Python Ironic Client Documentation

Release 5.1.1.dev3

OpenStack LLC

Mar 19, 2025

CONTENTS

This is a client for the OpenStack Ironic API. It provides:

- a Python API: the `ironicclient` module, and
- command-line interface: `openstack baremetal`

CHAPTER

ONE

CONTENTS

1.1 ironicclient Python API

The ironicclient python API lets you access ironic, the OpenStack Bare Metal Provisioning Service.

For example, to manipulate nodes, you interact with an *ironicclient.v1.node.Node* object. You obtain access to nodes via attributes of the *ironicclient.v1.client.Client* object.

1.1.1 Usage

Get a Client object

First, create an *ironicclient.v1.client.Client* instance by passing your credentials to *ironicclient.client.get_client()*. By default, the Bare Metal Provisioning system is configured so that only administrators (users with admin role) have access.

Note: Explicit instantiation of *ironicclient.v1.client.Client* may cause errors since it doesnt verify provided arguments, using *ironicclient.client.get_client()* is preferred way to get client object.

There are two different sets of credentials that can be used:

```
* ironic endpoint and auth token
* Identity Service (keystone) credentials
```

Using ironic endpoint and auth token

An auth token and the ironic endpoint can be used to authenticate:

```
* os_auth_token: authentication token (from Identity Service)
* ironic_url: ironic API endpoint, eg http://ironic.example.org:6385/v1
```

To create the client, you can use the API like so:

```
>>> from ironicclient import client
>>>
>>> kwargs = {'os_auth_token': '3bcc3d3a03f44e3d8377f9247b0ad155',
```

(continues on next page)

(continued from previous page)

```
>>>             'ironic_url': 'http://ironic.example.org:6385/'}
>>> ironic = client.get_client(1, **kwargs)
```

Using Identity Service (keystone) credentials

These Identity Service credentials can be used to authenticate:

```
* os_username: name of user
* os_password: user's password
* os_auth_url: Identity Service endpoint for authorization
* insecure: Boolean. If True, does not perform X.509 certificate
  validation when establishing SSL connection with identity
  service. default: False (optional)
* os_tenant_{name|id}: name or ID of tenant
```

Also the following parameters are required when using the Identity API v3:

```
* os_user_domain_name: name of a domain the user belongs to,
  usually 'default'
* os_project_domain_name: name of a domain the project belongs to,
  usually 'default'
```

To create a client, you can use the API like so:

```
>>> from ironicclient import client
>>>
>>> kwargs = {'os_username': 'name',
>>>             'os_password': 'password',
>>>             'os_auth_url': 'http://keystone.example.org:5000/',
>>>             'os_project_name': 'project'}
>>> ironic = client.get_client(1, **kwargs)
```

Perform ironic operations

Once you have an *ironicclient.v1.client.Client*, you can perform various tasks:

```
>>> ironic.driver.list()  # list of drivers
>>> ironic.node.list()  # list of nodes
>>> ironic.node.get(node_uuid)  # information about a particular node
```

When the Client needs to propagate an exception, it will usually raise an instance subclassed from *ironicclient.common.apiclient.exceptions.ClientException*.

Refer to the modules themselves, for more details.

1.1.2 ironicclient Modules

- modindex

1.2 python-ironicclient User Documentation

1.2.1 baremetal Standalone Command-Line Interface (CLI)

Synopsis

baremetal [options] <command> [command-options]

baremetal help <command>

Description

The standalone baremetal tool allows interacting with the Bare Metal service without installing the OpenStack Client tool as in *openstack baremetal Command-Line Interface (CLI)*.

The standalone tool is mostly identical to its OSC counterpart, with two exceptions:

1. No need to prefix commands with openstack.
2. No authentication is assumed by default.

Check the *OSC CLI reference* for a list of available commands.

Inspector support

The standalone baremetal tool optionally supports the low-level bare metal introspection API provided by ironic-inspector. If ironic-inspector-client is installed, its commands are automatically available (also without the openstack prefix).

Standalone usage

To use the CLI with a standalone bare metal service, you need to provide an endpoint to connect to. It can be done in three ways:

1. Provide an explicit --os-endpoint argument, e.g.:

```
$ baremetal --os-endpoint https://ironic.host:6385 node list
```

2. Set the corresponding environment variable, e.g.:

```
$ export OS_ENDPOINT=https://ironic.host:6385
$ baremetal node list
```

3. Populate a clouds.yaml file, setting baremetal_endpoint_override, e.g.:

```
$ cat ~/.config/openstack/clouds.yaml
clouds:
  ironic:
    auth_type: none
    baremetal_endpoint_override: http://127.0.0.1:6385
$ export OS_CLOUD=ironic
$ baremetal node list
```

Inspector support works similarly, but the `clouds.yaml` option is called `baremetal_introspection_endpoint_override`. The two endpoints can be configured simultaneously, e.g.:

```
$ cat ~/.config/openstack/clouds.yaml
clouds:
  ironic:
    auth_type: none
    baremetal_endpoint_override: http://127.0.0.1:6385
    baremetal_introspection_endpoint_override: http://127.0.0.1:5050
$ export OS_CLOUD=ironic
$ baremetal node list
$ baremetal introspection list
```

Usage with OpenStack

The standalone CLI can also be used with the Bare Metal service installed as part of OpenStack. See *Authentication* for information on the required input.

1.2.2 `openstack baremetal` Command-Line Interface (CLI)

Synopsis

`openstack [options] baremetal` <command> [command-options]

`openstack help baremetal` <command>

Description

The OpenStack Client plugin interacts with the Bare Metal service through the `openstack baremetal` command line interface (CLI).

To use the `openstack` CLI, the OpenStackClient (python-openstackclient) package must be installed. There are two ways to do this:

- along with this python-ironicclient package:

  ```
  $ pip install python-ironicclient[cli]
  ```

- directly:

```
$ pip install python-openstackclient
```

This CLI is provided by python-openstackclient and osc-lib projects:

- https://opendev.org/openstack/python-openstackclient
- https://opendev.org/openstack/osc-lib

Authentication

To use the CLI, you must provide your OpenStack username, password, project, and auth endpoint. You can use configuration options `--os-username`, `--os-password`, `--os-project-id` (or `--os-project-name`), and `--os-auth-url`, or set the corresponding environment variables:

```
$ export OS_USERNAME=user
$ export OS_PASSWORD=password
$ export OS_PROJECT_NAME=project                    # or OS_PROJECT_ID
$ export OS_PROJECT_DOMAIN_ID=default
$ export OS_USER_DOMAIN_ID=default
$ export OS_IDENTITY_API_VERSION=3
$ export OS_AUTH_URL=http://auth.example.com:5000/identity
```

Getting help

To get a list of available (sub)commands and options, run:

```
$ openstack help baremetal
```

To get usage and options of a command, run:

```
$ openstack help baremetal <sub-command>
```

Examples

Get information about the openstack baremetal node create command:

```
$ openstack help baremetal node create
```

Get a list of available drivers:

```
$ openstack baremetal driver list
```

Enroll a node with the `ipmi` driver:

```
$ openstack baremetal node create --driver ipmi --driver-info ipmi_address=1.
↪2.3.4
```

Get a list of nodes:

```
$ openstack baremetal node list
```

The baremetal API version can be specified via:

- environment variable OS_BAREMETAL_API_VERSION:

```
$ export OS_BAREMETAL_API_VERSION=1.25
```

- or optional command line argument os-baremetal-api-version:

```
$ openstack baremetal port group list --os-baremetal-api-version 1.25
```

Command Reference

Command Reference

List of released CLI commands available in openstack client. These commands can be referenced by doing `openstack help baremetal`.

baremetal allocation

baremetal allocation create

Create a new baremetal allocation.

```
openstack baremetal allocation create
    [-f {json,shell,table,value,yaml}]
    [-c COLUMN]
    [--noindent]
    [--prefix PREFIX]
    [--max-width <integer>]
    [--fit-width]
    [--print-empty]
    [--resource-class RESOURCE_CLASS]
    [--trait TRAITS]
    [--candidate-node CANDIDATE_NODES]
    [--name NAME]
    [--uuid UUID]
    [--owner OWNER]
    [--extra <key=value>]
    [--wait [<time-out>]]
    [--node NODE]
```

-f <FORMATTER>, **--format** <FORMATTER>

the output format, defaults to table

-c COLUMN, **--column** COLUMN

specify the column(s) to include, can be repeated to show multiple columns

--noindent
whether to disable indenting the JSON

--prefix <PREFIX>
add a prefix to all variable names

--max-width <integer>
Maximum display width, <1 to disable. You can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.

--fit-width
Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable

--print-empty
Print empty table if there is no data to show.

--resource-class <RESOURCE_CLASS>
Resource class to request.

--trait <TRAITS>
A trait to request. Can be specified multiple times.

--candidate-node <CANDIDATE_NODES>
A candidate node for this allocation. Can be specified multiple times. If at least one is specified, only the provided candidate nodes are considered for the allocation.

--name <NAME>
Unique name of the allocation.

--uuid <UUID>
UUID of the allocation.

--owner <OWNER>
Owner of the allocation.

--extra <key=value>
Record arbitrary key/value metadata. Can be specified multiple times.

--wait <time-out>
Wait for the new allocation to become active. An error is returned if allocation fails and wait is used. Optionally takes a timeout value (in seconds). The default value is 0, meaning it will wait indefinitely.

--node <NODE>
Backfill this allocation from the provided node that has already been deployed. Bypasses the normal allocation process.

This command is provided by the python-ironicclient plugin.

baremetal allocation delete

Unregister baremetal allocation(s).

```
openstack baremetal allocation delete <allocation> [<allocation> ...]
```

allocation

Allocations(s) to delete (name or UUID).

This command is provided by the python-ironicclient plugin.

baremetal allocation list

List baremetal allocations.

```
openstack baremetal allocation list
    [-f {csv,json,table,value,yaml}]
    [-c COLUMN]
    [--quote {all,minimal,none,nonnumeric}]
    [--noindent]
    [--max-width <integer>]
    [--fit-width]
    [--print-empty]
    [--sort-column SORT_COLUMN]
    [--sort-ascending | --sort-descending]
    [--limit <limit>]
    [--marker <allocation>]
    [--sort <key>[:<direction>]]
    [--node <node>]
    [--resource-class <resource_class>]
    [--state <state>]
    [--owner <owner>]
    [--long | --fields <field> [<field> ...]]
```

-f <FORMATTER>, **--format** <FORMATTER>

the output format, defaults to table

-c COLUMN, **--column** COLUMN

specify the column(s) to include, can be repeated to show multiple columns

--quote <QUOTE_MODE>

when to include quotes, defaults to nonnumeric

--noindent

whether to disable indenting the JSON

--max-width <integer>

Maximum display width, <1 to disable. You can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.

--fit-width

Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable

--print-empty

Print empty table if there is no data to show.

--sort-column `SORT_COLUMN`

specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated

--sort-ascending

sort the column(s) in ascending order

--sort-descending

sort the column(s) in descending order

--limit `<limit>`

Maximum number of allocations to return per request, 0 for no limit. Default is the maximum number used by the Baremetal API Service.

--marker `<allocation>`

Allocation UUID (for example, of the last allocation in the list from a previous request). Returns the list of allocations after this UUID.

--sort `<key>[:<direction>]`

Sort output by specified allocation fields and directions (asc or desc) (default: asc). Multiple fields and directions can be specified, separated by comma.

--node `<node>`

Only list allocations of this node (name or UUID).

--resource-class `<resource_class>`

Only list allocations with this resource class.

--state `<state>`

Only list allocations in this state.

--owner `<owner>`

Only list allocations with this owner.

--long

Show detailed information about the allocations.

--fields `<field>`

One or more allocation fields. Only these fields will be fetched from the server. Can not be used when long is specified.

This command is provided by the python-ironicclient plugin.

baremetal allocation set

Set baremetal allocation properties.

```
openstack baremetal allocation set
    [--name <name>]
    [--extra <key=value>]
    <allocation>
```

--name <name>

Set the name of the allocation

--extra <key=value>

Extra property to set on this allocation (repeat option to set multiple extra properties)

allocation

Name or UUID of the allocation

This command is provided by the python-ironicclient plugin.

baremetal allocation show

Show baremetal allocation details.

```
openstack baremetal allocation show
    [-f {json,shell,table,value,yaml}]
    [-c COLUMN]
    [--noindent]
    [--prefix PREFIX]
    [--max-width <integer>]
    [--fit-width]
    [--print-empty]
    [--fields <field> [<field> ...]]
    <id>
```

-f <FORMATTER>, **--format** <FORMATTER>

the output format, defaults to table

-c COLUMN, **--column** COLUMN

specify the column(s) to include, can be repeated to show multiple columns

--noindent

whether to disable indenting the JSON

--prefix <PREFIX>

add a prefix to all variable names

--max-width <integer>

Maximum display width, <1 to disable. You can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.

--fit-width
Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable

--print-empty
Print empty table if there is no data to show.

--fields <field>
One or more allocation fields. Only these fields will be fetched from the server.

id
UUID or name of the allocation

This command is provided by the python-ironicclient plugin.

baremetal allocation unset

Unset baremetal allocation properties.

```
openstack baremetal allocation unset
    [--name]
    [--extra <key>]
    <allocation>
```

--name
Unset the name of the allocation

--extra <key>
Extra property to unset on this baremetal allocation (repeat option to unset multiple extra property).

allocation
Name or UUID of the allocation

This command is provided by the python-ironicclient plugin.

baremetal chassis

baremetal chassis create

Create a new chassis.

```
openstack baremetal chassis create
    [-f {json,shell,table,value,yaml}]
    [-c COLUMN]
    [--noindent]
    [--prefix PREFIX]
    [--max-width <integer>]
    [--fit-width]
    [--print-empty]
    [--description <description>]
    [--extra <key=value>]
    [--uuid <uuid>]
```

`-f <FORMATTER>, --format <FORMATTER>`

  the output format, defaults to table

`-c COLUMN, --column COLUMN`

  specify the column(s) to include, can be repeated to show multiple columns

`--noindent`

  whether to disable indenting the JSON

`--prefix <PREFIX>`

  add a prefix to all variable names

`--max-width <integer>`

  Maximum display width, <1 to disable. You can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.

`--fit-width`

  Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable

`--print-empty`

  Print empty table if there is no data to show.

`--description <description>`

  Description for the chassis

`--extra <key=value>`

  Record arbitrary key/value metadata. Can be specified multiple times.

`--uuid <uuid>`

  Unique UUID of the chassis

This command is provided by the python-ironicclient plugin.

baremetal chassis delete

Delete a chassis.

```
openstack baremetal chassis delete <chassis> [<chassis> ...]
```

`chassis`

  UUIDs of chassis to delete

This command is provided by the python-ironicclient plugin.

baremetal chassis list

List the chassis.

```
openstack baremetal chassis list
    [-f {csv,json,table,value,yaml}]
    [-c COLUMN]
    [--quote {all,minimal,none,nonnumeric}]
    [--noindent]
    [--max-width <integer>]
    [--fit-width]
    [--print-empty]
    [--sort-column SORT_COLUMN]
    [--sort-ascending | --sort-descending]
    [--fields <field> [<field> ...]]
    [--limit <limit>]
    [--long]
    [--marker <chassis>]
    [--sort <key>[:<direction>]]
```

-f <FORMATTER>, **--format** <FORMATTER>

the output format, defaults to table

-c COLUMN, **--column** COLUMN

specify the column(s) to include, can be repeated to show multiple columns

--quote <QUOTE_MODE>

when to include quotes, defaults to nonnumeric

--noindent

whether to disable indenting the JSON

--max-width <integer>

Maximum display width, <1 to disable. You can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.

--fit-width

Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable

--print-empty

Print empty table if there is no data to show.

--sort-column SORT_COLUMN

specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated

--sort-ascending

sort the column(s) in ascending order

--sort-descending

sort the column(s) in descending order

--fields <field>

One or more chassis fields. Only these fields will be fetched from the server. Cannot be used when long is specified.

--limit <limit>

Maximum number of chassis to return per request, 0 for no limit. Default is the maximum number used by the Baremetal API Service.

--long

Show detailed information about the chassis

--marker <chassis>

Chassis UUID (for example, of the last chassis in the list from a previous request). Returns the list of chassis after this UUID.

--sort <key>[:<direction>]

Sort output by specified chassis fields and directions (asc or desc) (default: asc). Multiple fields and directions can be specified, separated by comma.

This command is provided by the python-ironicclient plugin.

baremetal chassis set

Set chassis properties.

```
openstack baremetal chassis set
    [--description <description>]
    [--extra <key=value>]
    <chassis>
```

--description <description>

Set the description of the chassis

--extra <key=value>

Extra to set on this chassis (repeat option to set multiple extras)

chassis

UUID of the chassis

This command is provided by the python-ironicclient plugin.

baremetal chassis show

Show chassis details.

```
openstack baremetal chassis show
    [-f {json,shell,table,value,yaml}]
    [-c COLUMN]
    [--noindent]
    [--prefix PREFIX]
    [--max-width <integer>]
```

(continues on next page)

(continued from previous page)

```
    [--fit-width]
    [--print-empty]
    [--fields <field> [<field> ...]]
    <chassis>
```

-f <FORMATTER>, **--format** <FORMATTER>

the output format, defaults to table

-c COLUMN, **--column** COLUMN

specify the column(s) to include, can be repeated to show multiple columns

--noindent

whether to disable indenting the JSON

--prefix <PREFIX>

add a prefix to all variable names

--max-width <integer>

Maximum display width, <1 to disable. You can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.

--fit-width

Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable

--print-empty

Print empty table if there is no data to show.

--fields <field>

One or more chassis fields. Only these fields will be fetched from the server.

chassis

UUID of the chassis

This command is provided by the python-ironicclient plugin.

baremetal chassis unset

Unset chassis properties.

```
openstack baremetal chassis unset
    [--description]
    [--extra <key>]
    <chassis>
```

--description

Clear the chassis description

--extra <key>

Extra to unset on this chassis (repeat option to unset multiple extras)

chassis

UUID of the chassis

This command is provided by the python-ironicclient plugin.

baremetal create

baremetal create

Create resources from files

```
openstack baremetal create <file> [<file> ...]
```

file

File (.yaml or .json) containing descriptions of the resources to create. Can be specified multiple times.

This command is provided by the python-ironicclient plugin.

baremetal driver

baremetal driver list

List the enabled drivers.

```
openstack baremetal driver list
    [-f {csv,json,table,value,yaml}]
    [-c COLUMN]
    [--quote {all,minimal,none,nonnumeric}]
    [--noindent]
    [--max-width <integer>]
    [--fit-width]
    [--print-empty]
    [--sort-column SORT_COLUMN]
    [--sort-ascending | --sort-descending]
    [--type <type>]
    [--long | --fields <field> [<field> ...]]
```

-f <FORMATTER>, **--format** <FORMATTER>

the output format, defaults to table

-c COLUMN, **--column** COLUMN

specify the column(s) to include, can be repeated to show multiple columns

--quote <QUOTE_MODE>

when to include quotes, defaults to nonnumeric

--noindent

whether to disable indenting the JSON

--max-width <integer>

Maximum display width, <1 to disable. You can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.

--fit-width

Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable

--print-empty

Print empty table if there is no data to show.

--sort-column SORT_COLUMN

specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated

--sort-ascending

sort the column(s) in ascending order

--sort-descending

sort the column(s) in descending order

--type <type>

Type of driver (classic or dynamic). The default is to list all of them.

--long

Show detailed information about the drivers.

--fields <field>

One or more node fields. Only these fields will be fetched from the server. Can not be used when long is specified.

This command is provided by the python-ironicclient plugin.

baremetal driver passthru call

Call a vendor passthru method for a driver.

```
openstack baremetal driver passthru call
    [-f {json,shell,table,value,yaml}]
    [-c COLUMN]
    [--noindent]
    [--prefix PREFIX]
    [--max-width <integer>]
    [--fit-width]
    [--print-empty]
    [--arg <key=value>]
    [--http-method <http-method>]
    <driver>
    <method>
```

-f <FORMATTER>, **--format** <FORMATTER>

the output format, defaults to table

-c COLUMN, **--column** COLUMN

specify the column(s) to include, can be repeated to show multiple columns

--noindent

whether to disable indenting the JSON

--prefix <PREFIX>

add a prefix to all variable names

--max-width <integer>

Maximum display width, <1 to disable. You can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.

--fit-width

Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable

--print-empty

Print empty table if there is no data to show.

--arg <key=value>

Argument to pass to the passthru method (repeat option to specify multiple arguments).

--http-method <http-method>

The HTTP method to use in the passthru request. One of DELETE, GET, PATCH, POST, PUT. Defaults to POST.

driver

Name of the driver.

method

Vendor passthru method to be called.

This command is provided by the python-ironicclient plugin.

baremetal driver passthru list

List available vendor passthru methods for a driver.

```
openstack baremetal driver passthru list
    [-f {csv,json,table,value,yaml}]
    [-c COLUMN]
    [--quote {all,minimal,none,nonnumeric}]
    [--noindent]
    [--max-width <integer>]
    [--fit-width]
    [--print-empty]
    [--sort-column SORT_COLUMN]
    [--sort-ascending | --sort-descending]
    <driver>
```

-f <FORMATTER>, **--format** <FORMATTER>

the output format, defaults to table

-c COLUMN, **--column** COLUMN

specify the column(s) to include, can be repeated to show multiple columns

--quote <QUOTE_MODE>

when to include quotes, defaults to nonnumeric

--noindent

whether to disable indenting the JSON

--max-width <integer>

Maximum display width, <1 to disable. You can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.

--fit-width

Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable

--print-empty

Print empty table if there is no data to show.

--sort-column SORT_COLUMN

specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated

--sort-ascending

sort the column(s) in ascending order

--sort-descending

sort the column(s) in descending order

driver

Name of the driver.

This command is provided by the python-ironicclient plugin.

baremetal driver property list

List the driver properties.

```
openstack baremetal driver property list
    [-f {csv,json,table,value,yaml}]
    [-c COLUMN]
    [--quote {all,minimal,none,nonnumeric}]
    [--noindent]
    [--max-width <integer>]
    [--fit-width]
    [--print-empty]
    [--sort-column SORT_COLUMN]
    [--sort-ascending | --sort-descending]
    <driver>
```

-f <FORMATTER>, **--format** <FORMATTER>

the output format, defaults to table

-c COLUMN, **--column** COLUMN

specify the column(s) to include, can be repeated to show multiple columns

--quote <QUOTE_MODE>

when to include quotes, defaults to nonnumeric

--noindent

whether to disable indenting the JSON

--max-width <integer>

Maximum display width, <1 to disable. You can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.

--fit-width

Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable

--print-empty

Print empty table if there is no data to show.

--sort-column SORT_COLUMN

specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated

--sort-ascending

sort the column(s) in ascending order

--sort-descending

sort the column(s) in descending order

driver

Name of the driver.

