
Python Foreman Documentation

Release 0.1.2

David Caro <dcaroest@redhat.com>

April 12, 2016

1	Client module documentation	1
2	Tutorial	5
2.1	Definitions files	5
2.2	Basic Tutorial	5
2.3	Advanced Tutorial	6
3	Indices and tables	7
	Python Module Index	9

Client module documentation

This module provides access to the API of a foreman server

```
class foreman.client.Foreman (url, auth=None, version=None, api_version=None, use_cache=True,  
                             strict_cache=True, timeout=60, timeout_post=600, time-  
                             out_delete=600, timeout_put=None, verify=False)
```

Main client class. It's methods will be autogenerated, check the API docs for your foreman version [here](#).

```
do_delete (url, kwargs)
```

Parameters

- **url** – relative url to resource
- **kwargs** – parameters for the api call

```
do_get (url, kwargs)
```

Parameters

- **url** – relative url to resource
- **kwargs** – parameters for the api call

```
do_post (url, kwargs)
```

Parameters

- **url** – relative url to resource
- **kwargs** – parameters for the api call

```
do_put (url, kwargs)
```

Parameters

- **url** – relative url to resource
- **kwargs** – parameters for the api call

```
get_foreman_version ()
```

Even if we have an api method that returns the foreman version, we need the version first to know its path, so instead of that we get the main page and extract the version from the footer.

```
get_timeout (method=None)
```

Get timeout for given request method

Parameters **method** – Request method (eg. GET, POST, ..). If None, return default timeout.

set_timeout (*timeout, method='DEFAULT'*)

Set the timeout for any connection, the timeout is the requests module timeout (for connection inactivity rather than request total time)

Parameters

- **timeout** – Timeout in seconds for the connection inactivity
- **method** – Request method (eg. GET, POST, ..). By default, set default timeout.

unset_timeout (*method*)

Ensure timeout for given method is not set.

Parameters method – Request method (eg. GET, POST, ..)

exception `foreman.client.ForemanException` (*res, msg*)

exception `foreman.client.ForemanVersionException`

class `foreman.client.MetaForeman`

static convert_plugin_def (*http_method, funcs*)

This function parses one of the elements of the definitions dict for a plugin and extracts the relevant information

Parameters

- **http_method** – HTTP method that uses (GET, POST, DELETE, ...)
- **funcs** – functions related to that HTTP method

class `foreman.client.MethodAPIDescription` (*resource, method, api*)

classmethod **create_param_doc** (*param, prefix=None*)

Generate documentation for single parameter of function :param param: dict contains info about parameter
:param sub: prefix string for recursive purposes

exclude_html_reg = `<_sre.SRE_Pattern object>`

generate_func (*as_global=False*)

Generate function for specific method and using specific api

Parameters as_global – if set, will use the global function name, instead of the class method (usually {resource}_{class_method}) when defining the function

get_global_method_name ()

parse_resource_from_url (*url*)

Returns the appropriate resource name for the given URL.

Parameters url – API URL stub, like: '/api/hosts'

Returns Resource name, like 'hosts', or None if not found

resource_pattern = `<_sre.SRE_Pattern object>`

exception `foreman.client.ObjectNotFound` (*res, msg*)

class `foreman.client.Resource` (*foreman*)

Provides entry point for specific resource.

class `foreman.client.ResourceMeta`

This type composes methods for resource class

exception `foreman.client.Unacceptable` (*res, msg*)

`foreman.client.parse_resource_definition(resource_name, resource_dct)`

Returns all the info extracted from a resource section of the apipie json

Parameters

- **resource_name** – Name of the resource that is defined by the section
- **resrouce_dct** – Dictionary as generated by apipie of the resource definition

`foreman.client.parse_version(version_string)`

Parameters version_string – Version string to parse, like '1.2.3'

Passing to int as many of the elements as possible to support comparing ints of different number of chars (2<10 but '2'>'10'). So we just accept that any element with chars will be considered lesser to any int element.

