

api.php

```
switch($op) {

    ...
}
```

```
case "host_risk":
    if (isset($params['0'])) {
        $result = host_risk($params['0']);
    } else {
        $result = false;
    }
    break;
    ...
}
```

Function example that returns a list of values

Definition in the functions_api.php file:

functions_api.php

```
function all_host_risk() {

    Get all host
    $hosts = get_hosts();

    Check if there is host or not
    if (empty($hosts)) {
        return false;
    }

    $result = array();
    foreach ($hosts as $host) {
        $host_name = get_host_name($host['id']);
        $risk = return_last_risk_agent($host['id']);
        $aux = array();
        $aux['name'] = $host_name;
        $aux['data'] = $risk;

        array_push($result, $aux);
    }

    return $result;
}
```

Call in the api.php file:

api.php

```
switch($op) {
```

```
    ...  
    case "all_host_risk":  
        $result = all_host_risk();  
        break;  
    ...  
}
```

[Back to Pandora ITSM Documentation Index](#)