This command is provided by the python-ironicclient plugin.

baremetal driver raid property list

List a drivers RAID logical disk properties.

```
openstack baremetal driver raid property list
    [-f {csv,json,table,value,yaml}]
    [-c COLUMN]
    [--quote {all,minimal,none,nonnumeric}]
    [--noindent]
    [--max-width <integer>]
    [--fit-width]
    [--print-empty]
    [--sort-column SORT_COLUMN]
    [--sort-ascending | --sort-descending]
    <driver>
```

-f <FORMATTER>, **--format** <FORMATTER>

the output format, defaults to table

-c COLUMN, **--column** COLUMN

specify the column(s) to include, can be repeated to show multiple columns

--quote <QUOTE_MODE>

when to include quotes, defaults to nonnumeric

--noindent

whether to disable indenting the JSON

--max-width <integer>

Maximum display width, <1 to disable. You can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.

--fit-width

Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable

--print-empty

Print empty table if there is no data to show.

--sort-column SORT_COLUMN

specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated

--sort-ascending

sort the column(s) in ascending order

--sort-descending

sort the column(s) in descending order

driver

Name of the driver.

This command is provided by the python-ironicclient plugin.

baremetal driver show

Show information about a driver.

```
openstack baremetal driver show
    [-f {json,shell,table,value,yaml}]
    [-c COLUMN]
    [--noindent]
    [--prefix PREFIX]
    [--max-width <integer>]
    [--fit-width]
    [--print-empty]
    [--fields <field> [<field> ...]]
    <driver>
```

-f <FORMATTER>, **--format** <FORMATTER>

the output format, defaults to table

-c COLUMN, **--column** COLUMN

specify the column(s) to include, can be repeated to show multiple columns

--noindent

whether to disable indenting the JSON

--prefix <PREFIX>

add a prefix to all variable names

--max-width <integer>

Maximum display width, <1 to disable. You can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.

--fit-width

Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable

--print-empty

Print empty table if there is no data to show.

--fields <field>

One or more node fields. Only these fields will be fetched from the server.

driver

Name of the driver.

This command is provided by the python-ironicclient plugin.

baremetal node

baremetal node abort

Set provision state of baremetal node to abort

```
openstack baremetal node abort <node>
```

node

Name or UUID of the node.

This command is provided by the python-ironicclient plugin.

baremetal node add trait

Add traits to a node.

```
openstack baremetal node add trait <node> <trait> [<trait> ...]
```

node

Name or UUID of the node

trait

Trait(s) to add

This command is provided by the python-ironicclient plugin.

baremetal node adopt

Set provision state of baremetal node to adopt

```
openstack baremetal node adopt [--wait [<time-out>]] <node>
```

--wait <time-out>

Wait for a node to reach the desired state, active. Optionally takes a timeout value (in seconds). The default value is 0, meaning it will wait indefinitely.

node

Name or UUID of the node.

This command is provided by the python-ironicclient plugin.

baremetal node bios setting list

List a nodes BIOS settings.

```
openstack baremetal node bios setting list
    [-f {csv,json,table,value,yaml}]
    [-c COLUMN]
    [--quote {all,minimal,none,nonnumeric}]
    [--noindent]
    [--max-width <integer>]
    [--fit-width]
    [--print-empty]
    [--sort-column SORT_COLUMN]
    [--sort-ascending | --sort-descending]
    [--long | --fields <field> [<field> ...]]
    <node>
```

-f <FORMATTER>, **--format** <FORMATTER>

the output format, defaults to table

-c COLUMN, **--column** COLUMN

specify the column(s) to include, can be repeated to show multiple columns

--quote <QUOTE_MODE>

when to include quotes, defaults to nonnumeric

--noindent

whether to disable indenting the JSON

--max-width <integer>

Maximum display width, <1 to disable. You can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.

--fit-width

Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable

--print-empty
Print empty table if there is no data to show.

--sort-column SORT_COLUMN
specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated

--sort-ascending
sort the column(s) in ascending order

--sort-descending
sort the column(s) in descending order

--long
Show detailed information about the BIOS settings.

--fields <field>
One or more node fields. Only these fields will be fetched from the server. Can not be used when long is specified.

node
Name or UUID of the node

This command is provided by the python-ironicclient plugin.

baremetal node bios setting show

Show a specific BIOS setting for a node.

```
openstack baremetal node bios setting show
    [-f {json,shell,table,value,yaml}]
    [-c COLUMN]
    [--noindent]
    [--prefix PREFIX]
    [--max-width <integer>]
    [--fit-width]
    [--print-empty]
    <node>
    <setting
    name>
```

-f <FORMATTER>, **--format** <FORMATTER>
the output format, defaults to table

-c COLUMN, **--column** COLUMN
specify the column(s) to include, can be repeated to show multiple columns

--noindent
whether to disable indenting the JSON

--prefix <PREFIX>
add a prefix to all variable names

--max-width <integer>

Maximum display width, <1 to disable. You can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.

--fit-width

Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable

--print-empty

Print empty table if there is no data to show.

node

Name or UUID of the node

setting name

Setting name to show

This command is provided by the python-ironicclient plugin.

baremetal node boot device set

Set the boot device for a node

```
openstack baremetal node boot device set [--persistent] <node> <device>
```

--persistent

Make changes persistent for all future boots

node

Name or UUID of the node

device

One of bios, cdrom, disk, pxe, safe, wanboot

This command is provided by the python-ironicclient plugin.

baremetal node boot device show

Show the boot device information for a node

```
openstack baremetal node boot device show
    [-f {json,shell,table,value,yaml}]
    [-c COLUMN]
    [--noindent]
    [--prefix PREFIX]
    [--max-width <integer>]
    [--fit-width]
    [--print-empty]
    [--supported]
    <node>
```

-f <FORMATTER>, **--format** <FORMATTER>

the output format, defaults to table

-c COLUMN, **--column** COLUMN

specify the column(s) to include, can be repeated to show multiple columns

--noindent

whether to disable indenting the JSON

--prefix <PREFIX>

add a prefix to all variable names

--max-width <integer>

Maximum display width, <1 to disable. You can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.

--fit-width

Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable

--print-empty

Print empty table if there is no data to show.

--supported

Show the supported boot devices

node

Name or UUID of the node

This command is provided by the python-ironicclient plugin.

baremetal node boot mode set

Set the boot mode for the next baremetal node deployment

```
openstack baremetal node boot mode set <node> <boot_mode>
```

node

Name or UUID of the node.

boot_mode

The boot mode to set for node (uefi/bios)

This command is provided by the python-ironicclient plugin.

baremetal node clean

Set provision state of baremetal node to clean

```
openstack baremetal node clean
    [--wait [<time-out>]]
    --clean-steps <clean-steps>
    <node>
```

--wait <time-out>

Wait for a node to reach the desired state, manageable. Optionally takes a timeout value (in seconds). The default value is 0, meaning it will wait indefinitely.

--clean-steps <clean-steps>

The clean steps. May be the path to a YAML file containing the clean steps; OR -, with the clean steps being read from standard input; OR a JSON string. The value should be a list of clean-step dictionaries; each dictionary should have keys interface and step, and optional key args.

node

Name or UUID of the node.

This command is provided by the python-ironicclient plugin.

baremetal node console disable

Disable console access for a node

```
openstack baremetal node console disable <node>
```

node

Name or UUID of the node

This command is provided by the python-ironicclient plugin.

baremetal node console enable

Enable console access for a node

```
openstack baremetal node console enable <node>
```

node

Name or UUID of the node

This command is provided by the python-ironicclient plugin.

baremetal node console show

Show console information for a node

```
openstack baremetal node console show
    [-f {json,shell,table,value,yaml}]
    [-c COLUMN]
    [--noindent]
    [--prefix PREFIX]
    [--max-width <integer>]
    [--fit-width]
    [--print-empty]
    <node>
```

-f <FORMATTER>, **--format** <FORMATTER>

the output format, defaults to table

-c COLUMN, **--column** COLUMN

specify the column(s) to include, can be repeated to show multiple columns

--noindent

whether to disable indenting the JSON

--prefix <PREFIX>

add a prefix to all variable names

--max-width <integer>

Maximum display width, <1 to disable. You can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.

--fit-width

Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable

--print-empty

Print empty table if there is no data to show.

node

Name or UUID of the node

This command is provided by the python-ironicclient plugin.

baremetal node create

Register a new node with the baremetal service

```
openstack baremetal node create
    [-f {json,shell,table,value,yaml}]
    [-c COLUMN]
    [--noindent]
    [--prefix PREFIX]
    [--max-width <integer>]
```

(continues on next page)

(continued from previous page)

```
    [--fit-width]
    [--print-empty]
    [--chassis-uuid <chassis>]
    --driver <driver>
    [--driver-info <key=value>]
    [--property <key=value>]
    [--extra <key=value>]
    [--uuid <uuid>]
    [--name <name>]
    [--bios-interface <bios_interface>]
    [--boot-interface <boot_interface>]
    [--console-interface <console_interface>]
    [--deploy-interface <deploy_interface>]
    [--inspect-interface <inspect_interface>]
    [--management-interface <management_interface>]
    [--network-data <network data>]
    [--network-interface <network_interface>]
    [--power-interface <power_interface>]
    [--raid-interface <raid_interface>]
    [--rescue-interface <rescue_interface>]
    [--storage-interface <storage_interface>]
    [--vendor-interface <vendor_interface>]
    [--resource-class <resource_class>]
    [--conductor-group <conductor_group>]
    [--automated-clean | --no-automated-clean]
    [--owner <owner>]
    [--lessee <lessee>]
    [--description <description>]
```

-f <FORMATTER>, **--format** <FORMATTER>

the output format, defaults to table

-c COLUMN, **--column** COLUMN

specify the column(s) to include, can be repeated to show multiple columns

--noindent

whether to disable indenting the JSON

--prefix <PREFIX>

add a prefix to all variable names

--max-width <integer>

Maximum display width, <1 to disable. You can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.

--fit-width

Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable

--print-empty

Print empty table if there is no data to show.

--chassis-uuid <chassis>

UUID of the chassis that this node belongs to.

--driver <driver>

Driver used to control the node [REQUIRED].

--driver-info <key=value>

Key/value pair used by the driver, such as out-of-band management credentials. Can be specified multiple times.

--property <key=value>

Key/value pair describing the physical characteristics of the node. This is exported to Nova and used by the scheduler. Can be specified multiple times.

--extra <key=value>

Record arbitrary key/value metadata. Can be specified multiple times.

--uuid <uuid>

Unique UUID for the node.

--name <name>

Unique name for the node.

--bios-interface <bios_interface>

BIOS interface used by the nodes driver. This is only applicable when the specified driver is a hardware type.

--boot-interface <boot_interface>

Boot interface used by the nodes driver. This is only applicable when the specified driver is a hardware type.

--console-interface <console_interface>

Console interface used by the nodes driver. This is only applicable when the specified driver is a hardware type.

--deploy-interface <deploy_interface>

Deploy interface used by the nodes driver. This is only applicable when the specified driver is a hardware type.

--inspect-interface <inspect_interface>

Inspect interface used by the nodes driver. This is only applicable when the specified driver is a hardware type.

--management-interface <management_interface>

Management interface used by the nodes driver. This is only applicable when the specified driver is a hardware type.

--network-data <network data>

JSON string or a YAML file or - for stdin to read static network configuration for the baremetal node associated with this ironic node. Format of this file should comply with Nova network data metadata (network_data.json). Depending on ironic boot interface capabilities being used, network configuration may or may not been served to the node for offline network configuration.

--network-interface <network_interface>

Network interface used for switching node to cleaning/provisioning networks.

--power-interface <power_interface>

Power interface used by the nodes driver. This is only applicable when the specified driver is a hardware type.

--raid-interface <raid_interface>

RAID interface used by the nodes driver. This is only applicable when the specified driver is a hardware type.

--rescue-interface <rescue_interface>

Rescue interface used by the nodes driver. This is only applicable when the specified driver is a hardware type.

--storage-interface <storage_interface>

Storage interface used by the nodes driver.

--vendor-interface <vendor_interface>

Vendor interface used by the nodes driver. This is only applicable when the specified driver is a hardware type.

--resource-class <resource_class>

Resource class for mapping nodes to Nova flavors

--conductor-group <conductor_group>

Conductor group the node will belong to

--automated-clean

Enable automated cleaning for the node

--no-automated-clean

Explicitly disable automated cleaning for the node

--owner <owner>

Owner of the node.

--lessee <lessee>

Lessee of the node.

--description <description>

Description for the node.

This command is provided by the python-ironicclient plugin.

baremetal node delete

Unregister baremetal node(s)

```
openstack baremetal node delete <node> [<node> ...]
```

node

Node(s) to delete (name or UUID)

This command is provided by the python-ironicclient plugin.

baremetal node deploy

Set provision state of baremetal node to deploy

```
openstack baremetal node deploy
    [--wait [<time-out>]]
    [--config-drive <config-drive>]
    [--deploy-steps <deploy-steps>]
    <node>
```

--wait <time-out>

Wait for a node to reach the desired state, active. Optionally takes a timeout value (in seconds). The default value is 0, meaning it will wait indefinitely.

--config-drive <config-drive>

A gzipped, base64-encoded configuration drive string OR the path to the configuration drive file OR the path to a directory containing the config drive files OR a JSON object to build config drive from. In case its a directory, a config drive will be generated from it. In case its a JSON object with optional keys *meta_data*, *user_data* and *network_data*, a config drive will be generated on the server side (see the bare metal API reference for more details).

--deploy-steps <deploy-steps>

The deploy steps. May be the path to a YAML file containing the deploy steps; OR -, with the deploy steps being read from standard input; OR a JSON string. The value should be a list of deploy-step dictionaries; each dictionary should have keys interface and step, and optional key args.

node

Name or UUID of the node.

This command is provided by the python-ironicclient plugin.

baremetal node history get

Get history event for a baremetal node.

```
openstack baremetal node history get
    [-f {json,shell,table,value,yaml}]
    [-c COLUMN]
    [--noindent]
    [--prefix PREFIX]
    [--max-width <integer>]
    [--fit-width]
    [--print-empty]
    <node>
    <event>
```

-f <FORMATTER>, **--format** <FORMATTER>

the output format, defaults to table

-c COLUMN, **--column** COLUMN

specify the column(s) to include, can be repeated to show multiple columns

--noindent

whether to disable indenting the JSON

--prefix `<PREFIX>`

add a prefix to all variable names

--max-width `<integer>`

Maximum display width, <1 to disable. You can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.

--fit-width

Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable

--print-empty

Print empty table if there is no data to show.

node

Name or UUID of the node.

event

UUID of the event.

This command is provided by the python-ironicclient plugin.

baremetal node history list

Get history events for a baremetal node.

```
openstack baremetal node history list
    [-f {csv,json,table,value,yaml}]
    [-c COLUMN]
    [--quote {all,minimal,none,nonnumeric}]
    [--noindent]
    [--max-width <integer>]
    [--fit-width]
    [--print-empty]
    [--sort-column SORT_COLUMN]
    [--sort-ascending | --sort-descending]
    [--long]
    <node>
```

-f `<FORMATTER>`, **--format** `<FORMATTER>`

the output format, defaults to table

-c `COLUMN`, **--column** `COLUMN`

specify the column(s) to include, can be repeated to show multiple columns

--quote `<QUOTE_MODE>`

when to include quotes, defaults to nonnumeric

--noindent

whether to disable indenting the JSON

--max-width <integer>
Maximum display width, <1 to disable. You can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.

--fit-width
Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable

--print-empty
Print empty table if there is no data to show.

--sort-column SORT_COLUMN
specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated

--sort-ascending
sort the column(s) in ascending order

--sort-descending
sort the column(s) in descending order

--long
Show detailed information about the BIOS settings.

node
Name or UUID of the node.

This command is provided by the python-ironicclient plugin.

baremetal node inject nmi

Inject NMI to baremetal node

```
openstack baremetal node inject nmi <node>
```

node
Name or UUID of the node.

This command is provided by the python-ironicclient plugin.

baremetal node inspect

Set provision state of baremetal node to inspect

```
openstack baremetal node inspect [--wait [<time-out>]] <node>
```

--wait <time-out>
Wait for a node to reach the desired state, manageable. Optionally takes a timeout value (in seconds). The default value is 0, meaning it will wait indefinitely.

node
Name or UUID of the node.

This command is provided by the python-ironicclient plugin.

baremetal node list

List baremetal nodes

```
openstack baremetal node list
    [-f {csv,json,table,value,yaml}]
    [-c COLUMN]
    [--quote {all,minimal,none,nonnumeric}]
    [--noindent]
    [--max-width <integer>]
    [--fit-width]
    [--print-empty]
    [--sort-column SORT_COLUMN]
    [--sort-ascending | --sort-descending]
    [--limit <limit>]
    [--marker <node>]
    [--sort <key>[:<direction>]]
    [--maintenance | --no-maintenance]
    [--retired | --no-retired]
    [--fault <fault>]
    [--associated | --unassociated]
    [--provision-state <provision state>]
    [--driver <driver>]
    [--resource-class <resource class>]
    [--conductor-group <conductor_group>]
    [--conductor <conductor>]
    [--chassis <chassis UUID>]
    [--owner <owner>]
    [--lessee <lessee>]
    [--description-contains <description_contains>]
    [--long | --fields <field> [<field> ...]]
```

-f <FORMATTER>, **--format** <FORMATTER>

the output format, defaults to table

-c COLUMN, **--column** COLUMN

specify the column(s) to include, can be repeated to show multiple columns

--quote <QUOTE_MODE>

when to include quotes, defaults to nonnumeric

--noindent

whether to disable indenting the JSON

--max-width <integer>

Maximum display width, <1 to disable. You can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.

--fit-width

Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable

--print-empty
Print empty table if there is no data to show.

--sort-column SORT_COLUMN
specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated

--sort-ascending
sort the column(s) in ascending order

--sort-descending
sort the column(s) in descending order

--limit <limit>
Maximum number of nodes to return per request, 0 for no limit. Default is the maximum number used by the Baremetal API Service.

--marker <node>
Node UUID (for example, of the last node in the list from a previous request). Returns the list of nodes after this UUID.

--sort <key>[:<direction>]
Sort output by specified node fields and directions (asc or desc) (default: asc). Multiple fields and directions can be specified, separated by comma.

--maintenance
Limit list to nodes in maintenance mode

--no-maintenance
Limit list to nodes not in maintenance mode

--retired
Limit list to retired nodes.

--no-retired
Limit list to not retired nodes.

--fault <fault>
List nodes in specified fault.

--associated
List only nodes associated with an instance.

--unassociated
List only nodes not associated with an instance.

--provision-state <provision state>
List nodes in specified provision state.

--driver <driver>
Limit list to nodes with driver <driver>

--resource-class <resource class>
Limit list to nodes with resource class <resource class>

--conductor-group <conductor_group>

Limit list to nodes with conductor group <conductor group>

--conductor <conductor>

Limit list to nodes with conductor <conductor>

--chassis <chassis UUID>

Limit list to nodes of this chassis

--owner <owner>

Limit list to nodes with owner <owner>

--lessee <lessee>

Limit list to nodes with lessee <lessee>

--description-contains <description_contains>

Limit list to nodes with description contains <description_contains>

--long

Show detailed information about the nodes.

--fields <field>

One or more node fields. Only these fields will be fetched from the server. Can not be used when long is specified.

This command is provided by the python-ironicclient plugin.

baremetal node maintenance set

Set baremetal node to maintenance mode

```
openstack baremetal node maintenance set [--reason <reason>] <node>
```

--reason <reason>

Reason for setting maintenance mode.

node

Name or UUID of the node.

This command is provided by the python-ironicclient plugin.

baremetal node maintenance unset

Unset baremetal node from maintenance mode

```
openstack baremetal node maintenance unset <node>
```

node

Name or UUID of the node.

This command is provided by the python-ironicclient plugin.

baremetal node manage

Set provision state of baremetal node to manage

```
openstack baremetal node manage [--wait [<time-out>]] <node>
```

--wait <time-out>

Wait for a node to reach the desired state, manageable. Optionally takes a timeout value (in seconds). The default value is 0, meaning it will wait indefinitely.

node

Name or UUID of the node.

This command is provided by the python-ironicclient plugin.

baremetal node passthru call

Call a vendor passthu method for a node

```
openstack baremetal node passthru call
    [--arg <key=value>]
    [--http-method <http-method>]
    <node>
    <method>
```

--arg <key=value>

Argument to pass to the passthru method (repeat option to specify multiple arguments)

--http-method <http-method>

The HTTP method to use in the passthru request. One of DELETE, GET, PATCH, POST, PUT. Defaults to POST.

node

Name or UUID of the node

method

Vendor passthru method to be executed

This command is provided by the python-ironicclient plugin.

baremetal node passthru list

List vendor passthru methods for a node

```
openstack baremetal node passthru list
    [-f {csv,json,table,value,yaml}]
    [-c COLUMN]
    [--quote {all,minimal,none,nonnumeric}]
    [--noindent]
    [--max-width <integer>]
    [--fit-width]
```

(continues on next page)

(continued from previous page)

```
    [--print-empty]
    [--sort-column SORT_COLUMN]
    [--sort-ascending | --sort-descending]
    <node>
```

-f <FORMATTER>, **--format** <FORMATTER>
    the output format, defaults to table

-c COLUMN, **--column** COLUMN
    specify the column(s) to include, can be repeated to show multiple columns

--quote <QUOTE_MODE>
    when to include quotes, defaults to nonnumeric

--noindent
    whether to disable indenting the JSON

--max-width <integer>
    Maximum display width, <1 to disable. You can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.

--fit-width
    Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable

--print-empty
    Print empty table if there is no data to show.

--sort-column SORT_COLUMN
    specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated

--sort-ascending
    sort the column(s) in ascending order

--sort-descending
    sort the column(s) in descending order

node
    Name or UUID of the node

This command is provided by the python-ironicclient plugin.

baremetal node power off

Power off a node

```
openstack baremetal node power off
    [--power-timeout <power-timeout>]
    [--soft]
    <node>
```

--power-timeout <power-timeout>

Timeout (in seconds, positive integer) to wait for the target power state before erroring out.

--soft

Request graceful power-off.

node

Name or UUID of the node.

This command is provided by the python-ironicclient plugin.

baremetal node power on

Power on a node

```
openstack baremetal node power on
    [--power-timeout <power-timeout>]
    <node>
```

--power-timeout <power-timeout>

Timeout (in seconds, positive integer) to wait for the target power state before erroring out.

node

Name or UUID of the node.

This command is provided by the python-ironicclient plugin.

baremetal node provide

Set provision state of baremetal node to provide

```
openstack baremetal node provide [--wait [<time-out>]] <node>
```

--wait <time-out>

Wait for a node to reach the desired state, available. Optionally takes a timeout value (in seconds). The default value is 0, meaning it will wait indefinitely.

node

Name or UUID of the node.

This command is provided by the python-ironicclient plugin.

baremetal node reboot

Reboot baremetal node

```
openstack baremetal node reboot
    [--soft]
    [--power-timeout <power-timeout>]
    <node>
```

--soft

Request Graceful reboot.

--power-timeout <power-timeout>

Timeout (in seconds, positive integer) to wait for the target power state before erroring out.

node

Name or UUID of the node.

This command is provided by the python-ironicclient plugin.

baremetal node rebuild

Set provision state of baremetal node to rebuild

```
openstack baremetal node rebuild
    [--wait [<time-out>]]
    [--config-drive <config-drive>]
    [--deploy-steps <deploy-steps>]
    <node>
```

--wait <time-out>

Wait for a node to reach the desired state, active. Optionally takes a timeout value (in seconds). The default value is 0, meaning it will wait indefinitely.