`foreman.client.res_to_str(res)`

Parameters res – `requests.Response` object

Parse the given request and generate an informative string from it

`foreman.client.try_int(what)`

2.1 Definitions files

The Python Foreman can get the methods and it's definitions from two places the `definitions` directory or the foreman instance.

This `definitions` directory contains some `apipie` json definitions retrieved from different foreman versions and api versions, by default it will try to match the foreman version with the fittest of those files.

It can also get it's definitions from the live Foreman instance, to do that, you have to make sure that the urls:

- `FOREMAN_URL/apidoc/v2.json`
- `FOREMAN_URL/apidoc/v1.json`

are available, usually that means that you'll have to set the `config.use_cache` parameter for the `apipie` gem to `false` (normally found under `FOREMAN_HOME/config/initializers/apipie.rb`)

2.2 Basic Tutorial

2.2.1 Connect

To connect to a foreman server just instantiate a `foreman.client.Foreman` object with the server's url and authentication aparameters, like this:

```
>>> from getpass import getpass
>>> from foreman.client import Foreman
>>> f = Foreman('http://myforeman.server:3000', ('myuser', getpass()))
```

The available methods are separated in two ways, by resource (`apiv2`) and by method (`$type_$resource_$method`).

Take into account that all those methods are autogenerated and they might vary for different versions of Foreman.

2.2.2 index

Those are the main methods to get info for groups of objects, for example, to get a summary of all the hosts you could do:

```
>>> f.index_hosts()
>>> f.hosts.index()
```

Take into account that it accepts some parameters to handle the paging and the ammount of elements to get.

2.2.3 show

This methods give you all the information for a specific object, for example:

```
>>> f.show_hosts(id=1)
>>> f.hosts.show(id='myuber.host.com')
```

Will show all the info for the host with id 1.

2.2.4 create

This methods create a new object into foreman. An example:

```
>>> f.create_hosts(host={'name': 'mynewhost', 'ip': '192.168.1.1', 'mac': '00:00:00:00:00:00'})
>>> f.hosts.create(host={'name': 'mynewhost', 'ip': '192.168.1.1', 'mac': '00:00:00:00:00:00'})
```

To see the exact parameters look at the **‘foreman API docs<<http://theforeman.org/api.html>>’**.

2.2.5 update

This methods update the info for the given object, usually called with an id and a hash representating the object.

2.2.6 destroy

This methods give you a way to destroy any object.

2.3 Advanced Tutorial

TODO

Indices and tables

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

f

`foreman.client`, [1](#)

C

convert_plugin_def() (foreman.client.MetaForeman static method), 2
create_param_doc() (foreman.client.MethodAPIDescription class method), 2

D

do_delete() (foreman.client.Foreman method), 1
do_get() (foreman.client.Foreman method), 1
do_post() (foreman.client.Foreman method), 1
do_put() (foreman.client.Foreman method), 1

E

exclude_html_reg (foreman.client.MethodAPIDescription attribute), 2

F

Foreman (class in foreman.client), 1
foreman.client (module), 1
ForemanException, 2
ForemanVersionException, 2

G

generate_func() (foreman.client.MethodAPIDescription method), 2
get_foreman_version() (foreman.client.Foreman method), 1
get_global_method_name() (foreman.client.MethodAPIDescription method), 2
get_timeout() (foreman.client.Foreman method), 1

M

MetaForeman (class in foreman.client), 2
MethodAPIDescription (class in foreman.client), 2

O

ObjectNotFound, 2

P

parse_resource_definition() (in module foreman.client), 2
parse_resource_from_url() (foreman.client.MethodAPIDescription method), 2
parse_version() (in module foreman.client), 3

R

res_to_str() (in module foreman.client), 3
Resource (class in foreman.client), 2
resource_pattern (foreman.client.MethodAPIDescription attribute), 2
ResourceMeta (class in foreman.client), 2

S

set_timeout() (foreman.client.Foreman method), 1

T

try_int() (in module foreman.client), 3

U

Unacceptable, 2
unset_timeout() (foreman.client.Foreman method), 2