

---

# phpypam Documentation

**codeaffen**

**Jan 04, 2021**



# USER DOCUMENTATION

<b>1</b>	<b>phpypam: Python API client library for phpIPAM installation</b>	<b>1</b>
<b>2</b>	<b>phpypam</b>	<b>3</b>
2.1	phpypam package . . . . .	3
<b>3</b>	<b>How to contribute to phpipam-ansible-modules</b>	<b>7</b>
3.1	Did you found a bug . . . . .	7
3.2	Did you wrote a patch for an open bug . . . . .	7
3.3	Do you want to add a new feature . . . . .	7
3.4	Do you wnat to contribute to documentation . . . . .	7
3.5	Thank you for any contribution . . . . .	8
<b>4</b>	<b>Indices and tables</b>	<b>9</b>
	<b>Python Module Index</b>	<b>11</b>
	<b>Index</b>	<b>13</b>



## **PHPYPAM: PYTHON API CLIENT LIBRARY FOR PHPIPAM INSTALLATION**

[PyPI version](#) [Codacy Badge](#) [Documentation](#) [Status](#)

phpypam is intended to be a complete library for speaking with phpIPAM API.

Using phpypam is as easy as using the UI.

```
import phpypam

pi = phpypam.api(
    url='https://ipam.example.com',
    app_id='ansible',
    username='apiuser',
    password='apiP455wd',
    ssl_verify=True
)
pi.get_entity(controller='sections')
```



## 2.1 phypam package

Package that provides phpIPAM API interface.

### 2.1.1 Subpackages

#### phypam.core package

Core package provide api and exception classes.

#### Submodules

#### phypam.core.api module

Default class to handle all api interactions with phpIPAM server.

```
class phypam.core.api.Api (url, app_id, username=None, password=None, token=None, encryption=False, timeout=None, ssl_verify=True, user_agent=None)
```

Bases: `object`

The main class.

It generates the API object where you can run different actions again to *create*, *update* and *delete* entities. It also provides functions with informational character only.

**controllers** ()

Report all controllers from phpIPAM API.

This method is used to report all known controllers of phpIPAM API. Unfortunately the API doesn't report all nor the correct paths for all 'controllers'.

**Returns** Returns a tuple of controller paths.

**Return type** tuple

```
create_entity (controller, controller_path=None, data=None, params=None)
```

Create an entity.

**Parameters**

- **controller** (*str*) – Name of the controller to use.

- **controller\_path** (*str*, *optional*) – The path which is used to query for entities, defaults to None
- **data** (*dict*) – Dictionary, list of tuples, bytes, or file-like object to send in the body of the Request.
- **params** (*dict*, *optional*) – Dictionary list of tuples or bytes to send in the query string for the Request., defaults to None

**Returns** Returns the newly created entity.

**Return type** Union[dict, list]

**delete\_entity** (*controller*, *controller\_path*, *params=None*)

Delete an entity.

**Parameters**

- **controller** (*str*) – Name of the controller to use.
- **controller\_path** (*str*) – The path wich is used to access the entity to delete.
- **params** (*dict*, *optional*) – Dictionary, list of tuples or bytes to send in the query string for the Request., defaults to None

**Returns** Returns True if entity was deleted successfully or either 'dict' or 'list' of entities to work on.

**Return type** Union[book, dict, list]

**get\_entity** (*controller*, *controller\_path=None*, *params=None*)

Get existing entity from phpIPAM server.

This method query for existing entity. If there a result it will be returned otherwise an PhpIPAMEntityNotFound exception is raised from underlying method.

**Parameters**

- **controller** (*str*) – Name of the controller to request entity from.
- **controller\_path** (*str*, *optional*) – The path which is used to query for entities, defaults to None
- **params** (*dict*, *optional*) – Request parameters which have to be append to the request URI, defaults to None

**Returns** Result of the query. It can be either a 'list' or 'dict'.

**Return type** Union[dict, list]

**get\_token** ()

Return last login token.

**Returns** Returns the api token from the last successful login.

**Return type** str

**update\_entity** (*controller*, *controller\_path=None*, *data=None*, *params=None*)

Update an entity.