--config-drive <config-drive>

A gzipped, base64-encoded configuration drive string OR the path to the configuration drive file OR the path to a directory containing the config drive files OR a JSON object to build config drive from. In case its a directory, a config drive will be generated from it. In case its a JSON object with optional keys *meta_data*, *user_data* and *network_data*, a config drive will be generated on the server side (see the bare metal API reference for more details).

--deploy-steps <deploy-steps>

The deploy steps in JSON format. May be the path to a file containing the deploy steps; OR -, with the deploy steps being read from standard input; OR a string. The value should be a list of deploy-step dictionaries; each dictionary should have keys interface, step, priority and optional key args.

node

Name or UUID of the node.

This command is provided by the python-ironicclient plugin.

baremetal node remove trait

Remove trait(s) from a node.

```
openstack baremetal node remove trait [--all] <node> [<trait> ...]
```

--all

Remove all traits

node

Name or UUID of the node

trait

Trait(s) to remove

This command is provided by the python-ironicclient plugin.

baremetal node rescue

Set provision state of baremetal node to rescue

```
openstack baremetal node rescue
    [--wait [<time-out>]]
    --rescue-password <rescue-password>
    <node>
```

--wait <time-out>

Wait for a node to reach the desired state, rescue. Optionally takes a timeout value (in seconds). The default value is 0, meaning it will wait indefinitely.

--rescue-password <rescue-password>

The password that will be used to login to the rescue ramdisk. The value should be a non-empty string.

node

Name or UUID of the node.

This command is provided by the python-ironicclient plugin.

baremetal node secure boot off

Turn secure boot off

```
openstack baremetal node secure boot off <node>
```

node

Name or UUID of the node

This command is provided by the python-ironicclient plugin.

baremetal node secure boot on

Turn secure boot on

```
openstack baremetal node secure boot on <node>
```

node

Name or UUID of the node

This command is provided by the python-ironicclient plugin.

baremetal node set

Set baremetal properties

```
openstack baremetal node set
    [--instance-uuid <uuid>]
    [--name <name>]
    [--chassis-uuid <chassis UUID>]
    [--driver <driver>]
    [--bios-interface <bios_interface> | --reset-bios-interface]
    [--boot-interface <boot_interface> | --reset-boot-interface]
    [--console-interface <console_interface> | --reset-console-interface]
    [--deploy-interface <deploy_interface> | --reset-deploy-interface]
    [--inspect-interface <inspect_interface> | --reset-inspect-interface]
    [--management-interface <management_interface> | --reset-management-
↪interface]
    [--network-interface <network_interface> | --reset-network-interface]
    [--network-data <network data>]
    [--power-interface <power_interface> | --reset-power-interface]
    [--raid-interface <raid_interface> | --reset-raid-interface]
    [--rescue-interface <rescue_interface> | --reset-rescue-interface]
    [--storage-interface <storage_interface> | --reset-storage-interface]
    [--vendor-interface <vendor_interface> | --reset-vendor-interface]
    [--reset-interfaces]
    [--resource-class <resource_class>]
    [--conductor-group <conductor_group>]
    [--automated-clean | --no-automated-clean]
    [--protected]
    [--protected-reason <protected_reason>]
    [--retired]
    [--retired-reason <retired_reason>]
    [--target-raid-config <target_raid_config>]
    [--property <key=value>]
    [--extra <key=value>]
    [--driver-info <key=value>]
    [--instance-info <key=value>]
    [--owner <owner>]
    [--lessee <lessee>]
    [--description <description>]
    <node>
```

--instance-uuid <uuid>

Set instance UUID of node to <uuid>

--name <name>

Set the name of the node

--chassis-uuid <chassis UUID>

Set the chassis for the node

--driver <driver>

Set the driver for the node

--bios-interface <bios_interface>
Set the BIOS interface for the node

--reset-bios-interface
Reset the BIOS interface to its hardware type default

--boot-interface <boot_interface>
Set the boot interface for the node

--reset-boot-interface
Reset the boot interface to its hardware type default

--console-interface <console_interface>
Set the console interface for the node

--reset-console-interface
Reset the console interface to its hardware type default

--deploy-interface <deploy_interface>
Set the deploy interface for the node

--reset-deploy-interface
Reset the deploy interface to its hardware type default

--inspect-interface <inspect_interface>
Set the inspect interface for the node

--reset-inspect-interface
Reset the inspect interface to its hardware type default

--management-interface <management_interface>
Set the management interface for the node

--reset-management-interface
Reset the management interface to its hardware type default

--network-interface <network_interface>
Set the network interface for the node

--reset-network-interface
Reset the network interface to its hardware type default

--network-data <network data>
JSON string or a YAML file or - for stdin to read static network configuration for the baremetal node associated with this ironic node. Format of this file should comply with Nova network data metadata (network_data.json). Depending on ironic boot interface capabilities being used, network configuration may or may not been served to the node for offline network configuration.

--power-interface <power_interface>
Set the power interface for the node

--reset-power-interface
Reset the power interface to its hardware type default

--raid-interface <raid_interface>
Set the RAID interface for the node

--reset-raid-interface
Reset the RAID interface to its hardware type default

--rescue-interface <rescue_interface>
Set the rescue interface for the node

--reset-rescue-interface
Reset the rescue interface to its hardware type default

--storage-interface <storage_interface>
Set the storage interface for the node

--reset-storage-interface
Reset the storage interface to its hardware type default

--vendor-interface <vendor_interface>
Set the vendor interface for the node

--reset-vendor-interface
Reset the vendor interface to its hardware type default

--reset-interfaces
Reset all interfaces not specified explicitly to their default implementations. Only valid with driver.

--resource-class <resource_class>
Set the resource class for the node

--conductor-group <conductor_group>
Set the conductor group for the node

--automated-clean
Enable automated cleaning for the node

--no-automated-clean
Explicitly disable automated cleaning for the node

--protected
Mark the node as protected

--protected-reason <protected_reason>
Set the reason of marking the node as protected

--retired
Mark the node as retired

--retired-reason <retired_reason>
Set the reason of marking the node as retired

--target-raid-config <target_raid_config>
Set the target RAID configuration (JSON) for the node. This can be one of: 1. a file containing YAML data of the RAID configuration; 2. - to read the contents from standard input; or 3. a valid JSON string.

--property <key=value>
Property to set on this baremetal node (repeat option to set multiple properties)

--extra <key=value>

Extra to set on this baremetal node (repeat option to set multiple extras)

--driver-info <key=value>

Driver information to set on this baremetal node (repeat option to set multiple driver infos)

--instance-info <key=value>

Instance information to set on this baremetal node (repeat option to set multiple instance infos)

--owner <owner>

Set the owner for the node

--lessee <lessee>

Set the lessee for the node

--description <description>

Set the description for the node

node

Name or UUID of the node.

This command is provided by the python-ironicclient plugin.

baremetal node show

Show baremetal node details

```
openstack baremetal node show
    [-f {json,shell,table,value,yaml}]
    [-c COLUMN]
    [--noindent]
    [--prefix PREFIX]
    [--max-width <integer>]
    [--fit-width]
    [--print-empty]
    [--instance]
    [--fields <field> [<field> ...]]
    <node>
```

-f <FORMATTER>, **--format** <FORMATTER>

the output format, defaults to table

-c COLUMN, **--column** COLUMN

specify the column(s) to include, can be repeated to show multiple columns

--noindent

whether to disable indenting the JSON

--prefix <PREFIX>

add a prefix to all variable names

--max-width <integer>

Maximum display width, <1 to disable. You can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.

--fit-width

Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable

--print-empty

Print empty table if there is no data to show.

--instance

<node> is an instance UUID.

--fields <field>

One or more node fields. Only these fields will be fetched from the server.

node

Name or UUID of the node (or instance UUID if instance is specified)

This command is provided by the python-ironicclient plugin.

baremetal node trait list

List a nodes traits.

```
openstack baremetal node trait list
    [-f {csv,json,table,value,yaml}]
    [-c COLUMN]
    [--quote {all,minimal,none,nonnumeric}]
    [--noindent]
    [--max-width <integer>]
    [--fit-width]
    [--print-empty]
    [--sort-column SORT_COLUMN]
    [--sort-ascending | --sort-descending]
    <node>
```

-f <FORMATTER>, **--format** <FORMATTER>

the output format, defaults to table

-c COLUMN, **--column** COLUMN

specify the column(s) to include, can be repeated to show multiple columns

--quote <QUOTE_MODE>

when to include quotes, defaults to nonnumeric

--noindent

whether to disable indenting the JSON

--max-width <integer>

Maximum display width, <1 to disable. You can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.

--fit-width

Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable

--print-empty
Print empty table if there is no data to show.

--sort-column SORT_COLUMN
specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated

--sort-ascending
sort the column(s) in ascending order

--sort-descending
sort the column(s) in descending order

node
Name or UUID of the node

This command is provided by the python-ironicclient plugin.

baremetal node undeploy

Set provision state of baremetal node to deleted

```
openstack baremetal node undeploy [--wait [<time-out>]] <node>
```

--wait <time-out>
Wait for a node to reach the desired state, available. Optionally takes a timeout value (in seconds). The default value is 0, meaning it will wait indefinitely.

node
Name or UUID of the node.

This command is provided by the python-ironicclient plugin.

baremetal node unrescue

Set provision state of baremetal node to unrescue

```
openstack baremetal node unrescue [--wait [<time-out>]] <node>
```

--wait <time-out>
Wait for a node to reach the desired state, active. Optionally takes a timeout value (in seconds). The default value is 0, meaning it will wait indefinitely.

node
Name or UUID of the node.

This command is provided by the python-ironicclient plugin.

baremetal node unset

Unset baremetal properties

```
openstack baremetal node unset
    [--instance-uuid]
    [--name]
    [--resource-class]
    [--target-raid-config]
    [--property <key>]
    [--extra <key>]
    [--driver-info <key>]
    [--instance-info <key>]
    [--chassis-uuid]
    [--bios-interface]
    [--boot-interface]
    [--console-interface]
    [--deploy-interface]
    [--inspect-interface]
    [--network-data]
    [--management-interface]
    [--network-interface]
    [--power-interface]
    [--raid-interface]
    [--rescue-interface]
    [--storage-interface]
    [--vendor-interface]
    [--conductor-group]
    [--automated-clean]
    [--protected]
    [--protected-reason]
    [--retired]
    [--retired-reason]
    [--owner]
    [--lessee]
    [--description]
    <node>
```

--instance-uuid

Unset instance UUID on this baremetal node

--name

Unset the name of the node

--resource-class

Unset the resource class of the node

--target-raid-config

Unset the target RAID configuration of the node

--property <key>

Property to unset on this baremetal node (repeat option to unset multiple properties)

`--extra <key>`
Extra to unset on this baremetal node (repeat option to unset multiple extras)

`--driver-info <key>`
Driver information to unset on this baremetal node (repeat option to unset multiple driver informations)

`--instance-info <key>`
Instance information to unset on this baremetal node (repeat option to unset multiple instance informations)

`--chassis-uuid`
Unset chassis UUID on this baremetal node

`--bios-interface`
Unset BIOS interface on this baremetal node

`--boot-interface`
Unset boot interface on this baremetal node

`--console-interface`
Unset console interface on this baremetal node

`--deploy-interface`
Unset deploy interface on this baremetal node

`--inspect-interface`
Unset inspect interface on this baremetal node

`--network-data`
Unset network data on this baremetal port.

`--management-interface`
Unset management interface on this baremetal node

`--network-interface`
Unset network interface on this baremetal node

`--power-interface`
Unset power interface on this baremetal node

`--raid-interface`
Unset RAID interface on this baremetal node

`--rescue-interface`
Unset rescue interface on this baremetal node

`--storage-interface`
Unset storage interface on this baremetal node

`--vendor-interface`
Unset vendor interface on this baremetal node

`--conductor-group`
Unset conductor group for this baremetal node (the default group will be used)

--automated-clean

Unset automated clean option on this baremetal node (the value from configuration will be used)

--protected

Unset the protected flag on the node

--protected-reason

Unset the protected reason (gets unset automatically when protected is unset)

--retired

Unset the retired flag on the node

--retired-reason

Unset the retired reason (gets unset automatically when retired is unset)

--owner

Unset the owner field of the node

--lessee

Unset the lessee field of the node

--description

Unset the description field of the node

node

Name or UUID of the node.

This command is provided by the python-ironicclient plugin.

baremetal node validate

Validate a nodes driver interfaces

```
openstack baremetal node validate
    [-f {csv,json,table,value,yaml}]
    [-c COLUMN]
    [--quote {all,minimal,none,nonnumeric}]
    [--noindent]
    [--max-width <integer>]
    [--fit-width]
    [--print-empty]
    [--sort-column SORT_COLUMN]
    [--sort-ascending | --sort-descending]
    <node>
```

-f <FORMATTER>, **--format** <FORMATTER>

the output format, defaults to table

-c COLUMN, **--column** COLUMN

specify the column(s) to include, can be repeated to show multiple columns

--quote <QUOTE_MODE>

when to include quotes, defaults to nonnumeric

--noindent

whether to disable indenting the JSON

--max-width <integer>

Maximum display width, <1 to disable. You can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.

--fit-width

Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable

--print-empty

Print empty table if there is no data to show.

--sort-column SORT_COLUMN

specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated

--sort-ascending

sort the column(s) in ascending order

--sort-descending

sort the column(s) in descending order

node

Name or UUID of the node

This command is provided by the python-ironicclient plugin.

baremetal node vif attach

Attach VIF to a given node

```
openstack baremetal node vif attach
    [--port-uuid <port-uuid>]
    [--vif-info <key=value>]
    <node>
    <vif-id>
```

--port-uuid <port-uuid>

UUID of the baremetal port to attach the VIF to.

--vif-info <key=value>

Record arbitrary key/value metadata. Can be specified multiple times. The mandatory id parameter cannot be specified as a key.

node

Name or UUID of the node

vif-id

Name or UUID of the VIF to attach to a node.

This command is provided by the python-ironicclient plugin.

baremetal node vif detach

Detach VIF from a given node

```
openstack baremetal node vif detach <node> <vif-id>
```

node

Name or UUID of the node

vif-id

Name or UUID of the VIF to detach from a node.

This command is provided by the python-ironicclient plugin.

baremetal node vif list

Show attached VIFs for a node

```
openstack baremetal node vif list
    [-f {csv,json,table,value,yaml}]
    [-c COLUMN]
    [--quote {all,minimal,none,nonnumeric}]
    [--noindent]
    [--max-width <integer>]
    [--fit-width]
    [--print-empty]
    [--sort-column SORT_COLUMN]
    [--sort-ascending | --sort-descending]
    <node>
```

-f <FORMATTER>, **--format** <FORMATTER>

the output format, defaults to table

-c COLUMN, **--column** COLUMN

specify the column(s) to include, can be repeated to show multiple columns

--quote <QUOTE_MODE>

when to include quotes, defaults to nonnumeric

--noindent

whether to disable indenting the JSON

--max-width <integer>

Maximum display width, <1 to disable. You can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.

--fit-width

Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable

--print-empty

Print empty table if there is no data to show.

--sort-column SORT_COLUMN

specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated

--sort-ascending

sort the column(s) in ascending order

--sort-descending

sort the column(s) in descending order

node

Name or UUID of the node

This command is provided by the python-ironicclient plugin.

baremetal port

baremetal port create

Create a new port

```
openstack baremetal port create
    [-f {json,shell,table,value,yaml}]
    [-c COLUMN]
    [--noindent]
    [--prefix PREFIX]
    [--max-width <integer>]
    [--fit-width]
    [--print-empty]
    --node <uuid>
    [--uuid <uuid>]
    [--extra <key=value>]
    [--local-link-connection <key=value>]
    [-l <key=value>]
    [--pxe-enabled <boolean>]
    [--port-group <uuid>]
    [--physical-network <physical network>]
    [--is-smartnic]
    <address>
```

-f <FORMATTER>, **--format** <FORMATTER>

the output format, defaults to table

-c COLUMN, **--column** COLUMN

specify the column(s) to include, can be repeated to show multiple columns

--noindent

whether to disable indenting the JSON

--prefix <PREFIX>

add a prefix to all variable names

--max-width <integer>

Maximum display width, <1 to disable. You can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.

--fit-width

Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable

--print-empty

Print empty table if there is no data to show.

--node <uuid>

UUID of the node that this port belongs to.

--uuid <uuid>

UUID of the port.

--extra <key=value>

Record arbitrary key/value metadata. Argument can be specified multiple times.

--local-link-connection <key=value>

Key/value metadata describing Local link connection information. Valid keys are switch_info, switch_id, port_id and hostname. The keys switch_id and port_id are required. In case of a Smart NIC port, the required keys are port_id and hostname. Argument can be specified multiple times.

-l <key=value>

DEPRECATED. Please use local-link-connection instead. Key/value metadata describing Local link connection information. Valid keys are switch_info, switch_id, and port_id. The keys switch_id and port_id are required. Can be specified multiple times.

--pxe-enabled <boolean>

Indicates whether this Port should be used when PXE booting this Node.

--port-group <uuid>

UUID of the port group that this port belongs to.

--physical-network <physical network>

Name of the physical network to which this port is connected.

--is-smartnic

Indicates whether this Port is a Smart NIC port

address

MAC address for this port.

This command is provided by the python-ironicclient plugin.

baremetal port delete

Delete port(s).

```
openstack baremetal port delete <port> [<port> ...]
```

port

UUID(s) of the port(s) to delete.

This command is provided by the python-ironicclient plugin.

baremetal port group create

Create a new baremetal port group.

```
openstack baremetal port group create
    [-f {json,shell,table,value,yaml}]
    [-c COLUMN]
    [--noindent]
    [--prefix PREFIX]
    [--max-width <integer>]
    [--fit-width]
    [--print-empty]
    --node <uuid>
    [--address <mac-address>]
    [--name NAME]
    [--uuid UUID]
    [--extra <key=value>]
    [--mode MODE]
    [--property <key=value>]
    [--support-standalone-ports | --unsupport-standalone-ports]
```

-f <FORMATTER>, **--format** <FORMATTER>

the output format, defaults to table

-c COLUMN, **--column** COLUMN

specify the column(s) to include, can be repeated to show multiple columns

--noindent

whether to disable indenting the JSON

--prefix <PREFIX>

add a prefix to all variable names

--max-width <integer>

Maximum display width, <1 to disable. You can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.

--fit-width

Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable

--print-empty

Print empty table if there is no data to show.

--node <uuid>

UUID of the node that this port group belongs to.

--address <mac-address>

MAC address for this port group.

--name <NAME>

Name of the port group.

--uuid <UUID>

UUID of the port group.

--extra <key=value>

Record arbitrary key/value metadata. Can be specified multiple times.

--mode <MODE>

Mode of the port group. For possible values, refer to https://www.kernel.org/doc/Documentation/networking/bonding.txt.

--property <key=value>

Key/value property related to this port groups configuration. Can be specified multiple times.

--support-standalone-ports

Ports that are members of this port group can be used as stand-alone ports. (default)

--unsupport-standalone-ports

Ports that are members of this port group cannot be used as stand-alone ports.

This command is provided by the python-ironicclient plugin.

baremetal port group delete

Unregister baremetal port group(s).

```
openstack baremetal port group delete <port group> [<port group> ...]
```

port group

Port group(s) to delete (name or UUID).

This command is provided by the python-ironicclient plugin.

baremetal port group list

List baremetal port groups.

```
openstack baremetal port group list
    [-f {csv,json,table,value,yaml}]
    [-c COLUMN]
    [--quote {all,minimal,none,nonnumeric}]
```

(continues on next page)

(continued from previous page)

```
    [--noindent]
    [--max-width <integer>]
    [--fit-width]
    [--print-empty]
    [--sort-column SORT_COLUMN]
    [--sort-ascending | --sort-descending]
    [--limit <limit>]
    [--marker <port group>]
    [--sort <key>[:<direction>]]
    [--address <mac-address>]
    [--node <node>]
    [--long | --fields <field> [<field> ...]]
```

`-f` `<FORMATTER>`, **`--format`** `<FORMATTER>`

the output format, defaults to table

`-c` `COLUMN`, **`--column`** `COLUMN`

specify the column(s) to include, can be repeated to show multiple columns

`--quote` `<QUOTE_MODE>`

when to include quotes, defaults to nonnumeric

`--noindent`

whether to disable indenting the JSON

`--max-width` `<integer>`

Maximum display width, <1 to disable. You can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.

`--fit-width`

Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable

`--print-empty`

Print empty table if there is no data to show.

`--sort-column` `SORT_COLUMN`

specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated

`--sort-ascending`

sort the column(s) in ascending order

`--sort-descending`

sort the column(s) in descending order

`--limit` `<limit>`

Maximum number of port groups to return per request, 0 for no limit. Default is the maximum number used by the Baremetal API Service.

`--marker` `<port group>`

Port group UUID (for example, of the last port group in the list from a previous request). Returns the list of port groups after this UUID.

--sort <key>[:<direction>]

Sort output by specified port group fields and directions (asc or desc) (default: asc). Multiple fields and directions can be specified, separated by comma.

--address <mac-address>

Only show information for the port group with this MAC address.

--node <node>

Only list port groups of this node (name or UUID).

--long

Show detailed information about the port groups.

--fields <field>

One or more port group fields. Only these fields will be fetched from the server. Can not be used when long is specified.

This command is provided by the python-ironicclient plugin.

baremetal port group set

Set baremetal port group properties.

```
openstack baremetal port group set
    [--node <uuid>]
    [--address <mac-address>]
    [--name <name>]
    [--extra <key=value>]
    [--mode MODE]
    [--property <key=value>]
    [--support-standalone-ports | --unsupport-standalone-ports]
    <port
    group>
```

--node <uuid>

Update UUID of the node that this port group belongs to.

--address <mac-address>

MAC address for this port group.

--name <name>

Name of the port group.

--extra <key=value>

Extra to set on this baremetal port group (repeat option to set multiple extras).

--mode <MODE>

Mode of the port group. For possible values, refer to https://www.kernel.org/doc/Documentation/networking/bonding.txt.

--property <key=value>

Key/value property related to this port groups configuration (repeat option to set multiple properties).

--support-standalone-ports
Ports that are members of this port group can be used as stand-alone ports.

--unsupport-standalone-ports
Ports that are members of this port group cannot be used as stand-alone ports.

port group
Name or UUID of the port group.

This command is provided by the python-ironicclient plugin.

baremetal port group show

Show baremetal port group details.

```
openstack baremetal port group show
    [-f {json,shell,table,value,yaml}]
    [-c COLUMN]
    [--noindent]
    [--prefix PREFIX]
    [--max-width <integer>]
    [--fit-width]
    [--print-empty]
    [--address]
    [--fields <field> [<field> ...]]
    <id>
```

-f <FORMATTER>, **--format** <FORMATTER>
the output format, defaults to table

-c COLUMN, **--column** COLUMN
specify the column(s) to include, can be repeated to show multiple columns

--noindent
whether to disable indenting the JSON

--prefix <PREFIX>
add a prefix to all variable names

--max-width <integer>
Maximum display width, <1 to disable. You can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.

