phpypam Documentation

codeaffen

Oct 30, 2020

USER DOCUMENTATION

CHAPTER

ONE

PHPYPAM: PYTHON API CLIENT LIBRARY FOR PHPIPAM INSTALLATION

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')
```

CHAPTER

TWO

PHPYPAM

2.1 phpypam package

2.1.1 Subpackages

phpypam.core package

Submodules

phpypam.core.api module

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

controllers()

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*)

This method is used to 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*)

Method to get an existing entity

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()

Method to grap 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*)

This method is used to 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

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`

PHPyPAMExeption, 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.

CHAPTER

THREE

HOW TO CONTRIBUTE TO PHPIPAM-ANSIBLE-MODULES

3.1 Did you found a bug

- Do not open Github issue it 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 an 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 describe 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 discribtion 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 wnat to contribute to documentation

- Write you 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.

CHAPTER

FOUR

INDICES AND TABLES

- genindex
- modindex
- search

PYTHON MODULE INDEX

INDEX