**Parameters**

- **controller** (*str*) – Name of the controller to use.
- **controller\_path** (*str*, *optional*) – The path which is used to access the entity to update., defaults to None



- **data** (*dict, optional*) – Dictionary, list of tuples, bytes, or file-like object to send in the body of the Request., defaults to None
- **params** (*dict, optional*) – Dictionary list of tuples or bytes to send in the query string for the Request., defaults to None

**Returns** Returns either a ‘dict’ or ‘list’ of the changed entity

**Return type** Union[dict, list]

## phpypam.core.exceptions module

Class to provided different Exceptions.

**exception** `phpypam.core.exceptions.PHPyPAMEntityNotFoundException` (*\*args, \*\*kwargs*)

Bases: `Exception`

Exception `PHPyPAMEntityNotFoundException`, children of `Exception`.

This Exception is raised if an entity was not found.

**exception** `phpypam.core.exceptions.PHPyPAMException` (*\*args, code=None, message=None*)

Bases: `Exception`

`PHPyPAMException`, children of `Exception`.

This exception is raised if anythings in `phpypam.api` doesn’t work out.

**exception** `phpypam.core.exceptions.PHPyPAMInvalidCredentials` (*\*args, \*\*kwargs*)

Bases: `Exception`

Exception `PHPyPAMInvalidCredentials`, children of `Exception`.

This Exception is raised if there are any issues with the authentication against `phpIPAM` api.

**exception** `phpypam.core.exceptions.PHPyPAMInvalidSyntax` (*\*args, \*\*kwargs*)

Bases: `Exception`

Exception `PHPyPAMInvalidSyntax`, children of `Exception`.

This Exception is raised if there are any issues with syntax of request against `phpIPAM` api.



## HOW TO CONTRIBUTE TO PHPIPAM-ANSIBLE-MODULES

### 3.1 Did you found a bug

- Do not open Github issue if the bug concerns `{php}IPAM` and not the ansible modules.
- Make sure the bug is not already opened by another user.
- If you can't find an open issue which reflects your observed problem go ahead and [open a new bug](#).
- Provide as much information as mentioned in the bug report template.

### 3.2 Did you wrote a patch for an open bug

- Open new pull request containing the patch.
- Provide a clear description which describes the problem and the solution. Link the existing bug to the PR.

### 3.3 Do you want to add a new feature

- Make sure there isn't already a feature request.
- If you can't find an open feature request which describes your feature idea or parts of it feel free to [open a new feature request](#).
- Suggest your feature idea within the created feature request.
- Provide as much description as possible to enable others to have a good understanding of what you are doing.
- Point out that you want to start to work on the new feature

### 3.4 Do you want to contribute to documentation

- Write your documentation change.
- Open a PR with your change.
- Discuss with the team about your changes.

## 3.5 Thank you for any contribution

We will thank you for heed the contribution guidelines and we encourage you to contribute and join the team.

## INDICES AND TABLES

- `genindex`
- `modindex`
- `search`



## PYTHON MODULE INDEX

### p

- `phpypam`, 3
- `phpypam.core`, 3
- `phpypam.core.api`, 3
- `phpypam.core.exceptions`, 5





## INDEX

### A

`Api` (class in `phpypam.core.api`), 3

### C

`controllers()` (*phpypam.core.api.Api* method), 3

`create_entity()` (*phpypam.core.api.Api* method), 3

### D

`delete_entity()` (*phpypam.core.api.Api* method), 4

### G

`get_entity()` (*phpypam.core.api.Api* method), 4

`get_token()` (*phpypam.core.api.Api* method), 4

### M

module

`phpypam`, 3

`phpypam.core`, 3

`phpypam.core.api`, 3

`phpypam.core.exceptions`, 5

### P

`phpypam`

    module, 3

`phpypam.core`

    module, 3

`phpypam.core.api`

    module, 3

`phpypam.core.exceptions`

    module, 5

`PHPyPAMEntityNotFoundException`, 5

`PHPyPAMException`, 5

`PHPyPAMInvalidCredentials`, 5

`PHPyPAMInvalidSyntax`, 5

### U

`update_entity()` (*phpypam.core.api.Api* method), 4