--fit-width
Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable

--print-empty
Print empty table if there is no data to show.

--address
<id> is the MAC address (instead of UUID or name) of the port group.

--fields <field>

One or more port group fields. Only these fields will be fetched from the server.

id

UUID or name of the port group (or MAC address if address is specified).

This command is provided by the python-ironicclient plugin.

baremetal port group unset

Unset baremetal port group properties.

```
openstack baremetal port group unset
    [--name]
    [--address]
    [--extra <key>]
    [--property <key>]
    <port
    group>
```

--name

Unset the name of the port group.

--address

Unset the address of the port group.

--extra <key>

Extra to unset on this baremetal port group (repeat option to unset multiple extras).

--property <key>

Property to unset on this baremetal port group (repeat option to unset multiple properties).

port group

Name or UUID of the port group.

This command is provided by the python-ironicclient plugin.

baremetal port list

List baremetal ports.

```
openstack baremetal port list
    [-f {csv,json,table,value,yaml}]
    [-c COLUMN]
    [--quote {all,minimal,none,nonnumeric}]
    [--noindent]
    [--max-width <integer>]
    [--fit-width]
    [--print-empty]
    [--sort-column SORT_COLUMN]
    [--sort-ascending | --sort-descending]
```

(continues on next page)

(continued from previous page)

```
    [--address <mac-address>]
    [--node <node>]
    [--port-group <port group>]
    [--limit <limit>]
    [--marker <port>]
    [--sort <key>[:<direction>]]
    [--long | --fields <field> [<field> ...]]
```

-f <FORMATTER>, **--format** <FORMATTER>
the output format, defaults to table

-c COLUMN, **--column** COLUMN
specify the column(s) to include, can be repeated to show multiple columns

--quote <QUOTE_MODE>
when to include quotes, defaults to nonnumeric

--noindent
whether to disable indenting the JSON

--max-width <integer>
Maximum display width, <1 to disable. You can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.

--fit-width
Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable

--print-empty
Print empty table if there is no data to show.

--sort-column SORT_COLUMN
specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated

--sort-ascending
sort the column(s) in ascending order

--sort-descending
sort the column(s) in descending order

--address <mac-address>
Only show information for the port with this MAC address.

--node <node>
Only list ports of this node (name or UUID).

--port-group <port group>
Only list ports of this port group (name or UUID).

--limit <limit>
Maximum number of ports to return per request, 0 for no limit. Default is the maximum number used by the Baremetal API Service.

--marker <port>

Port UUID (for example, of the last port in the list from a previous request). Returns the list of ports after this UUID.

--sort <key>[:<direction>]

Sort output by specified port fields and directions (asc or desc) (default: asc). Multiple fields and directions can be specified, separated by comma.

--long

Show detailed information about ports.

--fields <field>

One or more port fields. Only these fields will be fetched from the server. Can not be used when long is specified.

This command is provided by the python-ironicclient plugin.

baremetal port set

Set baremetal port properties.

```
openstack baremetal port set
    [--node <uuid>]
    [--address <address>]
    [--extra <key=value>]
    [--port-group <uuid>]
    [--local-link-connection <key=value>]
    [--pxe-enabled | --pxe-disabled]
    [--physical-network <physical network>]
    [--is-smartnic]
    <port>
```

--node <uuid>

Set UUID of the node that this port belongs to

--address <address>

Set MAC address for this port

--extra <key=value>

Extra to set on this baremetal port (repeat option to set multiple extras)

--port-group <uuid>

Set UUID of the port group that this port belongs to.

--local-link-connection <key=value>

Key/value metadata describing local link connection information. Valid keys are switch_info, switch_id, port_id and hostname. The keys switch_id and port_id are required. In case of a Smart NIC port, the required keys are port_id and hostname. Argument can be specified multiple times.

--pxe-enabled

Indicates that this port should be used when PXE booting this node (default)

--pxe-disabled

Indicates that this port should not be used when PXE booting this node

--physical-network <physical network>

Set the name of the physical network to which this port is connected.

--is-smartnic

Set port to be Smart NIC port

port

UUID of the port

This command is provided by the python-ironicclient plugin.

baremetal port show

Show baremetal port details.

```
openstack baremetal port show
    [-f {json,shell,table,value,yaml}]
    [-c COLUMN]
    [--noindent]
    [--prefix PREFIX]
    [--max-width <integer>]
    [--fit-width]
    [--print-empty]
    [--address]
    [--fields <field> [<field> ...]]
    <id>
```

-f <FORMATTER>, **--format** <FORMATTER>

the output format, defaults to table

-c COLUMN, **--column** COLUMN

specify the column(s) to include, can be repeated to show multiple columns

--noindent

whether to disable indenting the JSON

--prefix <PREFIX>

add a prefix to all variable names

--max-width <integer>

Maximum display width, <1 to disable. You can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.

--fit-width

Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable

--print-empty

Print empty table if there is no data to show.

--address

<id> is the MAC address (instead of the UUID) of the port.

--fields <field>

One or more port fields. Only these fields will be fetched from the server.

id

UUID of the port (or MAC address if address is specified).

This command is provided by the python-ironicclient plugin.

baremetal port unset

Unset baremetal port properties.

```
openstack baremetal port unset
    [--extra <key>]
    [--port-group]
    [--physical-network]
    [--is-smartnic]
    <port>
```

--extra <key>

Extra to unset on this baremetal port (repeat option to unset multiple extras)

--port-group

Remove port from the port group

--physical-network

Unset the physical network on this baremetal port.

--is-smartnic

Set Port as not Smart NIC port

port

UUID of the port.

This command is provided by the python-ironicclient plugin.

baremetal volume

baremetal volume connector create

Create a new baremetal volume connector.

```
openstack baremetal volume connector create
    [-f {json,shell,table,value,yaml}]
    [-c COLUMN]
    [--noindent]
    [--prefix PREFIX]
    [--max-width <integer>]
```

(continues on next page)

(continued from previous page)

```
    [--fit-width]
    [--print-empty]
    --node <uuid>
    --type <type>
    --connector-id <connector
    id>
    [--uuid <uuid>]
    [--extra <key=value>]
```

-f <FORMATTER>, **--format** <FORMATTER>
  the output format, defaults to table

-c COLUMN, **--column** COLUMN
  specify the column(s) to include, can be repeated to show multiple columns

--noindent
  whether to disable indenting the JSON

--prefix <PREFIX>
  add a prefix to all variable names

--max-width <integer>
  Maximum display width, <1 to disable. You can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.

--fit-width
  Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable

--print-empty
  Print empty table if there is no data to show.

--node <uuid>
  UUID of the node that this volume connector belongs to.

--type <type>
  Type of the volume connector. Can be iqn, ip, mac, wwnn, wwpn, port, portgroup.

--connector-id <connector id>
  ID of the volume connector in the specified type. For example, the iSCSI initiator IQN for the node if the type is iqn.

--uuid <uuid>
  UUID of the volume connector.

--extra <key=value>
  Record arbitrary key/value metadata. Can be specified multiple times.

This command is provided by the python-ironicclient plugin.

baremetal volume connector delete

Unregister baremetal volume connector(s).

```
openstack baremetal volume connector delete
    <volume
    connector>
    [<volume connector> ...]
```

volume connector

UUID(s) of the volume connector(s) to delete.

This command is provided by the python-ironicclient plugin.

baremetal volume connector list

List baremetal volume connectors.

```
openstack baremetal volume connector list
    [-f {csv,json,table,value,yaml}]
    [-c COLUMN]
    [--quote {all,minimal,none,nonnumeric}]
    [--noindent]
    [--max-width <integer>]
    [--fit-width]
    [--print-empty]
    [--sort-column SORT_COLUMN]
    [--sort-ascending | --sort-descending]
    [--node <node>]
    [--limit <limit>]
    [--marker <volume connector>]
    [--sort <key>[:<direction>]]
    [--long | --fields <field> [<field> ...]]
```

-f <FORMATTER>, **--format** <FORMATTER>

the output format, defaults to table

-c COLUMN, **--column** COLUMN

specify the column(s) to include, can be repeated to show multiple columns

--quote <QUOTE_MODE>

when to include quotes, defaults to nonnumeric

--noindent

whether to disable indenting the JSON

--max-width <integer>

Maximum display width, <1 to disable. You can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.

--fit-width

Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable

--print-empty

Print empty table if there is no data to show.

--sort-column SORT_COLUMN

specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated

--sort-ascending

sort the column(s) in ascending order

--sort-descending

sort the column(s) in descending order

--node <node>

Only list volume connectors of this node (name or UUID).

--limit <limit>

Maximum number of volume connectors to return per request, 0 for no limit. Default is the maximum number used by the Baremetal API Service.

--marker <volume connector>

Volume connector UUID (for example, of the last volume connector in the list from a previous request). Returns the list of volume connectors after this UUID.

--sort <key>[:<direction>]

Sort output by specified volume connector fields and directions (asc or desc) (default:asc). Multiple fields and directions can be specified, separated by comma.

--long

Show detailed information about volume connectors.

--fields <field>

One or more volume connector fields. Only these fields will be fetched from the server. Can not be used when long is specified.

This command is provided by the python-ironicclient plugin.

baremetal volume connector set

Set baremetal volume connector properties.

```
openstack baremetal volume connector set
    [--node <uuid>]
    [--type <type>]
    [--connector-id <connector id>]
    [--extra <key=value>]
    <volume
    connector>
```

--node <uuid>

UUID of the node that this volume connector belongs to.

--type <type>

Type of the volume connector. Can be iqn, ip, mac, wwnn, wwpn, port, portgroup.

--connector-id <connector id>

ID of the volume connector in the specified type.

--extra <key=value>

Record arbitrary key/value metadata. Can be specified multiple times.

volume connector

UUID of the volume connector.

This command is provided by the python-ironicclient plugin.

baremetal volume connector show

Show baremetal volume connector details.

```
openstack baremetal volume connector show
    [-f {json,shell,table,value,yaml}]
    [-c COLUMN]
    [--noindent]
    [--prefix PREFIX]
    [--max-width <integer>]
    [--fit-width]
    [--print-empty]
    [--fields <field> [<field> ...]]
    <id>
```

-f <FORMATTER>, **--format** <FORMATTER>

the output format, defaults to table

-c COLUMN, **--column** COLUMN

specify the column(s) to include, can be repeated to show multiple columns

--noindent

whether to disable indenting the JSON

--prefix <PREFIX>

add a prefix to all variable names

--max-width <integer>

Maximum display width, <1 to disable. You can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.

--fit-width

Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable

--print-empty
Print empty table if there is no data to show.

--fields <field>
One or more volume connector fields. Only these fields will be fetched from the server.

id
UUID of the volume connector.

This command is provided by the python-ironicclient plugin.

baremetal volume connector unset

Unset baremetal volume connector properties.

```
openstack baremetal volume connector unset
    [--extra <key>]
    <volume
    connector>
```

--extra <key>
Extra to unset (repeat option to unset multiple extras)

volume connector
UUID of the volume connector.

This command is provided by the python-ironicclient plugin.

baremetal volume target create

Create a new baremetal volume target.

```
openstack baremetal volume target create
    [-f {json,shell,table,value,yaml}]
    [-c COLUMN]
    [--noindent]
    [--prefix PREFIX]
    [--max-width <integer>]
    [--fit-width]
    [--print-empty]
    --node <uuid>
    --type <volume
    type>
    [--property <key=value>]
    --boot-index <boot
    index>
    --volume-id <volume
    id>
    [--uuid <uuid>]
    [--extra <key=value>]
```

-f <FORMATTER>, **--format** <FORMATTER>
: the output format, defaults to table

-c COLUMN, **--column** COLUMN
: specify the column(s) to include, can be repeated to show multiple columns

--noindent
: whether to disable indenting the JSON

--prefix <PREFIX>
: add a prefix to all variable names

--max-width <integer>
: Maximum display width, <1 to disable. You can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.

--fit-width
: Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable

--print-empty
: Print empty table if there is no data to show.

--node <uuid>
: UUID of the node that this volume target belongs to.

--type <volume type>
: Type of the volume target, e.g. iscsi, fibre_channel.

--property <key=value>
: Key/value property related to the type of this volume target. Can be specified multiple times.

--boot-index <boot index>
: Boot index of the volume target.

--volume-id <volume id>
: ID of the volume associated with this target.

--uuid <uuid>
: UUID of the volume target.

--extra <key=value>
: Record arbitrary key/value metadata. Can be specified multiple times.

This command is provided by the python-ironicclient plugin.

baremetal volume target delete

Unregister baremetal volume target(s).

```
openstack baremetal volume target delete
    <volume
    target>
    [<volume target> ...]
```

volume target

UUID(s) of the volume target(s) to delete.

This command is provided by the python-ironicclient plugin.

baremetal volume target list

List baremetal volume targets.

```
openstack baremetal volume target list
    [-f {csv,json,table,value,yaml}]
    [-c COLUMN]
    [--quote {all,minimal,none,nonnumeric}]
    [--noindent]
    [--max-width <integer>]
    [--fit-width]
    [--print-empty]
    [--sort-column SORT_COLUMN]
    [--sort-ascending | --sort-descending]
    [--node <node>]
    [--limit <limit>]
    [--marker <volume target>]
    [--sort <key>[:<direction>]]
    [--long | --fields <field> [<field> ...]]
```

-f <FORMATTER>, **--format** <FORMATTER>

the output format, defaults to table

-c COLUMN, **--column** COLUMN

specify the column(s) to include, can be repeated to show multiple columns

--quote <QUOTE_MODE>

when to include quotes, defaults to nonnumeric

--noindent

whether to disable indenting the JSON

--max-width <integer>

Maximum display width, <1 to disable. You can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.

--fit-width

Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable

--print-empty

Print empty table if there is no data to show.

--sort-column SORT_COLUMN

specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated

--sort-ascending

sort the column(s) in ascending order

--sort-descending

sort the column(s) in descending order

--node `<node>`

Only list volume targets of this node (name or UUID).

--limit `<limit>`

Maximum number of volume targets to return per request, 0 for no limit. Default is the maximum number used by the Baremetal API Service.

--marker `<volume target>`

Volume target UUID (for example, of the last volume target in the list from a previous request). Returns the list of volume targets after this UUID.

--sort `<key>[:<direction>]`

Sort output by specified volume target fields and directions (asc or desc) (default:asc). Multiple fields and directions can be specified, separated by comma.

--long

Show detailed information about volume targets.

--fields `<field>`

One or more volume target fields. Only these fields will be fetched from the server. Can not be used when long is specified.

This command is provided by the python-ironicclient plugin.

baremetal volume target set

Set baremetal volume target properties.

```
openstack baremetal volume target set
    [--node <uuid>]
    [--type <volume type>]
    [--property <key=value>]
    [--boot-index <boot index>]
    [--volume-id <volume id>]
    [--extra <key=value>]
    <volume
    target>
```

--node `<uuid>`

UUID of the node that this volume target belongs to.

--type `<volume type>`

Type of the volume target, e.g. iscsi, fibre_channel.

--property `<key=value>`

Key/value property related to the type of this volume target. Can be specified multiple times.

--boot-index <boot index>

Boot index of the volume target.

--volume-id <volume id>

ID of the volume associated with this target.

--extra <key=value>

Record arbitrary key/value metadata. Can be specified multiple times.

volume target

UUID of the volume target.

This command is provided by the python-ironicclient plugin.

baremetal volume target show

Show baremetal volume target details.

```
openstack baremetal volume target show
    [-f {json,shell,table,value,yaml}]
    [-c COLUMN]
    [--noindent]
    [--prefix PREFIX]
    [--max-width <integer>]
    [--fit-width]
    [--print-empty]
    [--fields <field> [<field> ...]]
    <id>
```

-f <FORMATTER>, **--format** <FORMATTER>

the output format, defaults to table

-c COLUMN, **--column** COLUMN

specify the column(s) to include, can be repeated to show multiple columns

--noindent

whether to disable indenting the JSON

--prefix <PREFIX>

add a prefix to all variable names

--max-width <integer>

Maximum display width, <1 to disable. You can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.

--fit-width

Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable

--print-empty

Print empty table if there is no data to show.

--fields <field>

One or more volume target fields. Only these fields will be fetched from the server.

id

UUID of the volume target.

This command is provided by the python-ironicclient plugin.

baremetal volume target unset

Unset baremetal volume target properties.

```
openstack baremetal volume target unset
    [--extra <key>]
    [--property <key>]
    <volume
    target>
```

--extra <key>

Extra to unset (repeat option to unset multiple extras)

--property <key>

Property to unset on this baremetal volume target (repeat option to unset multiple properties).

volume target

UUID of the volume target.

This command is provided by the python-ironicclient plugin.

1.3 Creating the Bare Metal service resources from file

It is possible to create a set of resources using their descriptions in JSON or YAML format. It can be done in one of two ways:

1. Using OpenStackClient bare metal plugin CLIs command `openstack baremetal create`:

```
$ openstack -h baremetal create
usage: openstack baremetal create [-h] <file> [<file> ...]

Create resources from files

positional arguments:
  <file>      File (.yaml or .json) containing descriptions of the
              resources to create. Can be specified multiple times.
```

2. Programmatically using the Python API:

`ironicclient.v1.create_resources.`**`create_resources`**(*client*, *filenames*)

Create resources using their JSON or YAML descriptions.

Parameters

- **client** an instance of ironic client;

- **filenames** a list of filenames containing JSON or YAML resources definitions.

Raises ClientException if any operation during files processing/resource creation fails.

1.3.1 File containing Resource Descriptions

The resources to be created can be described either in JSON or YAML. A file ending with `.json` is assumed to contain valid JSON, and a file ending with `.yaml` is assumed to contain valid YAML. Specifying a file with any other extension leads to an error.

The resources that can be created are chassis, nodes, port groups and ports. A chassis can contain nodes (and resources of nodes) definitions nested under `"nodes"` key. A node can contain port groups definitions nested under `"portgroups"`, and ports definitions under `"ports"` keys. Ports can be also nested under port groups in `"ports"` key.

The schema used to validate the supplied data is the following:

```
{
    "$schema": "http://json-schema.org/draft-04/schema#",
    "description": "Schema for ironic resources file",
    "type": "object",
    "properties": {
        "chassis": {
            "type": "array",
            "items": {
                "type": "object"
            }
        },
        "nodes": {
            "type": "array",
            "items": {
                "type": "object"
            }
        }
    },
    "additionalProperties": False
}
```

More detailed description of the creation process can be seen in the following sections.

1.3.2 Examples

Here is an example of the JSON file that can be passed to the `create` command:

```
{
    "chassis": [
        {
            "description": "chassis 3 in row 23",
            "nodes": [
```

(continues on next page)

(continued from previous page)

```
                {
                    "name": "node-3",
                    "driver": "ipmi",
                    "portgroups": [
                        {
                            "name": "switch.cz7882.ports.1-2",
                            "ports": [
                                {
                                    "address": "ff:00:00:00:00:00"
                                },
                                {
                                    "address": "ff:00:00:00:00:01"
                                }
                            ]
                        }
                    ],
                    "ports": [
                        {
                            "address": "00:00:00:00:00:02"
                        },
                        {
                            "address": "00:00:00:00:00:03"
                        }
                    ],
                    "driver_info": {
                        "ipmi_address": "192.168.1.23",
                        "ipmi_username": "BmcUsername",
                        "ipmi_password": "BmcPassword",
                    }
                },
                {
                    "name": "node-4",
                    "driver": "ipmi",
                    "ports": [
                        {
                            "address": "00:00:00:00:00:04"
                        },
                        {
                            "address": "00:00:00:00:00:01"
                        }
                    ]
                }
            ]
        }
    ],
    "nodes": [
        {
            "name": "node-5",
            "driver": "ipmi",
```

(continues on next page)

(continued from previous page)

```
            "chassis_uuid": "74d93e6e-7384-4994-a614-fd7b399b0785",
            "ports": [
                {
                    "address": "00:00:00:00:00:00"
                }
            ]
        },
        {
            "name": "node-6",
            "driver": "ipmi"
        }
    ]
}
```

1.3.3 Creation Process

1. The client deserializes the files contents and validates that the top-level dictionary in each of them contains only chassis and/or nodes keys, and their values are lists. The creation process is aborted if any failure is encountered in this stage. The rest of the validation is done by the ironic-api service.
2. Each resource is created via issuing a POST request (with the resources dictionary representation in the body) to the ironic-api service. In the case of nested resources (`"nodes"` key inside chassis, `"portgroups"` key inside nodes, `"ports"` key inside nodes or portgroups), the top-level resource is created first, followed by the sub-resources. For example, if a chassis contains a list of nodes, the chassis will be created first followed by the creation of each node. The same is true for ports and port groups described within nodes.
3. If a resource could not be created, it does not stop the entire process. Any sub-resources of the failed resource will not be created, but otherwise, the rest of the resources will be created if possible. Any failed resources will be mentioned in the response.

1.4 python-ironicclient Contributor Documentation

1.4.1 Contributing to python-ironicclient

If youre interested in contributing to the python-ironicclient project, the following will help get you started.

#openstack-ironic on OFTC IRC Network

There is a very active chat channel at irc://irc.oftc.net/#openstack-ironic. This is usually the best place to ask questions and find your way around. IRC stands for Internet Relay Chat and it is a way to chat online in real time. You can ask a question and come back later to read the answer in the log files. Logs for the #openstack-ironic IRC channel are stored at http://eavesdrop.openstack.org/irclogs/%23openstack-ironic/.

Contributor License Agreement

In order to contribute to the python-ironicclient project, you need to have signed OpenStacks contributors agreement.

See also:

- https://docs.openstack.org/infra/manual/developers.html
- https://wiki.openstack.org/wiki/CLA

Project Hosting Details

Bug tracker https://storyboard.openstack.org/#!/project/959

Mailing list (prefix subjects with `[ironic]` for faster responses) http://lists.openstack.org/cgi-bin/mailman/listinfo/openstack-discuss

Code Hosting https://opendev.org/openstack/python-ironicclient

Code Review https://review.opendev.org/#/q/status:open+project:openstack/python-ironicclient,n,z

1.4.2 Testing

Python Guideline Enforcement

All code has to pass the pep8 style guideline to merge into OpenStack, to validate the code against these guidelines you can run:

```
$ tox -e pep8
```

Unit Testing

It is strongly encouraged to run the unit tests locally under one or more test environments prior to submitting a patch. To run all the recommended environments sequentially and pep8 style guideline run:

```
$ tox
```

You can also selectively pick specific test environments by listing your chosen environments after a -e flag:

```
$ tox -e py3,pep8
```

Note: Tox sets up virtual environment and installs all necessary dependencies. Sharing the environment with devstack testing is not recommended due to conflicting configuration with system dependencies.

Functional Testing

Functional testing assumes the existence of the script run_functional.sh in the python-ironicclient/tools directory. The script run_functional.sh generates test.conf file. To run functional tests just run ./run_functional.sh.

Also, the test.conf file could be created manually or generated from environment variables. It assumes the existence of an openstack cloud installation along with admin credentials. The test.conf file lives in ironicclient/tests/functional/ directory. To run functional tests in that way create test.conf manually and run:

```
$ tox -e functional
```

An example test.conf file:

```
[functional]
api_version = 1
os_auth_url=http://192.168.0.2:5000/v2.0/
os_username=admin
os_password=admin
os_project_name=admin
```

If you are testing ironic in standalone mode, only the parameters auth_strategy, os_auth_token and ironic_url are required; all others will be ignored.

An example test.conf file for standalone host:

```
[functional]
auth_strategy = noauth
os_auth_token = fake
ironic_url = http://10.0.0.2:6385
```

1.5 Full Ironic Client Python API Reference

1.5.1 ironicclient

ironicclient package

Subpackages

ironicclient.common package

Subpackages

ironicclient.common.apiclient package

Submodules

ironicclient.common.apiclient.base module

Base utilities to build API operation managers and objects on top of.

class ironicclient.common.apiclient.base.**BaseManager**(*client*)

Bases: *ironicclient.common.apiclient.base.HookableMixin*

Basic manager type providing common operations.

Managers interact with a particular type of API (servers, flavors, images, etc.) and provide CRUD operations for them.

resource_class = None

class ironicclient.common.apiclient.base.**CrudManager**(*client*)

Bases: *ironicclient.common.apiclient.base.BaseManager*

Base manager class for manipulating entities.

Children of this class are expected to define a *collection_key* and *key*.

- *collection_key*: Usually a plural noun by convention (e.g. *entities*); used to refer collections in both URLs (e.g. */v3/entities*) and JSON objects containing a list of member resources (e.g. *{entities: [{}, {}, {}]}*).
- *key*: Usually a singular noun by convention (e.g. *entity*); used to refer to an individual member of the collection.

build_url(*base_url=None*, ***kwargs*)

Builds a resource URL for the given kwargs.

Given an example collection where *collection_key = entities* and *key = entity*, the following URLs could be generated.

By default, the URL will represent a collection of entities, e.g.:

```
/entities
```

If kwargs contains an *entity_id*, then the URL will represent a specific member, e.g.:

```
/entities/{entity_id}
```

Parameters **base_url** if provided, the generated URL will be appended to it

collection_key = None

create(***kwargs*)

delete(***kwargs*)

find(*base_url=None*, ***kwargs*)

Find a single item with attributes matching `**kwargs`.

Parameters **base_url** if provided, the generated URL will be appended to it

get(***kwargs*)

head(***kwargs*)

key = None

list(*base_url=None*, ***kwargs*)

List the collection.

Parameters **base_url** if provided, the generated URL will be appended to it

put(*base_url=None*, ***kwargs*)

Update an element.

Parameters **base_url** if provided, the generated URL will be appended to it

update(***kwargs*)

class ironicclient.common.apiclient.base.**Extension**(*name*, *module*)

Bases: *ironicclient.common.apiclient.base.HookableMixin*

Extension descriptor.

SUPPORTED_HOOKS = ('__pre_parse_args__', '__post_parse_args__')

manager_class = None

class ironicclient.common.apiclient.base.**HookableMixin**

Bases: object

Mixin so classes can register and run hooks.

classmethod **add_hook**(*hook_type*, *hook_func*)

Add a new hook of specified type.

Parameters

- **cls** class that registers hooks
- **hook_type** hook type, e.g., __pre_parse_args__
- **hook_func** hook function

classmethod **run_hooks**(*hook_type*, **args*, ***kwargs*)

Run all hooks of specified type.

Parameters

- **cls** class that registers hooks
- **hook_type** hook type, e.g., __pre_parse_args__
- **args** args to be passed to every hook function
- **kwargs** kwargs to be passed to every hook function

class ironicclient.common.apiclient.base.**ManagerWithFind**(*client*)

Bases: *ironicclient.common.apiclient.base.BaseManager*

Manager with additional *find()*/*findall()* methods.

find(***kwargs*)

Find a single item with attributes matching `**kwargs`.

This isnt very efficient: it loads the entire list then filters on the Python side.

findall(***kwargs*)

Find all items with attributes matching `**kwargs`.

This isnt very efficient: it loads the entire list then filters on the Python side.

abstract list()

class `ironicclient.common.apiclient.base.`**Resource**(*manager*, *info*, *loaded=False*)

Bases: `object`

Base class for OpenStack resources (tenant, user, etc.).

This is pretty much just a bag for attributes.

HUMAN_ID = False

NAME_ATTR = 'name'

get()

Support for lazy loading details.

Some clients, such as novaclient have the option to lazy load the details, details which can be loaded with this function.

property human_id

Human-readable ID which can be used for bash completion.

is_loaded()

set_loaded(*val*)

to_dict()

`ironicclient.common.apiclient.base.`**getid**(*obj*)

Return id if argument is a Resource.

Abstracts the common pattern of allowing both an object or an objects ID (UUID) as a parameter when dealing with relationships.

ironicclient.common.apiclient.exceptions module

Exception definitions.

exception `ironicclient.common.apiclient.exceptions.`**AmbiguousEndpoints**(*endpoints=None*)

Bases: *`ironicclient.common.apiclient.exceptions.EndpointException`*

Found more than one matching endpoint in Service Catalog.

exception `ironicclient.common.apiclient.exceptions.`**AuthPluginOptionsMissing**(*opt_names*)

Bases: *`ironicclient.common.apiclient.exceptions.AuthorizationFailure`*

Auth plugin misses some options.

exception ironicclient.common.apiclient.exceptions.**AuthSystemNotFound**(*auth_system*)

Bases: *ironicclient.common.apiclient.exceptions.AuthorizationFailure*

User has specified an AuthSystem that is not installed.

exception ironicclient.common.apiclient.exceptions.**AuthorizationFailure**

Bases: *ironicclient.common.apiclient.exceptions.ClientException*

Cannot authorize API client.

exception ironicclient.common.apiclient.exceptions.**BadGateway**(*message=None*, *details=None*, *response=None*, *request_id=None*, *url=None*, *method=None*, *http_status=None*)

Bases: *ironicclient.common.apiclient.exceptions.HttpServerError*

HTTP 502 - Bad Gateway.

The server was acting as a gateway or proxy and received an invalid response from the upstream server.

http_status = 502

message = 'Bad Gateway'

exception ironicclient.common.apiclient.exceptions.**BadRequest**(*message=None*, *details=None*, *response=None*, *request_id=None*, *url=None*, *method=None*, *http_status=None*)

Bases: *ironicclient.common.apiclient.exceptions.HTTPClientError*

HTTP 400 - Bad Request.

The request cannot be fulfilled due to bad syntax.

http_status = 400

message = 'Bad Request'

exception ironicclient.common.apiclient.exceptions.**ClientException**

Bases: Exception

The base exception class for all exceptions this library raises.

exception ironicclient.common.apiclient.exceptions.**CommandError**

Bases: *ironicclient.common.apiclient.exceptions.ClientException*

Error in CLI tool.

exception ironicclient.common.apiclient.exceptions.**Conflict**(*message=None*, *details=None*, *response=None*, *request_id=None*, *url=None*, *method=None*, *http_status=None*)

Bases: *ironicclient.common.apiclient.exceptions.HTTPClientError*

HTTP 409 - Conflict.

Indicates that the request could not be processed because of conflict in the request, such as an edit conflict.

http_status = 409

message = 'Conflict'

exception ironicclient.common.apiclient.exceptions.**ConnectionError**

Bases: *ironicclient.common.apiclient.exceptions.ClientException*

Cannot connect to API service.

exception ironicclient.common.apiclient.exceptions.**ConnectionRefused**

Bases: *ironicclient.common.apiclient.exceptions.ConnectionError*

Connection refused while trying to connect to API service.

exception ironicclient.common.apiclient.exceptions.**EndpointException**

Bases: *ironicclient.common.apiclient.exceptions.ClientException*

Something is rotten in Service Catalog.

exception ironicclient.common.apiclient.exceptions.**EndpointNotFound**

Bases: *ironicclient.common.apiclient.exceptions.EndpointException*

Could not find requested endpoint in Service Catalog.

exception ironicclient.common.apiclient.exceptions.**ExpectationFailed**(*message=None*, *details=None*, *response=None*, *request_id=None*, *url=None*, *method=None*, *http_status=None*)

Bases: *ironicclient.common.apiclient.exceptions.HTTPClientError*

HTTP 417 - Expectation Failed.

The server cannot meet the requirements of the Expect request-header field.

http_status = 417

message = 'Expectation Failed'

exception ironicclient.common.apiclient.exceptions.**Forbidden**(*message=None*, *details=None*, *response=None*, *request_id=None*, *url=None*, *method=None*, *http_status=None*)

Bases: *ironicclient.common.apiclient.exceptions.HTTPClientError*

HTTP 403 - Forbidden.

The request was a valid request, but the server is refusing to respond to it.

http_status = 403

message = 'Forbidden'

exception ironicclient.common.apiclient.exceptions.**GatewayTimeout**(*message=None*, *details=None*, *response=None*, *request_id=None*, *url=None*, *method=None*, *http_status=None*)

Bases: *ironicclient.common.apiclient.exceptions.HttpServerError*

HTTP 504 - Gateway Timeout.

The server was acting as a gateway or proxy and did not receive a timely response from the upstream server.

http_status = 504

message = 'Gateway Timeout'

exception ironicclient.common.apiclient.exceptions.**Gone**(*message=None*, *details=None*, *response=None*, *request_id=None*, *url=None*, *method=None*, *http_status=None*)

Bases: *ironicclient.common.apiclient.exceptions.HTTPClientError*

HTTP 410 - Gone.

Indicates that the resource requested is no longer available and will not be available again.

http_status = 410

message = 'Gone'

exception ironicclient.common.apiclient.exceptions.**HTTPClientError**(*message=None*, *details=None*, *response=None*, *request_id=None*, *url=None*, *method=None*, *http_status=None*)

Bases: *ironicclient.common.apiclient.exceptions.HttpError*

Client-side HTTP error.

Exception for cases in which the client seems to have erred.

message = 'HTTP Client Error'

exception ironicclient.common.apiclient.exceptions.**HTTPRedirection**(*message=None*, *details=None*, *response=None*, *request_id=None*, *url=None*, *method=None*, *http_status=None*)

Bases: *ironicclient.common.apiclient.exceptions.HttpError*

HTTP Redirection.

message = 'HTTP Redirection'

exception ironicclient.common.apiclient.exceptions.**HttpError**(*message=None*, *details=None*, *response=None*, *request_id=None*, *url=None*, *method=None*, *http_status=None*)

Bases: *ironicclient.common.apiclient.exceptions.ClientException*

The base exception class for all HTTP exceptions.

http_status = 0

message = 'HTTP Error'

exception ironicclient.common.apiclient.exceptions.**HttpNotImplemented**(*message=None, details=None, response=None, request_id=None, url=None, method=None, http_status=None*)

Bases: *ironicclient.common.apiclient.exceptions.HttpServerError*

HTTP 501 - Not Implemented.

The server either does not recognize the request method, or it lacks the ability to fulfill the request.

http_status = 501

message = 'Not Implemented'

exception ironicclient.common.apiclient.exceptions.**HttpServerError**(*message=None, details=None, response=None, request_id=None, url=None, method=None, http_status=None*)

Bases: *ironicclient.common.apiclient.exceptions.HttpError*

Server-side HTTP error.

Exception for cases in which the server is aware that it has erred or is incapable of performing the request.

message = 'HTTP Server Error'

exception ironicclient.common.apiclient.exceptions.**HttpVersionNotSupported**(*message=None, details=None, response=None, request_id=None, url=None, method=None, http_status=None*)

Bases: *ironicclient.common.apiclient.exceptions.HttpServerError*

HTTP 505 - HttpVersion Not Supported.

The server does not support the HTTP protocol version used in the request.

http_status = 505

message = 'HTTP Version Not Supported'

exception ironicclient.common.apiclient.exceptions.**InternalServerError**(*message=None, details=None, response=None, request_id=None, url=None, method=None, http_status=None*)

Bases: *ironicclient.common.apiclient.exceptions.HttpServerError*

HTTP 500 - Internal Server Error.

A generic error message, given when no more specific message is suitable.

http_status = 500

message = 'Internal Server Error'

exception ironicclient.common.apiclient.exceptions.**LengthRequired**(*message=None, details=None, response=None, request_id=None, url=None, method=None, http_status=None*)

Bases: *ironicclient.common.apiclient.exceptions.HTTPClientError*

HTTP 411 - Length Required.

The request did not specify the length of its content, which is required by the requested resource.

http_status = 411

message = 'Length Required'

exception ironicclient.common.apiclient.exceptions.**MethodNotAllowed**(*message=None, details=None, response=None, request_id=None, url=None, method=None, http_status=None*)

Bases: *ironicclient.common.apiclient.exceptions.HTTPClientError*

HTTP 405 - Method Not Allowed.

A request was made of a resource using a request method not supported by that resource.

http_status = 405

message = 'Method Not Allowed'

exception ironicclient.common.apiclient.exceptions.**MultipleChoices**(*message=None*, *details=None*, *response=None*, *request_id=None*, *url=None*, *method=None*, *http_status=None*)

Bases: *ironicclient.common.apiclient.exceptions.HTTPRedirection*

HTTP 300 - Multiple Choices.

Indicates multiple options for the resource that the client may follow.

http_status = 300

message = 'Multiple Choices'

exception ironicclient.common.apiclient.exceptions.**NoUniqueMatch**

Bases: *ironicclient.common.apiclient.exceptions.ClientException*

Multiple entities found instead of one.

exception ironicclient.common.apiclient.exceptions.**NotAcceptable**(*message=None*, *details=None*, *response=None*, *request_id=None*, *url=None*, *method=None*, *http_status=None*)

Bases: *ironicclient.common.apiclient.exceptions.HTTPClientError*

HTTP 406 - Not Acceptable.

The requested resource is only capable of generating content not acceptable according to the Accept headers sent in the request.

http_status = 406

message = 'Not Acceptable'

exception ironicclient.common.apiclient.exceptions.**NotFound**(*message=None*, *details=None*, *response=None*, *request_id=None*, *url=None*, *method=None*, *http_status=None*)

Bases: *ironicclient.common.apiclient.exceptions.HTTPClientError*

HTTP 404 - Not Found.

The requested resource could not be found but may be available again in the future.

http_status = 404

message = 'Not Found'

exception ironicclient.common.apiclient.exceptions.**PaymentRequired**(*message=None, details=None, response=None, request_id=None, url=None, method=None, http_status=None*)

Bases: *ironicclient.common.apiclient.exceptions.HTTPClientError*

HTTP 402 - Payment Required.

Reserved for future use.

http_status = 402

message = 'Payment Required'

exception ironicclient.common.apiclient.exceptions.**PreconditionFailed**(*message=None, details=None, response=None, request_id=None, url=None, method=None, http_status=None*)

Bases: *ironicclient.common.apiclient.exceptions.HTTPClientError*

HTTP 412 - Precondition Failed.

The server does not meet one of the preconditions that the requester put on the request.

http_status = 412

message = 'Precondition Failed'

exception ironicclient.common.apiclient.exceptions.**ProxyAuthenticationRequired**(*message=None, details=None, response=None, request_id=None, url=None, method=None, http_status=Nor*

Bases: *ironicclient.common.apiclient.exceptions.HTTPClientError*

HTTP 407 - Proxy Authentication Required.

The client must first authenticate itself with the proxy.

http_status = 407

message = 'Proxy Authentication Required'

exception ironicclient.common.apiclient.exceptions.**RequestEntityTooLarge**(**args*, ***kwargs*)

Bases: *ironicclient.common.apiclient.exceptions.HTTPClientError*

HTTP 413 - Request Entity Too Large.

The request is larger than the server is willing or able to process.

http_status = 413

message = 'Request Entity Too Large'

exception ironicclient.common.apiclient.exceptions.**RequestTimeout**(*message=None*, *details=None*, *response=None*, *request_id=None*, *url=None*, *method=None*, *http_status=None*)

Bases: *ironicclient.common.apiclient.exceptions.HTTPClientError*

HTTP 408 - Request Timeout.

The server timed out waiting for the request.

http_status = 408

message = 'Request Timeout'

exception ironicclient.common.apiclient.exceptions.**RequestUriTooLong**(*message=None*, *details=None*, *response=None*, *request_id=None*, *url=None*, *method=None*, *http_status=None*)

Bases: *ironicclient.common.apiclient.exceptions.HTTPClientError*

HTTP 414 - Request-URI Too Long.

The URI provided was too long for the server to process.

http_status = 414

message = 'Request-URI Too Long'

exception ironicclient.common.apiclient.exceptions.**RequestedRangeNotSatisfiable**(*message=None details=None, response=None, request_id=None url=None, method=None, http_status=No*

Bases: *ironicclient.common.apiclient.exceptions.HTTPClientError*

HTTP 416 - Requested Range Not Satisfiable.

The client has asked for a portion of the file, but the server cannot supply that portion.

http_status = 416

message = 'Requested Range Not Satisfiable'

exception ironicclient.common.apiclient.exceptions.**ServiceUnavailable**(*message=None, details=None, response=None, request_id=None, url=None, method=None, http_status=None*)

Bases: *ironicclient.common.apiclient.exceptions.HttpServerError*

HTTP 503 - Service Unavailable.

The server is currently unavailable.

http_status = 503

message = 'Service Unavailable'

exception ironicclient.common.apiclient.exceptions.**Unauthorized**(*message=None, details=None, response=None, request_id=None, url=None, method=None, http_status=None*)

Bases: *ironicclient.common.apiclient.exceptions.HTTPClientError*

HTTP 401 - Unauthorized.

Similar to 403 Forbidden, but specifically for use when authentication is required and has failed or has not yet been provided.

http_status = 401

message = 'Unauthorized'

exception ironicclient.common.apiclient.exceptions.**UnprocessableEntity**(*message=None*, *details=None*, *response=None*, *request_id=None*, *url=None*, *method=None*, *http_status=None*)

Bases: *ironicclient.common.apiclient.exceptions.HTTPClientError*

HTTP 422 - Unprocessable Entity.

The request was well-formed but was unable to be followed due to semantic errors.

http_status = 422

message = 'Unprocessable Entity'

exception ironicclient.common.apiclient.exceptions.**UnsupportedMediaType**(*message=None*, *details=None*, *response=None*, *request_id=None*, *url=None*, *method=None*, *http_status=None*)

Bases: *ironicclient.common.apiclient.exceptions.HTTPClientError*

HTTP 415 - Unsupported Media Type.

The request entity has a media type which the server or resource does not support.

http_status = 415

message = 'Unsupported Media Type'

exception ironicclient.common.apiclient.exceptions.**UnsupportedVersion**

Bases: *ironicclient.common.apiclient.exceptions.ClientException*

User is trying to use an unsupported version of the API.

exception ironicclient.common.apiclient.exceptions.**ValidationError**

Bases: *ironicclient.common.apiclient.exceptions.ClientException*

Error in validation on API client side.

ironicclient.common.apiclient.exceptions.**from_response**(*response*, *method*, *url*)

Returns an instance of *HttpError* or subclass based on response.

Parameters

- **response** instance of *requests.Response* class
- **method** HTTP method used for request
- **url** URL used for request

Module contents

Submodules

ironicclient.common.base module

Base utilities to build API operation managers and objects on top of.

`class ironicclient.common.base.CreateManager(api)`

Bases: *ironicclient.common.base.Manager*

Provides creation operations with a particular API.

`create(os_ironic_api_version=None, global_request_id=None, **kwargs)`

Create a resource based on a kwargs dictionary of attributes.

Parameters

- **kwargs** A dictionary containing the attributes of the resource that will be created.
- **os_ironic_api_version** String version (e.g. 1.35) to use for the request. If not specified, the clients default is used.
- **global_request_id** String containing global request ID header value (in form req-<UUID>) to use for the request.

Raises *exc.InvalidAttribute* For invalid attributes that are not needed to create the resource.

`class ironicclient.common.base.Manager(api)`

Bases: `object`

Provides CRUD operations with a particular API.

`abstract property resource_class`

The resource class

`class ironicclient.common.base.Resource(manager, info, loaded=False)`

Bases: *ironicclient.common.apiclient.base.Resource*

Represents a particular instance of an object (tenant, user, etc).

This is pretty much just a bag for attributes.

`to_dict()`

`ironicclient.common.base.`**`getid`**`(`*obj*`)`

Wrapper to get objects ID.

Abstracts the common pattern of allowing both an object or an objects ID (UUID) as a parameter when dealing with relationships.

ironicclient.common.filecache module

`ironicclient.common.filecache.`**`retrieve_data`**`(`*host*, *port*, *expiry=None*`)`

Retrieve the version stored for an ironic host, if its not stale.

Check to see if there is valid cached data for the host/port combination and return that if it isnt stale.

param host: The host that we need to retrieve data for param port: The port on the host that we need to retrieve data for param expiry: The age in seconds before cached data is deemed invalid

`ironicclient.common.filecache.`**`save_data`**`(`*host*, *port*, *data*`)`

Save data for a particular host in the appropriate cache dir.

param host: The host that we need to save data for param port: The port on the host that we need to save data for param data: The data we want saved

ironicclient.common.http module

`class` `ironicclient.common.http.`**`SessionClient`**`(`*os_ironic_api_version*, *api_version_select_state*, *max_retries*, *retry_interval*, ***kwargs*`)`

Bases: *`ironicclient.common.http.VersionNegotiationMixin`*, `keystoneauth1.adapter.LegacyJsonAdapter`

HTTP client based on Keystone client session.

`json_request``(`*method*, *url*, ***kwargs*`)`

`raw_request``(`*method*, *url*, ***kwargs*`)`

`class` `ironicclient.common.http.`**`VersionNegotiationMixin`**

Bases: `object`

`negotiate_version``(`*conn*, *resp*`)`

Negotiate the server version

Assumption: Called after receiving a 406 error when doing a request.

Parameters

- **`conn`** A connection object
- **`resp`** The response object from http request

`ironicclient.common.http.`**`get_server`**`(`*url*`)`

Extract and return the server & port.

`ironicclient.common.http.`**`with_retries`**`(`*func*`)`

Wrapper for _http_request adding support for retries.

ironicclient.common.i18n module

ironicclient.common.utils module

class ironicclient.common.utils.**HelpFormatter**(*prog*, *indent_increment=2*, *max_help_position=24*, *width=None*)

Bases: argparse.HelpFormatter

start_section(*heading*)

ironicclient.common.utils.**args_array_to_dict**(*kwargs*, *key_to_convert*)

Convert the value in a dictionary entry to a dictionary.

From the kwargs dictionary, converts the value of the key_to_convert entry from a list of key-value pairs to a dictionary.

Parameters

- **kwargs** a dictionary
- **key_to_convert** the key (in kwargs), whose value is expected to be a list of key=value strings. This value will be converted to a dictionary.

Returns kwargs, the (modified) dictionary

ironicclient.common.utils.**args_array_to_patch**(*op*, *attributes*)

ironicclient.common.utils.**bool_argument_value**(*arg_name*, *bool_str*, *strict=True*, *default=False*)

Returns the Boolean represented by bool_str.

Returns the Boolean value for the argument named arg_name. The value is represented by the string bool_str. If the string is an invalid Boolean string: if strict is True, a CommandError exception is raised; otherwise the default value is returned.

Parameters

- **arg_name** The name of the argument
- **bool_str** The string representing a Boolean value
- **strict** Used if the string is invalid. If True, raises an exception. If False, returns the default value.
- **default** The default value to return if the string is invalid and not strict

Returns the Boolean value represented by bool_str or the default value if bool_str is invalid and strict is False

Raises *CommandError* if bool_str is an invalid Boolean string

ironicclient.common.utils.**check_empty_arg**(*arg*, *arg_descriptor*)

ironicclient.common.utils.**check_for_invalid_fields**(*fields*, *valid_fields*)

Check for invalid fields.

Parameters

- **fields** A list of fields specified by the user.

- **valid_fields** A list of valid fields.

Raises *CommandError* If invalid fields were specified by the user.

ironicclient.common.utils.**common_filters**(*marker=None*, *limit=None*, *sort_key=None*, *sort_dir=None*, *fields=None*, *detail=False*)

Generate common filters for any list request.

Parameters

- **marker** entity ID from which to start returning entities.
- **limit** maximum number of entities to return.
- **sort_key** field to use for sorting.
- **sort_dir** direction of sorting: asc or desc.
- **fields** a list with a specified set of fields of the resource to be returned.
- **detail** Boolean, True to return detailed information. This parameter can be used for resources which accept detail as a URL parameter.

Returns list of string filters.

ironicclient.common.utils.**common_params_for_list**(*args*, *fields*, *field_labels*)

Generate params dict that is common for every list command.

Parameters

- **args** arguments from command line.
- **fields** possible fields for sorting.
- **field_labels** possible field labels for sorting.

Returns a dict with params to pass to the client method.

ironicclient.common.utils.**convert_list_props_to_comma_separated**(*data*, *props=None*)

Convert the list-type properties to comma-separated strings

Parameters

- **data** the input dict object.
- **props** the properties whose values will be converted. Default to None to convert all list-type properties of the input.

Returns the result dict instance.

ironicclient.common.utils.**define_command**(*subparsers*, *command*, *callback*, *cmd_mapper*)

Define a command in the subparsers collection.

Parameters

- **subparsers** subparsers collection where the command will go
- **command** command name
- **callback** function that will be used to process the command

ironicclient.common.utils.**define_commands_from_module**(*subparsers*, *command_module*, *cmd_mapper*)

Add *do_* methods in a module and add as commands into a subparsers.

ironicclient.common.utils.**get_from_stdin**(*info_desc*)

Read information from stdin.

Parameters **info_desc** A string description of the desired information

Raises InvalidAttribute if there was a problem reading from stdin

Returns the string that was read from stdin

ironicclient.common.utils.**handle_json_arg**(*json_arg*, *info_desc*)

Read a JSON argument from stdin, file or string.

Parameters

- **json_arg** May be a file name containing the JSON, a JSON string, or - indicating that the argument should be read from standard input.
- **info_desc** A string description of the desired information

Returns A list or dictionary parsed from JSON.

Raises InvalidAttribute if the argument cannot be parsed.

ironicclient.common.utils.**handle_json_or_file_arg**(*json_arg*)

Attempts to read JSON argument from file or string.

Parameters **json_arg** May be a file name containing the YAML or JSON, or a JSON string.

Returns A list or dictionary parsed from JSON.

Raises InvalidAttribute if the argument cannot be parsed.

ironicclient.common.utils.**key_value_pairs_to_dict**(*key_value_pairs*)

Convert a list of key-value pairs to a dictionary.

Parameters **key_value_pairs** a list of strings, each string is in the form <key>=<value>

Returns a dictionary, possibly empty

ironicclient.common.utils.**make_configdrive**(*path*)

Make the config drive file.

Parameters **path** The directory containing the config drive files.

Returns A gzipped and base64 encoded configdrive string.

ironicclient.common.utils.**poll**(*timeout*, *poll_interval*, *poll_delay_function*, *timeout_message*)

ironicclient.common.utils.**split_and_deserialize**(*string*)

Split and try to JSON deserialize a string.

Gets a string with the KEY=VALUE format, split it (using = as the separator) and try to JSON deserialize the VALUE.

Returns A tuple of (key, value).

ironicclient.common.utils.**tempdir**(**args*, ***kwargs*)

Module contents

ironicclient.osc package

Subpackages

ironicclient.osc.v1 package

Submodules

ironicclient.osc.v1.baremetal_allocation module

class ironicclient.osc.v1.baremetal_allocation.**CreateBaremetalAllocation**(*app: cliff.app.App, app_args: Optional[argparse.Namesp cmd_name: Optional[str] = None*)

Bases: `osc_lib.command.command.ShowOne`

Create a new baremetal allocation.

get_parser(*prog_name*)

Return an `argparse.ArgumentParser`.

log = <Logger ironicclient.osc.v1.baremetal_allocation.CreateBaremetalAllocation (WARNING)>

take_action(*parsed_args*)

Return a two-part tuple with a tuple of column names and a tuple of values.

class ironicclient.osc.v1.baremetal_allocation.**DeleteBaremetalAllocation**(*app: _app.App, app_args: Optional[argparse.Namesp cmd_name: Optional[str] = None*)

Bases: `osc_lib.command.command.Command`

Unregister baremetal allocation(s).

get_parser(*prog_name*)

Return an `argparse.ArgumentParser`.

log = <Logger ironicclient.osc.v1.baremetal_allocation.DeleteBaremetalAllocation (WARNING)>

take_action(*parsed_args*)

Override to do something useful.

The returned value will be returned by the program.

class `ironicclient.osc.v1.baremetal_allocation.`**ListBaremetalAllocation**(*app: cliff.app.App*, *app_args: Optional[argparse.Namespac*
cmd_name: Optional[str] = None)

Bases: `osc_lib.command.command.Lister`

List baremetal allocations.

get_parser(*prog_name*)

Return an `argparse.ArgumentParser`.

log = <Logger ironicclient.osc.v1.baremetal_allocation.ListBaremetalAllocation (WARNING)>

take_action(*parsed_args*)

Run command.

Return a tuple containing the column names and an iterable containing the data to be listed.

class `ironicclient.osc.v1.baremetal_allocation.`**SetBaremetalAllocation**(*app: _app.App*, *app_args: Optional[argparse.Namespace*
cmd_name: Optional[str] = None)

Bases: `osc_lib.command.command.Command`

Set baremetal allocation properties.

get_parser(*prog_name*)

Return an `argparse.ArgumentParser`.

log = <Logger ironicclient.osc.v1.baremetal_allocation.SetBaremetalAllocation (WARNING)>

take_action(*parsed_args*)

Override to do something useful.

The returned value will be returned by the program.

class ironicclient.osc.v1.baremetal_allocation.**ShowBaremetalAllocation**(*app: cliff.app.App, app_args: Optional[argparse.Namespac cmd_name: Optional[str] = None*)

Bases: osc_lib.command.command.ShowOne

Show baremetal allocation details.

get_parser(*prog_name*)

Return an argparse.ArgumentParser.

log = <Logger ironicclient.osc.v1.baremetal_allocation.ShowBaremetalAllocation (WARNING)>

take_action(*parsed_args*)

Return a two-part tuple with a tuple of column names and a tuple of values.

class ironicclient.osc.v1.baremetal_allocation.**UnsetBaremetalAllocation**(*app: _app.App, app_args: Optional[argparse.Namespa cmd_name: Optional[str] = None*)

Bases: osc_lib.command.command.Command

Unset baremetal allocation properties.

get_parser(*prog_name*)

Return an argparse.ArgumentParser.

log = <Logger ironicclient.osc.v1.baremetal_allocation.UnsetBaremetalAllocation (WARNING)>

take_action(*parsed_args*)

Override to do something useful.

The returned value will be returned by the program.

ironicclient.osc.v1.baremetal_chassis module

`class ironicclient.osc.v1.baremetal_chassis.`**`CreateBaremetalChassis`**(*app: cliff.app.App, app_args: Optional[argparse.Namespace], cmd_name: Optional[str] = None*)

Bases: `osc_lib.command.command.ShowOne`

Create a new chassis.

`get_parser`(*prog_name*)

Return an `argparse.ArgumentParser`.

`log = <Logger ironicclient.osc.v1.baremetal_chassis.CreateBaremetalChassis (WARNING)>`

`take_action`(*parsed_args*)

Return a two-part tuple with a tuple of column names and a tuple of values.

`class ironicclient.osc.v1.baremetal_chassis.`**`DeleteBaremetalChassis`**(*app: _app.App, app_args: Optional[argparse.Namespace], cmd_name: Optional[str] = None*)

Bases: `osc_lib.command.command.Command`

Delete a chassis.

`get_parser`(*prog_name*)

Return an `argparse.ArgumentParser`.

`log = <Logger ironicclient.osc.v1.baremetal_chassis.DeleteBaremetalChassis (WARNING)>`

`take_action`(*parsed_args*)

Override to do something useful.

The returned value will be returned by the program.

`class ironicclient.osc.v1.baremetal_chassis.`**`ListBaremetalChassis`**(*app: cliff.app.App, app_args: Optional[argparse.Namespace], cmd_name: Optional[str] = None*)

Bases: `osc_lib.command.command.Lister`

List the chassis.

get_parser(*prog_name*)

Return an argparse.ArgumentParser.

log = <Logger ironicclient.osc.v1.baremetal_chassis.ListBaremetalChassis (WARNING)>

take_action(*parsed_args*)

Run command.

Return a tuple containing the column names and an iterable containing the data to be listed.

class ironicclient.osc.v1.baremetal_chassis.**SetBaremetalChassis**(*app: _app.App, app_args: Optional[argparse.Namespace], cmd_name: Optional[str] = None*)

Bases: osc_lib.command.command.Command

Set chassis properties.

get_parser(*prog_name*)

Return an argparse.ArgumentParser.

log = <Logger ironicclient.osc.v1.baremetal_chassis.SetBaremetalChassis (WARNING)>

take_action(*parsed_args*)

Override to do something useful.

The returned value will be returned by the program.

class ironicclient.osc.v1.baremetal_chassis.**ShowBaremetalChassis**(*app: cliff.app.App, app_args: Optional[argparse.Namespace], cmd_name: Optional[str] = None*)

Bases: osc_lib.command.command.ShowOne

Show chassis details.

get_parser(*prog_name*)

Return an argparse.ArgumentParser.

log = <Logger ironicclient.osc.v1.baremetal_chassis.ShowBaremetalChassis (WARNING)>

take_action(*parsed_args*)

Return a two-part tuple with a tuple of column names and a tuple of values.

class `ironicclient.osc.v1.baremetal_chassis.`**`UnsetBaremetalChassis`**(*app: _app.App*, *app_args: Optional[argparse.Namespace]*, *cmd_name: Optional[str] = None*)

Bases: `osc_lib.command.command.Command`

Unset chassis properties.

`get_parser`(*prog_name*)

Return an `argparse.ArgumentParser`.

`log = <Logger ironicclient.osc.v1.baremetal_chassis.UnsetBaremetalChassis (WARNING)>`

`take_action`(*parsed_args*)

Override to do something useful.

The returned value will be returned by the program.

ironicclient.osc.v1.baremetal_conductor module

class `ironicclient.osc.v1.baremetal_conductor.`**`ListBaremetalConductor`**(*app: cliff.app.App*, *app_args: Optional[argparse.Namespace]*, *cmd_name: Optional[str] = None*)

Bases: `osc_lib.command.command.Lister`

List baremetal conductors

`get_parser`(*prog_name*)

Return an `argparse.ArgumentParser`.

`log = <Logger ironicclient.osc.v1.baremetal_conductor.ListBaremetalNode (WARNING)>`

`take_action`(*parsed_args*)

Run command.

Return a tuple containing the column names and an iterable containing the data to be listed.

class ironicclient.osc.v1.baremetal_conductor.**ShowBaremetalConductor**(*app: cliff.app.App*, *app_args: Optional[argparse.Namespace]*, *cmd_name: Optional[str] = None*)

Bases: `osc_lib.command.command.ShowOne`

Show baremetal conductor details

get_parser(*prog_name*)

Return an `argparse.ArgumentParser`.

log = <Logger ironicclient.osc.v1.baremetal_conductor.ShowBaremetalConductor (WARNING)>

take_action(*parsed_args*)

Return a two-part tuple with a tuple of column names and a tuple of values.

ironicclient.osc.v1.baremetal_create module

class ironicclient.osc.v1.baremetal_create.**CreateBaremetal**(*app: _app.App*, *app_args: Optional[argparse.Namespace]*, *cmd_name: Optional[str] = None*)

Bases: `osc_lib.command.command.Command`

Create resources from files

get_parser(*prog_name*)

Return an `argparse.ArgumentParser`.

log = <Logger ironicclient.osc.v1.baremetal_create.CreateBaremetal (WARNING)>

take_action(*parsed_args*)

Override to do something useful.

The returned value will be returned by the program.

ironicclient.osc.v1.baremetal_deploy_template module

class ironicclient.osc.v1.baremetal_deploy_template.**CreateBaremetalDeployTemplate**(*app: cliff.app.App, app_args: Optional[argpa cmd_name: Optional[str] = None*)

Bases: osc_lib.command.command.ShowOne

Create a new deploy template

get_parser(*prog_name*)

Return an argparse.ArgumentParser.

log = <Logger ironicclient.osc.v1.baremetal_deploy_template.CreateBaremetalDeployTemplate (WARNING)>

take_action(*parsed_args*)

Return a two-part tuple with a tuple of column names and a tuple of values.

class ironicclient.osc.v1.baremetal_deploy_template.**DeleteBaremetalDeployTemplate**(*app: _app.App, app_args: Optional[argpa cmd_name: Optional[str] = None*)

Bases: osc_lib.command.command.Command

Delete deploy template(s).

get_parser(*prog_name*)

Return an argparse.ArgumentParser.

log = <Logger ironicclient.osc.v1.baremetal_deploy_template.DeleteBaremetalDeployTemplate (WARNING)>

take_action(*parsed_args*)

Override to do something useful.

The returned value will be returned by the program.

class ironicclient.osc.v1.baremetal_deploy_template.**ListBaremetalDeployTemplate**(*app: cliff.app.App, app_args: Optional[argpars cmd_name: Optional[str] = None*)

Bases: osc_lib.command.command.Lister

List baremetal deploy templates.

get_parser(*prog_name*)

Return an argparse.ArgumentParser.

log = <Logger ironicclient.osc.v1.baremetal_deploy_template.ListBaremetalDeployTemplate (WARNING)>

take_action(*parsed_args*)

Run command.

Return a tuple containing the column names and an iterable containing the data to be listed.

class ironicclient.osc.v1.baremetal_deploy_template.**SetBaremetalDeployTemplate**(*app: _app.App, app_args: Optional[argparse. cmd_name: Optional[str] = None*)

Bases: osc_lib.command.command.Command

Set baremetal deploy template properties.

get_parser(*prog_name*)

Return an argparse.ArgumentParser.

log = <Logger ironicclient.osc.v1.baremetal_deploy_template.SetBaremetalDeployTemplate (WARNING)>

take_action(*parsed_args*)

Override to do something useful.

The returned value will be returned by the program.

class ironicclient.osc.v1.baremetal_deploy_template.**ShowBaremetalDeployTemplate**(*app: cliff.app.App*, *app_args: Optional[argpars*, *cmd_name: Optional[str] = None*)

Bases: `osc_lib.command.command.ShowOne`

Show baremetal deploy template details.

get_parser(*prog_name*)

Return an `argparse.ArgumentParser`.

log = <Logger ironicclient.osc.v1.baremetal_deploy_template.ShowBaremetalDeployTemplate (WARNING)>

take_action(*parsed_args*)

Return a two-part tuple with a tuple of column names and a tuple of values.

class ironicclient.osc.v1.baremetal_deploy_template.**UnsetBaremetalDeployTemplate**(*app: _app.App*, *app_args: Optional[argpar*, *cmd_name: Optional[str] = None*)

Bases: `osc_lib.command.command.Command`

Unset baremetal deploy template properties.

get_parser(*prog_name*)

Return an `argparse.ArgumentParser`.

log = <Logger ironicclient.osc.v1.baremetal_deploy_template.UnsetBaremetalDeployTemplate (WARNING)>

take_action(*parsed_args*)

Override to do something useful.

The returned value will be returned by the program.

ironicclient.osc.v1.baremetal_driver module

class ironicclient.osc.v1.baremetal_driver.**ListBaremetalDriver**(*app: cliff.app.App*, *app_args: Optional[argparse.Namespace]*, *cmd_name: Optional[str] = None*)

Bases: `osc_lib.command.command.Lister`

List the enabled drivers.

get_parser(*prog_name*)

Return an `argparse.ArgumentParser`.

log = <Logger ironicclient.osc.v1.baremetal_driver.ListBaremetalDriver (WARNING)>

take_action(*parsed_args*)

Run command.

Return a tuple containing the column names and an iterable containing the data to be listed.

class ironicclient.osc.v1.baremetal_driver.**ListBaremetalDriverProperty**(*app: cliff.app.App*, *app_args: Optional[argparse.Namespac*, *cmd_name: Optional[str] = None*)

Bases: `osc_lib.command.command.Lister`

List the driver properties.

get_parser(*prog_name*)

Return an `argparse.ArgumentParser`.

log = <Logger ironicclient.osc.v1.baremetal_driver.ListBaremetalDriverProperty (WARNING)>

take_action(*parsed_args*)

Run command.

Return a tuple containing the column names and an iterable containing the data to be listed.

`class ironicclient.osc.v1.baremetal_driver.ListBaremetalDriverRaidProperty`(*app: cliff.app.App, app_args: Optional[argparse.Name*
cmd_name: Optional[str] = None)

Bases: `osc_lib.command.command.Lister`

List a drivers RAID logical disk properties.

`get_parser`(*prog_name*)

Return an `argparse.ArgumentParser`.

`log = <Logger ironicclient.osc.v1.baremetal_driver.ListBaremetalDriverRaidProperty (WARNING)>`

`take_action`(*parsed_args*)

Run command.

Return a tuple containing the column names and an iterable containing the data to be listed.

`class ironicclient.osc.v1.baremetal_driver.PassthruCallBaremetalDriver`(*app: cliff.app.App, app_args: Optional[argparse.Namespac*
cmd_name: Optional[str] = None)

Bases: `osc_lib.command.command.ShowOne`

Call a vendor passthru method for a driver.

`get_parser`(*prog_name*)

Return an `argparse.ArgumentParser`.

`log = <Logger ironicclient.osc.v1.baremetal_driver.PassthruCallBaremetalDriver (WARNING)>`

`take_action`(*parsed_args*)

Return a two-part tuple with a tuple of column names and a tuple of values.

class ironicclient.osc.v1.baremetal_driver.**PassthruListBaremetalDriver**(*app: cliff.app.App, app_args: Optional[argparse.Namespac cmd_name: Optional[str] = None*)

Bases: `osc_lib.command.command.Lister`

List available vendor passthru methods for a driver.

get_parser(*prog_name*)

Return an `argparse.ArgumentParser`.

log = <Logger ironicclient.osc.v1.baremetal_driver.PassthruListBaremetalDriver (WARNING)>

take_action(*parsed_args*)

Run command.

Return a tuple containing the column names and an iterable containing the data to be listed.

class ironicclient.osc.v1.baremetal_driver.**ShowBaremetalDriver**(*app: cliff.app.App, app_args: Optional[argparse.Namespace], cmd_name: Optional[str] = None*)

Bases: `osc_lib.command.command.ShowOne`

Show information about a driver.

get_parser(*prog_name*)

Return an `argparse.ArgumentParser`.

log = <Logger ironicclient.osc.v1.baremetal_driver.ShowBaremetalDriver (WARNING)>

take_action(*parsed_args*)

Return a two-part tuple with a tuple of column names and a tuple of values.

ironicclient.osc.v1.baremetal_node module

class ironicclient.osc.v1.baremetal_node.**AbortBaremetalNode**(*app: _app.App, app_args: Optional[argparse.Namespace], cmd_name: Optional[str] = None*)

Bases: *ironicclient.osc.v1.baremetal_node.ProvisionStateBaremetalNode*

Set provision state of baremetal node to abort

PROVISION_STATE = 'abort'

log = <Logger ironicclient.osc.v1.baremetal_node.AbortBaremetalNode (WARNING)>

class `ironicclient.osc.v1.baremetal_node.`**AddTraitBaremetalNode**(*app: _app.App, app_args: Optional[argparse.Namespace], cmd_name: Optional[str] = None*)

Bases: `osc_lib.command.command.Command`

Add traits to a node.

get_parser(*prog_name*)

Return an `argparse.ArgumentParser`.

log = <Logger ironicclient.osc.v1.baremetal_node.AddTraitBaremetalNode (WARNING)>

take_action(*parsed_args*)

Override to do something useful.

The returned value will be returned by the program.

class `ironicclient.osc.v1.baremetal_node.`**AdoptBaremetalNode**(*app: _app.App, app_args: Optional[argparse.Namespace], cmd_name: Optional[str] = None*)

Bases: *`ironicclient.osc.v1.baremetal_node.ProvisionStateWithWait`*

Set provision state of baremetal node to adopt

PROVISION_STATE = 'adopt'

log = <Logger ironicclient.osc.v1.baremetal_node.AdoptBaremetalNode (WARNING)>

class `ironicclient.osc.v1.baremetal_node.`**BIOSSettingShowBaremetalNode**(*app: cliff.app.App, app_args: Optional[argparse.Namespace cmd_name: Optional[str] = None*)

Bases: `osc_lib.command.command.ShowOne`

Show a specific BIOS setting for a node.

get_parser(*prog_name*)

Return an `argparse.ArgumentParser`.

log = <Logger ironicclient.osc.v1.baremetal_node.BIOSSettingShowBaremetalNode (WARNING)>

take_action(*parsed_args*)

Return a two-part tuple with a tuple of column names and a tuple of values.

class ironicclient.osc.v1.baremetal_node.**BootdeviceSetBaremetalNode**(*app: _app.App, app_args: Optional[argparse.Namespace], cmd_name: Optional[str] = None*)

Bases: `osc_lib.command.command.Command`

Set the boot device for a node

get_parser(*prog_name*)

Return an `argparse.ArgumentParser`.

log = <Logger ironicclient.osc.v1.baremetal_node.BootdeviceSetBaremetalNode (WARNING)>

take_action(*parsed_args*)

Override to do something useful.

The returned value will be returned by the program.

class ironicclient.osc.v1.baremetal_node.**BootdeviceShowBaremetalNode**(*app: cliff.app.App, app_args: Optional[argparse.Namespace], cmd_name: Optional[str] = None*)

Bases: `osc_lib.command.command.ShowOne`

Show the boot device information for a node

get_parser(*prog_name*)

Return an `argparse.ArgumentParser`.

log = <Logger ironicclient.osc.v1.baremetal_node.BootdeviceShowBaremetalNode (WARNING)>

take_action(*parsed_args*)

Return a two-part tuple with a tuple of column names and a tuple of values.

class ironicclient.osc.v1.baremetal_node.**BootmodeSetBaremetalNode**(*app: _app.App, app_args: Optional[argparse.Namespace], cmd_name: Optional[str] = None*)

Bases: `osc_lib.command.command.Command`

Set the boot mode for the next baremetal node deployment

`get_parser`(*prog_name*)

Return an `argparse.ArgumentParser`.

`log = <Logger ironicclient.osc.v1.baremetal_node.BootmodeSetBaremetalNode (WARNING)>`

`take_action`(*parsed_args*)

Override to do something useful.

The returned value will be returned by the program.

class `ironicclient.osc.v1.baremetal_node.`**`CleanBaremetalNode`**(*app: _app.App, app_args: Optional[argparse.Namespace], cmd_name: Optional[str] = None*)

Bases: *`ironicclient.osc.v1.baremetal_node.ProvisionStateWithWait`*

Set provision state of baremetal node to clean

`PROVISION_STATE = 'clean'`

`get_parser`(*prog_name*)

Return an `argparse.ArgumentParser`.

`log = <Logger ironicclient.osc.v1.baremetal_node.CleanBaremetalNode (WARNING)>`

class `ironicclient.osc.v1.baremetal_node.`**`ConsoleDisableBaremetalNode`**(*app: _app.App, app_args: Optional[argparse.Namespace], cmd_name: Optional[str] = None*)

Bases: `osc_lib.command.command.Command`

Disable console access for a node

`get_parser`(*prog_name*)

Return an `argparse.ArgumentParser`.

`log = <Logger ironicclient.osc.v1.baremetal_node.ConsoleDisableBaremetalNode (WARNING)>`

`take_action`(*parsed_args*)

Override to do something useful.

The returned value will be returned by the program.

class ironicclient.osc.v1.baremetal_node.**ConsoleEnableBaremetalNode**(*app: _app.App, app_args: Optional[argparse.Namespace], cmd_name: Optional[str] = None*)

Bases: `osc_lib.command.command.Command`

Enable console access for a node

get_parser(*prog_name*)

Return an `argparse.ArgumentParser`.

log = <Logger ironicclient.osc.v1.baremetal_node.ConsoleEnableBaremetalNode (WARNING)>

take_action(*parsed_args*)

Override to do something useful.

The returned value will be returned by the program.

class ironicclient.osc.v1.baremetal_node.**ConsoleShowBaremetalNode**(*app: cliff.app.App, app_args: Optional[argparse.Namespace], cmd_name: Optional[str] = None*)

Bases: `osc_lib.command.command.ShowOne`

Show console information for a node

get_parser(*prog_name*)

Return an `argparse.ArgumentParser`.

log = <Logger ironicclient.osc.v1.baremetal_node.ConsoleShowBaremetalNode (WARNING)>

take_action(*parsed_args*)

Return a two-part tuple with a tuple of column names and a tuple of values.

class ironicclient.osc.v1.baremetal_node.**CreateBaremetalNode**(*app: cliff.app.App, app_args: Optional[argparse.Namespace], cmd_name: Optional[str] = None*)

Bases: `osc_lib.command.command.ShowOne`

Register a new node with the baremetal service

get_parser(*prog_name*)

Return an `argparse.ArgumentParser`.

log = <Logger ironicclient.osc.v1.baremetal_node.CreateBaremetalNode (WARNING)>

take_action(*parsed_args*)

Return a two-part tuple with a tuple of column names and a tuple of values.

class ironicclient.osc.v1.baremetal_node.**DeleteBaremetalNode**(*app: _app.App, app_args: Optional[argparse.Namespace], cmd_name: Optional[str] = None*)

Bases: `osc_lib.command.command.Command`

Unregister baremetal node(s)

get_parser(*prog_name*)

Return an `argparse.ArgumentParser`.

log = <Logger ironicclient.osc.v1.baremetal_node.DeleteBaremetalNode (WARNING)>

take_action(*parsed_args*)

Override to do something useful.

The returned value will be returned by the program.

class ironicclient.osc.v1.baremetal_node.**DeployBaremetalNode**(*app: _app.App, app_args: Optional[argparse.Namespace], cmd_name: Optional[str] = None*)

Bases: *ironicclient.osc.v1.baremetal_node.ProvisionStateWithWait*

Set provision state of baremetal node to deploy

PROVISION_STATE = 'active'

get_parser(*prog_name*)

Return an `argparse.ArgumentParser`.

log = <Logger ironicclient.osc.v1.baremetal_node.DeployBaremetalNode (WARNING)>

class ironicclient.osc.v1.baremetal_node.**InjectNmiBaremetalNode**(*app: _app.App, app_args: Optional[argparse.Namespace], cmd_name: Optional[str] = None*)

Bases: `osc_lib.command.command.Command`

Inject NMI to baremetal node

get_parser(*prog_name*)

Return an `argparse.ArgumentParser`.

log = <Logger ironicclient.osc.v1.baremetal_node.InjectNmiBaremetalNode (WARNING)>

take_action(*parsed_args*)

Override to do something useful.

The returned value will be returned by the program.

class ironicclient.osc.v1.baremetal_node.**InspectBaremetalNode**(*app: _app.App, app_args: Optional[argparse.Namespace], cmd_name: Optional[str] = None*)

Bases: *ironicclient.osc.v1.baremetal_node.ProvisionStateWithWait*

Set provision state of baremetal node to inspect

PROVISION_STATE = 'inspect'

log = <Logger ironicclient.osc.v1.baremetal_node.InspectBaremetalNode (WARNING)>

class ironicclient.osc.v1.baremetal_node.**ListBIOSSettingBaremetalNode**(*app: cliff.app.App, app_args: Optional[argparse.Namespace*

cmd_name: Optional[str] = None)

Bases: osc_lib.command.command.Lister

List a nodes BIOS settings.

get_parser(*prog_name*)

Return an argparse.ArgumentParser.

log = <Logger ironicclient.osc.v1.baremetal_node.ListBIOSSettingBaremetalNode (WARNING)>

take_action(*parsed_args*)

Run command.

Return a tuple containing the column names and an iterable containing the data to be listed.

class ironicclient.osc.v1.baremetal_node.**ListBaremetalNode**(*app: cliff.app.App, app_args: Optional[argparse.Namespace], cmd_name: Optional[str] = None*)

Bases: osc_lib.command.command.Lister

List baremetal nodes

PROVISION_STATES = ['active', 'deleted', 'rebuild', 'inspect', 'provide', 'manage', 'clean', 'adopt', 'abort']

get_parser(*prog_name*)

Return an argparse.ArgumentParser.

log = <Logger ironicclient.osc.v1.baremetal_node.ListBaremetalNode (WARNING)>

take_action(*parsed_args*)

Run command.

Return a tuple containing the column names and an iterable containing the data to be listed.

class ironicclient.osc.v1.baremetal_node.**ListTraitsBaremetalNode**(*app: cliff.app.App, app_args: Optional[argparse.Namespace], cmd_name: Optional[str] = None*)

Bases: osc_lib.command.command.Lister

List a nodes traits.

get_parser(*prog_name*)

Return an argparse.ArgumentParser.

log = <Logger ironicclient.osc.v1.baremetal_node.ListTraitsBaremetalNode (WARNING)>

take_action(*parsed_args*)

Run command.

Return a tuple containing the column names and an iterable containing the data to be listed.

class ironicclient.osc.v1.baremetal_node.**MaintenanceSetBaremetalNode**(*app: _app.App, app_args: Optional[argparse.Namespace], cmd_name: Optional[str] = None*)

Bases: osc_lib.command.command.Command

Set baremetal node to maintenance mode

get_parser(*prog_name*)

Return an argparse.ArgumentParser.

log = <Logger ironicclient.osc.v1.baremetal_node.MaintenanceSetBaremetalNode (WARNING)>

take_action(*parsed_args*)

Override to do something useful.

The returned value will be returned by the program.

class ironicclient.osc.v1.baremetal_node.**MaintenanceUnsetBaremetalNode**(*app: _app.App, app_args: Optional[argparse.Namespac cmd_name: Optional[str] = None*)

Bases: osc_lib.command.command.Command

Unset baremetal node from maintenance mode

get_parser(*prog_name*)

Return an argparse.ArgumentParser.

log = <Logger ironicclient.osc.v1.baremetal_node.MaintenanceUnsetBaremetalNode (WARNING)>

take_action(*parsed_args*)

Override to do something useful.

The returned value will be returned by the program.

class ironicclient.osc.v1.baremetal_node.**ManageBaremetalNode**(*app: _app.App, app_args: Optional[argparse.Namespace], cmd_name: Optional[str] = None*)

Bases: *ironicclient.osc.v1.baremetal_node.ProvisionStateWithWait*

Set provision state of baremetal node to manage

PROVISION_STATE = 'manage'

log = <Logger ironicclient.osc.v1.baremetal_node.ManageBaremetalNode (WARNING)>

class ironicclient.osc.v1.baremetal_node.**NodeHistoryEventGet**(*app: cliff.app.App, app_args: Optional[argparse.Namespace], cmd_name: Optional[str] = None*)

Bases: osc_lib.command.command.ShowOne

Get history event for a baremetal node.

get_parser(*prog_name*)

Return an argparse.ArgumentParser.

log = <Logger ironicclient.osc.v1.baremetal_node.NodeHistoryEventGet (WARNING)>

take_action(*parsed_args*)

Return a two-part tuple with a tuple of column names and a tuple of values.

class ironicclient.osc.v1.baremetal_node.**NodeHistoryList**(*app: cliff.app.App, app_args: Optional[argparse.Namespace], cmd_name: Optional[str] = None*)

Bases: `osc_lib.command.command.Lister`

Get history events for a baremetal node.

get_parser(*prog_name*)

Return an `argparse.ArgumentParser`.

log = <Logger ironicclient.osc.v1.baremetal_node.NodeHistoryList (WARNING)>

take_action(*parsed_args*)

Run command.

Return a tuple containing the column names and an iterable containing the data to be listed.

class ironicclient.osc.v1.baremetal_node.**PassthruCallBaremetalNode**(*app: _app.App, app_args: Optional[argparse.Namespace], cmd_name: Optional[str] = None*)

Bases: `osc_lib.command.command.Command`

Call a vendor passthu method for a node

get_parser(*prog_name*)

Return an `argparse.ArgumentParser`.

log = <Logger ironicclient.osc.v1.baremetal_node.PassthuCallBaremetalNode (WARNING)>

take_action(*parsed_args*)

Override to do something useful.

The returned value will be returned by the program.

class ironicclient.osc.v1.baremetal_node.**PassthruListBaremetalNode**(*app: cliff.app.App, app_args: Optional[argparse.Namespace], cmd_name: Optional[str] = None*)

Bases: `osc_lib.command.command.Lister`

List vendor passthru methods for a node

get_parser(*prog_name*)

Return an `argparse.ArgumentParser`.

log = <Logger ironicclient.osc.v1.baremetal_node.PassthruListBaremetalNode (WARNING)>

take_action(*parsed_args*)

Run command.

Return a tuple containing the column names and an iterable containing the data to be listed.

class `ironicclient.osc.v1.baremetal_node.`**PowerBaremetalNode**(*app: _app.App, app_args: Optional[argparse.Namespace], cmd_name: Optional[str] = None*)

Bases: `osc_lib.command.command.Command`

Base power state class, for setting the power of a node

get_parser(*prog_name*)

Return an `argparse.ArgumentParser`.

log = <Logger ironicclient.osc.v1.baremetal_node.PowerBaremetalNode (WARNING)>

take_action(*parsed_args*)

Override to do something useful.

The returned value will be returned by the program.

class `ironicclient.osc.v1.baremetal_node.`**PowerOffBaremetalNode**(*app: _app.App, app_args: Optional[argparse.Namespace], cmd_name: Optional[str] = None*)

Bases: *ironicclient.osc.v1.baremetal_node.PowerBaremetalNode*

Power off a node

POWER_STATE = 'off'

get_parser(*prog_name*)

Return an `argparse.ArgumentParser`.

log = <Logger ironicclient.osc.v1.baremetal_node.PowerOffBaremetalNode (WARNING)>

class ironicclient.osc.v1.baremetal_node.**PowerOnBaremetalNode**(*app: _app.App*, *app_args: Optional[argparse.Namespace]*, *cmd_name: Optional[str] = None*)

Bases: *ironicclient.osc.v1.baremetal_node.PowerBaremetalNode*

Power on a node

POWER_STATE = 'on'

log = <Logger ironicclient.osc.v1.baremetal_node.PowerOnBaremetalNode (WARNING)>

class ironicclient.osc.v1.baremetal_node.**ProvideBaremetalNode**(*app: _app.App*, *app_args: Optional[argparse.Namespace]*, *cmd_name: Optional[str] = None*)

Bases: *ironicclient.osc.v1.baremetal_node.ProvisionStateWithWait*

Set provision state of baremetal node to provide

PROVISION_STATE = 'provide'

log = <Logger ironicclient.osc.v1.baremetal_node.ProvideBaremetalNode (WARNING)>

class ironicclient.osc.v1.baremetal_node.**ProvisionStateBaremetalNode**(*app: _app.App*, *app_args: Optional[argparse.Namespace]*, *cmd_name: Optional[str] = None*)

Bases: osc_lib.command.command.Command

Base provision state class

get_parser(*prog_name*)

Return an argparse.ArgumentParser.

log = <Logger ironicclient.osc.v1.baremetal_node.ProvisionStateBaremetalNode (WARNING)>

take_action(*parsed_args*)

Override to do something useful.

The returned value will be returned by the program.

class ironicclient.osc.v1.baremetal_node.**ProvisionStateWithWait**(*app: _app.App*, *app_args: Optional[argparse.Namespace]*, *cmd_name: Optional[str] = None*)

Bases: *ironicclient.osc.v1.baremetal_node.ProvisionStateBaremetalNode*

Provision state class adding wait flag.

get_parser(*prog_name*)

Return an `argparse.ArgumentParser`.

log = <Logger ironicclient.osc.v1.baremetal_node.ProvisionStateWithWait (WARNING)>

take_action(*parsed_args*)

Override to do something useful.

The returned value will be returned by the program.

class ironicclient.osc.v1.baremetal_node.**RebootBaremetalNode**(*app: _app.App*, *app_args: Optional[argparse.Namespace]*, *cmd_name: Optional[str] = None*)

Bases: `osc_lib.command.command.Command`

Reboot baremetal node

get_parser(*prog_name*)

Return an `argparse.ArgumentParser`.

log = <Logger ironicclient.osc.v1.baremetal_node.RebootBaremetalNode (WARNING)>

take_action(*parsed_args*)

Override to do something useful.

The returned value will be returned by the program.

class ironicclient.osc.v1.baremetal_node.**RebuildBaremetalNode**(*app: _app.App*, *app_args: Optional[argparse.Namespace]*, *cmd_name: Optional[str] = None*)

Bases: *ironicclient.osc.v1.baremetal_node.ProvisionStateWithWait*

Set provision state of baremetal node to rebuild

PROVISION_STATE = 'rebuild'

get_parser(*prog_name*)

Return an `argparse.ArgumentParser`.

log = <Logger ironicclient.osc.v1.baremetal_node.RebuildBaremetalNode (WARNING)>

class ironicclient.osc.v1.baremetal_node.**RemoveTraitBaremetalNode**(*app: _app.App, app_args: Optional[argparse.Namespace], cmd_name: Optional[str] = None*)

Bases: osc_lib.command.command.Command

Remove trait(s) from a node.

get_parser(*prog_name*)

Return an argparse.ArgumentParser.

log = <Logger ironicclient.osc.v1.baremetal_node.RemoveTraitBaremetalNode (WARNING)>

take_action(*parsed_args*)

Override to do something useful.

The returned value will be returned by the program.

class ironicclient.osc.v1.baremetal_node.**RescueBaremetalNode**(*app: _app.App, app_args: Optional[argparse.Namespace], cmd_name: Optional[str] = None*)

Bases: *ironicclient.osc.v1.baremetal_node.ProvisionStateWithWait*

Set provision state of baremetal node to rescue

PROVISION_STATE = 'rescue'

get_parser(*prog_name*)

Return an argparse.ArgumentParser.

log = <Logger ironicclient.osc.v1.baremetal_node.RescueBaremetalNode (WARNING)>

class ironicclient.osc.v1.baremetal_node.**SecurebootOffBaremetalNode**(*app: _app.App, app_args: Optional[argparse.Namespace], cmd_name: Optional[str] = None*)

Bases: osc_lib.command.command.Command

Turn secure boot off

get_parser(*prog_name*)

Return an argparse.ArgumentParser.

log = <Logger ironicclient.osc.v1.baremetal_node.SecurebootOffBaremetalNode (WARNING)>

take_action(*parsed_args*)

Override to do something useful.

The returned value will be returned by the program.

class ironicclient.osc.v1.baremetal_node.**SecurebootOnBaremetalNode**(*app: _app.App*, *app_args: Optional[argparse.Namespace]*, *cmd_name: Optional[str] = None*)

Bases: `osc_lib.command.command.Command`

Turn secure boot on

get_parser(*prog_name*)

Return an `argparse.ArgumentParser`.

log = <Logger ironicclient.osc.v1.baremetal_node.SecurebootOnBaremetalNode (WARNING)>

take_action(*parsed_args*)

Override to do something useful.

The returned value will be returned by the program.

class ironicclient.osc.v1.baremetal_node.**SetBaremetalNode**(*app: _app.App*, *app_args: Optional[argparse.Namespace]*, *cmd_name: Optional[str] = None*)

Bases: `osc_lib.command.command.Command`

Set baremetal properties

get_parser(*prog_name*)

Return an `argparse.ArgumentParser`.

log = <Logger ironicclient.osc.v1.baremetal_node.SetBaremetalNode (WARNING)>

take_action(*parsed_args*)

Override to do something useful.

The returned value will be returned by the program.

class ironicclient.osc.v1.baremetal_node.**ShowBaremetalNode**(*app: cliff.app.App*, *app_args: Optional[argparse.Namespace]*, *cmd_name: Optional[str] = None*)

Bases: `osc_lib.command.command.ShowOne`

Show baremetal node details

`get_parser`(*prog_name*)

Return an `argparse.ArgumentParser`.

`log = <Logger ironicclient.osc.v1.baremetal_node.ShowBaremetalNode (WARNING)>`

`take_action`(*parsed_args*)

Return a two-part tuple with a tuple of column names and a tuple of values.

class `ironicclient.osc.v1.baremetal_node.`**`UndeployBaremetalNode`**(*app: _app.App, app_args: Optional[argparse.Namespace], cmd_name: Optional[str] = None*)

Bases: *`ironicclient.osc.v1.baremetal_node.ProvisionStateWithWait`*

Set provision state of baremetal node to deleted

`PROVISION_STATE = 'deleted'`

`log = <Logger ironicclient.osc.v1.baremetal_node.UndeployBaremetalNode (WARNING)>`

class `ironicclient.osc.v1.baremetal_node.`**`UnrescueBaremetalNode`**(*app: _app.App, app_args: Optional[argparse.Namespace], cmd_name: Optional[str] = None*)

Bases: *`ironicclient.osc.v1.baremetal_node.ProvisionStateWithWait`*

Set provision state of baremetal node to unrescue

`PROVISION_STATE = 'unrescue'`

`log = <Logger ironicclient.osc.v1.baremetal_node.UnrescueBaremetalNode (WARNING)>`

class `ironicclient.osc.v1.baremetal_node.`**`UnsetBaremetalNode`**(*app: _app.App, app_args: Optional[argparse.Namespace], cmd_name: Optional[str] = None*)

Bases: `osc_lib.command.command.Command`

Unset baremetal properties

`get_parser`(*prog_name*)

Return an `argparse.ArgumentParser`.

log = <Logger ironicclient.osc.v1.baremetal_node.UnsetBaremetalNode (WARNING)>

take_action(*parsed_args*)

Override to do something useful.

The returned value will be returned by the program.

class ironicclient.osc.v1.baremetal_node.**ValidateBaremetalNode**(*app: cliff.app.App, app_args: Optional[argparse.Namespace], cmd_name: Optional[str] = None*)

Bases: `osc_lib.command.command.Lister`

Validate a nodes driver interfaces

get_parser(*prog_name*)

Return an `argparse.ArgumentParser`.

log = <Logger ironicclient.osc.v1.baremetal_node.ValidateBaremetalNode (WARNING)>

take_action(*parsed_args*)

Run command.

Return a tuple containing the column names and an iterable containing the data to be listed.

class ironicclient.osc.v1.baremetal_node.**VifAttachBaremetalNode**(*app: _app.App, app_args: Optional[argparse.Namespace], cmd_name: Optional[str] = None*)

Bases: `osc_lib.command.command.Command`

Attach VIF to a given node

get_parser(*prog_name*)

Return an `argparse.ArgumentParser`.

log = <Logger ironicclient.osc.v1.baremetal_node.VifAttachBaremetalNode (WARNING)>

take_action(*parsed_args*)

Override to do something useful.

The returned value will be returned by the program.

class ironicclient.osc.v1.baremetal_node.**VifDetachBaremetalNode**(*app: _app.App, app_args: Optional[argparse.Namespace], cmd_name: Optional[str] = None*)

Bases: `osc_lib.command.command.Command`

Detach VIF from a given node

`get_parser`(*prog_name*)

Return an `argparse.ArgumentParser`.

`log = <Logger ironicclient.osc.v1.baremetal_node.VifDetachBaremetalNode (WARNING)>`

`take_action`(*parsed_args*)

Override to do something useful.

The returned value will be returned by the program.

`class` `ironicclient.osc.v1.baremetal_node.`**`VifListBaremetalNode`**(*app: cliff.app.App, app_args: Optional[argparse.Namespace], cmd_name: Optional[str] = None*)

Bases: `osc_lib.command.command.Lister`

Show attached VIFs for a node

`get_parser`(*prog_name*)

Return an `argparse.ArgumentParser`.

`log = <Logger ironicclient.osc.v1.baremetal_node.VifListBaremetalNode (WARNING)>`

`take_action`(*parsed_args*)

Run command.

Return a tuple containing the column names and an iterable containing the data to be listed.

ironicclient.osc.v1.baremetal_port module

`class` `ironicclient.osc.v1.baremetal_port.`**`CreateBaremetalPort`**(*app: cliff.app.App, app_args: Optional[argparse.Namespace], cmd_name: Optional[str] = None*)

Bases: `osc_lib.command.command.ShowOne`

Create a new port

`get_parser`(*prog_name*)

Return an `argparse.ArgumentParser`.

`log = <Logger ironicclient.osc.v1.baremetal_port.CreateBaremetalPort (WARNING)>`

take_action(*parsed_args*)

Return a two-part tuple with a tuple of column names and a tuple of values.

class ironicclient.osc.v1.baremetal_port.**DeleteBaremetalPort**(*app: _app.App*, *app_args: Optional[argparse.Namespace]*, *cmd_name: Optional[str] = None*)

Bases: `osc_lib.command.command.Command`

Delete port(s).

get_parser(*prog_name*)

Return an `argparse.ArgumentParser`.

log = <Logger ironicclient.osc.v1.baremetal_port.DeleteBaremetalPort (WARNING)>

take_action(*parsed_args*)

Override to do something useful.

The returned value will be returned by the program.

class ironicclient.osc.v1.baremetal_port.**ListBaremetalPort**(*app: cliff.app.App*, *app_args: Optional[argparse.Namespace]*, *cmd_name: Optional[str] = None*)

Bases: `osc_lib.command.command.Lister`

List baremetal ports.

get_parser(*prog_name*)

Return an `argparse.ArgumentParser`.

log = <Logger ironicclient.osc.v1.baremetal_port.ListBaremetalPort (WARNING)>

take_action(*parsed_args*)

Run command.

Return a tuple containing the column names and an iterable containing the data to be listed.

class ironicclient.osc.v1.baremetal_port.**SetBaremetalPort**(*app: _app.App*, *app_args: Optional[argparse.Namespace]*, *cmd_name: Optional[str] = None*)

Bases: `osc_lib.command.command.Command`

Set baremetal port properties.

get_parser(*prog_name*)

Return an `argparse.ArgumentParser`.

log = <Logger ironicclient.osc.v1.baremetal_port.SetBaremetalPort (WARNING)>

take_action(*parsed_args*)

Override to do something useful.

The returned value will be returned by the program.

class ironicclient.osc.v1.baremetal_port.**ShowBaremetalPort**(*app: cliff.app.App, app_args: Optional[argparse.Namespace], cmd_name: Optional[str] = None*)

Bases: `osc_lib.command.command.ShowOne`

Show baremetal port details.

get_parser(*prog_name*)

Return an `argparse.ArgumentParser`.

log = <Logger ironicclient.osc.v1.baremetal_port.ShowBaremetalPort (WARNING)>

take_action(*parsed_args*)

Return a two-part tuple with a tuple of column names and a tuple of values.

class ironicclient.osc.v1.baremetal_port.**UnsetBaremetalPort**(*app: _app.App, app_args: Optional[argparse.Namespace], cmd_name: Optional[str] = None*)

Bases: `osc_lib.command.command.Command`

Unset baremetal port properties.

get_parser(*prog_name*)

Return an `argparse.ArgumentParser`.

log = <Logger ironicclient.osc.v1.baremetal_port.UnsetBaremetalPort (WARNING)>

take_action(*parsed_args*)

Override to do something useful.

The returned value will be returned by the program.

ironicclient.osc.v1.baremetal_portgroup module

class ironicclient.osc.v1.baremetal_portgroup.**CreateBaremetalPortGroup**(*app: cliff.app.App, app_args: Optional[argparse.Namespac cmd_name: Optional[str] = None*)

Bases: `osc_lib.command.command.ShowOne`

Create a new baremetal port group.

get_parser(*prog_name*)

Return an `argparse.ArgumentParser`.

log = <Logger ironicclient.osc.v1.baremetal_portgroup.CreateBaremetalPortGroup (WARNING)>

take_action(*parsed_args*)

Return a two-part tuple with a tuple of column names and a tuple of values.

class ironicclient.osc.v1.baremetal_portgroup.**DeleteBaremetalPortGroup**(*app: _app.App, app_args: Optional[argparse.Namespac cmd_name: Optional[str] = None*)

Bases: `osc_lib.command.command.Command`

Unregister baremetal port group(s).

get_parser(*prog_name*)

Return an `argparse.ArgumentParser`.

log = <Logger ironicclient.osc.v1.baremetal_portgroup.DeleteBaremetalPortGroup (WARNING)>

take_action(*parsed_args*)

Override to do something useful.

The returned value will be returned by the program.

class ironicclient.osc.v1.baremetal_portgroup.**ListBaremetalPortGroup**(*app: cliff.app.App, app_args: Optional[argparse.Namespace], cmd_name: Optional[str] = None*)

Bases: `osc_lib.command.command.Lister`

List baremetal port groups.

get_parser(*prog_name*)

Return an `argparse.ArgumentParser`.

log = <Logger ironicclient.osc.v1.baremetal_portgroup.ListBaremetalPortGroup (WARNING)>

take_action(*parsed_args*)

Run command.

Return a tuple containing the column names and an iterable containing the data to be listed.

class ironicclient.osc.v1.baremetal_portgroup.**SetBaremetalPortGroup**(*app: _app.App, app_args: Optional[argparse.Namespace], cmd_name: Optional[str] = None*)

Bases: `osc_lib.command.command.Command`

Set baremetal port group properties.

get_parser(*prog_name*)

Return an `argparse.ArgumentParser`.

log = <Logger ironicclient.osc.v1.baremetal_portgroup.SetBaremetalPortGroup (WARNING)>

take_action(*parsed_args*)

Override to do something useful.

The returned value will be returned by the program.

class ironicclient.osc.v1.baremetal_portgroup.**ShowBaremetalPortGroup**(*app: cliff.app.App, app_args: Optional[argparse.Namespace], cmd_name: Optional[str] = None*)

Bases: `osc_lib.command.command.ShowOne`

Show baremetal port group details.

`get_parser`(*prog_name*)

Return an `argparse.ArgumentParser`.

`log = <Logger ironicclient.osc.v1.baremetal_portgroup.ShowBaremetalPortGroup (WARNING)>`

`take_action`(*parsed_args*)

Return a two-part tuple with a tuple of column names and a tuple of values.

class `ironicclient.osc.v1.baremetal_portgroup.`**`UnsetBaremetalPortGroup`**(*app: _app.App*, *app_args: Optional[argparse.Namespace*, *cmd_name: Optional[str] = None*)

Bases: `osc_lib.command.command.Command`

Unset baremetal port group properties.

`get_parser`(*prog_name*)

Return an `argparse.ArgumentParser`.

`log = <Logger ironicclient.osc.v1.baremetal_portgroup.UnsetBaremetalPortGroup (WARNING)>`

`take_action`(*parsed_args*)

Override to do something useful.

The returned value will be returned by the program.

ironicclient.osc.v1.baremetal_volume_connector module

class `ironicclient.osc.v1.baremetal_volume_connector.`**`CreateBaremetalVolumeConnector`**(*app: cliff.app.A*, *app_args: Optional[arg*, *cmd_nam Optional[str = None*)

Bases: `osc_lib.command.command.ShowOne`

Create a new baremetal volume connector.

get_parser(*prog_name*)

Return an `argparse.ArgumentParser`.

log = <Logger ironicclient.osc.v1.baremetal_volume_connector.CreateBaremetalVolumeConnector (WARNING)>

take_action(*parsed_args*)

Return a two-part tuple with a tuple of column names and a tuple of values.

class `ironicclient.osc.v1.baremetal_volume_connector.`**DeleteBaremetalVolumeConnector**(*app: _app.App app_args: Optional[arg cmd_nam Optional[str = None*)

Bases: `osc_lib.command.command.Command`

Unregister baremetal volume connector(s).

get_parser(*prog_name*)

Return an `argparse.ArgumentParser`.

log = <Logger ironicclient.osc.v1.baremetal_volume_connector.DeleteBaremetalVolumeConnector (WARNING)>

take_action(*parsed_args*)

Override to do something useful.

The returned value will be returned by the program.

class `ironicclient.osc.v1.baremetal_volume_connector.`**ListBaremetalVolumeConnector**(*app: cliff.app.App app_args: Optional[argpa cmd_name: Optional[str] = None*)

Bases: `osc_lib.command.command.Lister`

List baremetal volume connectors.

get_parser(*prog_name*)

Return an `argparse.ArgumentParser`.

log = <Logger ironicclient.osc.v1.baremetal_volume_connector.ListBaremetalVolumeConnector (WARNING)>

take_action(*parsed_args*)

Run command.

Return a tuple containing the column names and an iterable containing the data to be listed.

class ironicclient.osc.v1.baremetal_volume_connector.**SetBaremetalVolumeConnector**(*app: _app.App, app_args: Optional[argpar cmd_name: Optional[str] = None*)

Bases: `osc_lib.command.command.Command`

Set baremetal volume connector properties.

get_parser(*prog_name*)

Return an `argparse.ArgumentParser`.

log = <Logger ironicclient.osc.v1.baremetal_volume_connector.SetBaremetalVolumeConnector (WARNING)>

take_action(*parsed_args*)

Override to do something useful.

The returned value will be returned by the program.

class ironicclient.osc.v1.baremetal_volume_connector.**ShowBaremetalVolumeConnector**(*app: cliff.app.App app_args: Optional[argpa cmd_name: Optional[str] = None*)

Bases: `osc_lib.command.command.ShowOne`

Show baremetal volume connector details.

get_parser(*prog_name*)

Return an `argparse.ArgumentParser`.

log = <Logger ironicclient.osc.v1.baremetal_volume_connector.ShowBaremetalVolumeConnector (WARNING)>

take_action(*parsed_args*)

Return a two-part tuple with a tuple of column names and a tuple of values.

class ironicclient.osc.v1.baremetal_volume_connector.**UnsetBaremetalVolumeConnector**(*app: _app.App, app_args: Optional[argp cmd_name Optional[str] = None*)

Bases: `osc_lib.command.command.Command`

Unset baremetal volume connector properties.

get_parser(*prog_name*)

Return an `argparse.ArgumentParser`.

log = <Logger ironicclient.osc.v1.baremetal_volume_connectorUnsetBaremetalVolumeConnector (WARNING)>

take_action(*parsed_args*)

Override to do something useful.

The returned value will be returned by the program.

ironicclient.osc.v1.baremetal_volume_target module

class ironicclient.osc.v1.baremetal_volume_target.**CreateBaremetalVolumeTarget**(*app: cliff.app.App, app_args: Optional[argparse.N cmd_name: Optional[str] = None*)

Bases: `osc_lib.command.command.ShowOne`

Create a new baremetal volume target.

get_parser(*prog_name*)

Return an `argparse.ArgumentParser`.

log = <Logger ironicclient.osc.v1.baremetal_volume_target.CreateBaremetalVolumeTarget (WARNING)>

take_action(*parsed_args*)

Return a two-part tuple with a tuple of column names and a tuple of values.

class ironicclient.osc.v1.baremetal_volume_target.DeleteBaremetalVolumeTarget(*app: _app.App, app_args: Optional[argparse.N cmd_name: Optional[str] = None*)

Bases: osc_lib.command.command.Command

Unregister baremetal volume target(s).

get_parser(*prog_name*)

Return an argparse.ArgumentParser.

log = <Logger ironicclient.osc.v1.baremetal_volume_target.DeleteBaremetalVolumeTarget (WARNING)>

take_action(*parsed_args*)

Override to do something useful.

The returned value will be returned by the program.

class ironicclient.osc.v1.baremetal_volume_target.ListBaremetalVolumeTarget(*app: cliff.app.App, app_args: Optional[argparse.Nam cmd_name: Optional[str] = None*)

Bases: osc_lib.command.command.Lister

List baremetal volume targets.

get_parser(*prog_name*)

Return an argparse.ArgumentParser.

log = <Logger ironicclient.osc.v1.baremetal_volume_target.ListBaremetalVolumeTarget (WARNING)>

take_action(*parsed_args*)

Run command.

Return a tuple containing the column names and an iterable containing the data to be listed.

class ironicclient.osc.v1.baremetal_volume_target.**SetBaremetalVolumeTarget**(*app: _app.App, app_args: Optional[argparse.Name* *cmd_name: Optional[str] = None*)

Bases: osc_lib.command.command.Command

Set baremetal volume target properties.

get_parser(*prog_name*)

Return an argparse.ArgumentParser.

log = <Logger ironicclient.osc.v1.baremetal_volume_target.SetBaremetalVolumeTarget (WARNING)>

take_action(*parsed_args*)

Override to do something useful.

The returned value will be returned by the program.

class ironicclient.osc.v1.baremetal_volume_target.**ShowBaremetalVolumeTarget**(*app: cliff.app.App, app_args: Optional[argparse.Nan* *cmd_name: Optional[str] = None*)

Bases: osc_lib.command.command.ShowOne

Show baremetal volume target details.

get_parser(*prog_name*)

Return an argparse.ArgumentParser.

log = <Logger ironicclient.osc.v1.baremetal_volume_target.ShowBaremetalVolumeTarget (WARNING)>

take_action(*parsed_args*)

Return a two-part tuple with a tuple of column names and a tuple of values.

class ironicclient.osc.v1.baremetal_volume_target.**UnsetBaremetalVolumeTarget**(*app: _app.App*, *app_args: Optional[argparse.Na*, *cmd_name: Optional[str] = None*)

Bases: osc_lib.command.command.Command

Unset baremetal volume target properties.

get_parser(*prog_name*)

Return an argparse.ArgumentParser.

log = <Logger ironicclient.osc.v1.baremetal_volume_targetUnsetBaremetalVolumeTarget (WARNING)>

take_action(*parsed_args*)

Override to do something useful.

The returned value will be returned by the program.

Module contents

Submodules

ironicclient.osc.plugin module

OpenStackClient plugin for Bare Metal service.

class ironicclient.osc.plugin.**ReplaceLatestVersion**(*option_strings*, *dest*, *nargs=None*, *const=None*, *default=None*, *type=None*, *choices=None*, *required=False*, *help=None*, *metavar=None*)

Bases: argparse.Action

Replaces *latest* keyword by last known version.

OSC cannot accept the literal latest as a supported API version as it breaks the major version detection (OSC tries to load configuration options from setuptools entrypoint openstack.baremetal.vlatest). This action replaces latest with the latest known version, and sets the global OS_BAREMETAL_API_LATEST flag appropriately.

ironicclient.osc.plugin.**build_option_parser**(*parser*)

Hook to add global options.

ironicclient.osc.plugin.**make_client**(*instance*)

Returns a baremetal service client.

Module contents

ironicclient.v1 package

Submodules

ironicclient.v1.allocation module

class ironicclient.v1.allocation.**Allocation**(*manager*, *info*, *loaded=False*)

Bases: *ironicclient.common.base.Resource*

class ironicclient.v1.allocation.**AllocationManager**(*api*)

Bases: *ironicclient.common.base.CreateManager*

delete(*allocation_id*, *os_ironic_api_version=None*, *global_request_id=None*)

Delete the Allocation.

Parameters

- **allocation_id** The UUID or name of an allocation.
- **os_ironic_api_version** String version (e.g. 1.35) to use for the request. If not specified, the clients default is used.
- **global_request_id** String containing global request ID header value (in form req-<UUID>) to use for the request.

get(*allocation_id*, *fields=None*, *os_ironic_api_version=None*, *global_request_id=None*)

Get an allocation with the specified identifier.

Parameters

- **allocation_id** The UUID or name of an allocation.
- **fields** Optional, a list with a specified set of fields of the resource to be returned. Can not be used when detail is set.
- **os_ironic_api_version** String version (e.g. 1.35) to use for the request. If not specified, the clients default is used.
- **global_request_id** String containing global request ID header value (in form req-<UUID>) to use for the request.

Returns an *Allocation* object.

list(*resource_class=None*, *state=None*, *node=None*, *limit=None*, *marker=None*, *sort_key=None*, *sort_dir=None*, *fields=None*, *owner=None*, *os_ironic_api_version=None*, *global_request_id=None*)

Retrieve a list of allocations.

Parameters

- **resource_class** Optional, get allocations with this resource class.
- **state** Optional, get allocations in this state. One of `allocating`, `active` or `error`.
- **node** UUID or name of the node of the allocation.

- **marker** Optional, the UUID of an allocation, eg the last allocation from a previous result set. Return the next result set.
- **limit**

  The maximum number of results to return per request, if:

  1) limit > 0, the maximum number of allocations to return.
  2) limit == 0, return the entire list of allocations.
  3) limit == None, the number of items returned respect the maximum imposed by the Ironic API (see Ironics api.max_limit option).

- **sort_key** Optional, field used for sorting.
- **sort_dir** Optional, direction of sorting, either asc (the default) or desc.
- **fields** Optional, a list with a specified set of fields of the resource to be returned.
- **owner** Optional, project that owns the allocation.
- **os_ironic_api_version** String version (e.g. 1.35) to use for the request. If not specified, the clients default is used.
- **global_request_id** String containing global request ID header value (in form req-<UUID>) to use for the request.

Returns A list of allocations.

Raises InvalidAttribute if a subset of fields is requested with detail option set.

resource_class

alias of *ironicclient.v1.allocation.Allocation*

update(*allocation_id*, *patch*, *os_ironic_api_version=None*, *global_request_id=None*)

Updates the Allocation. Only name and extra field are allowed.

Parameters

- **allocation_id** The UUID or name of an allocation.
- **patch** a json PATCH document to apply to this allocation.
- **os_ironic_api_version** String version (e.g. 1.35) to use for the request. If not specified, the clients default is used.
- **global_request_id** String containing global request ID header value (in form req-<UUID>) to use for the request.

wait(*allocation_id*, *timeout=0*, *poll_interval=1*, *poll_delay_function=None*, *os_ironic_api_version=None*, *global_request_id=None*)

Wait for the Allocation to become active.

Parameters

- **timeout** timeout in seconds, no timeout if 0.
- **poll_interval** interval in seconds between polls.

- **poll_delay_function** function to use to wait between polls (defaults to time.sleep). Should take one argument - delay time in seconds. Any exceptions raised inside it will abort the wait.
- **os_ironic_api_version** String version (e.g. 1.35) to use for the request. If not specified, the clients default is used.
- **global_request_id** String containing global request ID header value (in form req-<UUID>) to use for the request.

Returns updated *Allocation* object.

Raises StateTransitionFailed if allocation reaches the error state.

Raises StateTransitionTimeout on timeout.

ironicclient.v1.chassis module

class ironicclient.v1.chassis.**Chassis**(*manager*, *info*, *loaded=False*)

Bases: *ironicclient.common.base.Resource*

class ironicclient.v1.chassis.**ChassisManager**(*api*)

Bases: *ironicclient.common.base.CreateManager*

delete(*chassis_id*, *os_ironic_api_version=None*, *global_request_id=None*)

get(*chassis_id*, *fields=None*, *os_ironic_api_version=None*, *global_request_id=None*)

list(*marker=None*, *limit=None*, *sort_key=None*, *sort_dir=None*, *detail=False*, *fields=None*, *os_ironic_api_version=None*, *global_request_id=None*)

Retrieve a list of chassis.

Parameters

- **marker** Optional, the UUID of a chassis, eg the last chassis from a previous result set. Return the next result set.
- **limit**

  The maximum number of results to return per request, if:

  1) limit > 0, the maximum number of chassis to return.
  2) limit == 0, return the entire list of chassis.
  3) limit param is NOT specified (None), the number of items returned respect the maximum imposed by the Ironic API (see Ironics api.max_limit option).

- **sort_key** Optional, field used for sorting.
- **sort_dir** Optional, direction of sorting, either asc (the default) or desc.
- **detail** Optional, boolean whether to return detailed information about chassis.
- **fields** Optional, a list with a specified set of fields of the resource to be returned. Can not be used when detail is set.

- `os_ironic_api_version` String version (e.g. 1.35) to use for the request. If not specified, the clients default is used.
- `global_request_id` String containing global request ID header value (in form req-<UUID>) to use for the request.

Returns A list of chassis.

`list_nodes`(*chassis_id*, *marker=None*, *limit=None*, *sort_key=None*, *sort_dir=None*, *detail=False*, *fields=None*, *associated=None*, *maintenance=None*, *provision_state=None*, *os_ironic_api_version=None*, *global_request_id=None*)

List all the nodes for a given chassis.

Parameters

- `chassis_id` The UUID of the chassis.
- `marker` Optional, the UUID of a node, eg the last node from a previous result set. Return the next result set.
- `limit`

  The maximum number of results to return per request, if:

  1) limit > 0, the maximum number of nodes to return.
  2) limit == 0, return the entire list of nodes.
  3) limit param is NOT specified (None), the number of items returned respect the maximum imposed by the Ironic API (see Ironics api.max_limit option).

- `sort_key` Optional, field used for sorting.
- `sort_dir` Optional, direction of sorting, either asc (the default) or desc.
- `detail` Optional, boolean whether to return detailed information about nodes.
- `fields` Optional, a list with a specified set of fields of the resource to be returned. Can not be used when detail is set.
- `associated` Optional. Either a Boolean or a string representation of a Boolean that indicates whether to return a list of associated (True or True) or unassociated (False or False) nodes.
- `maintenance` Optional. Either a Boolean or a string representation of a Boolean that indicates whether to return nodes in maintenance mode (True or True), or not in maintenance mode (False or False).
- `provision_state` Optional. String value to get only nodes in that provision state.
- `os_ironic_api_version` String version (e.g. 1.35) to use for the request. If not specified, the clients default is used.
- `global_request_id` String containing global request ID header value (in form req-<UUID>) to use for the request.

Returns A list of nodes.

resource_class

alias of *ironicclient.v1.chassis.Chassis*

update(*chassis_id*, *patch*, *os_ironic_api_version=None*, *global_request_id=None*)

ironicclient.v1.client module

class ironicclient.v1.client.**Client**(*endpoint_override=None*, **args*, ***kwargs*)

Bases: object

Client for the Ironic v1 API.

Parameters

- **endpoint_override** (*string*) A user-supplied endpoint URL for the ironic service.
- **session** A keystoneauth Session object (must be provided as a keyword argument).

property current_api_version

Return the current API version in use.

This returns the version of the REST API that the API client is presently set to request. This value may change as a result of API version negotiation.

property is_api_version_negotiated

Returns True if microversion negotiation has occured.

negotiate_api_version()

Triggers negotiation with the remote API endpoint.

Returns the negotiated API version.

ironicclient.v1.conductor module

class ironicclient.v1.conductor.**Conductor**(*manager*, *info*, *loaded=False*)

Bases: *ironicclient.common.base.Resource*

class ironicclient.v1.conductor.**ConductorManager**(*api*)

Bases: *ironicclient.common.base.Manager*

get(*hostname*, *fields=None*, *os_ironic_api_version=None*, *global_request_id=None*)

list(*marker=None*, *limit=None*, *sort_key=None*, *sort_dir=None*, *fields=None*, *detail=False*, *os_ironic_api_version=None*, *global_request_id=None*)

Retrieve a list of conductors.

Parameters

- **marker** Optional, the hostname of a conductor, eg the last conductor from a previous result set. Return the next result set.
- **limit**

  The maximum number of results to return per request, if:

1) limit > 0, the maximum number of ports to return.
2) limit == 0, return the entire list of ports.
3) limit param is NOT specified (None), the number of items returned respect the maximum imposed by the Ironic API (see Ironics api.max_limit option).

- **sort_key** Optional, field used for sorting.
- **sort_dir** Optional, direction of sorting, either asc (the default) or desc.
- **fields** Optional, a list with a specified set of fields of the resource to be returned. Can not be used when detail is set.
- **detail** Optional, boolean whether to return detailed information about conductors.
- **os_ironic_api_version** String version (e.g. 1.35) to use for the request. If not specified, the clients default is used.
- **global_request_id** String containing global request ID header value (in form req-<UUID>) to use for the request.

Returns A list of conductors.

`resource_class`

alias of *ironicclient.v1.conductor.Conductor*

ironicclient.v1.create_resources module

ironicclient.v1.create_resources.**create_chassis**(*client*, *chassis_list*)

Create chassis from dictionaries.

Parameters

- **client** ironic client instance.
- **chassis_list** list of dictionaries to be POSTed to /chassis endpoint, if some of them contain nodes key, its content is POSTed separately to /nodes endpoint.

Returns array of exceptions encountered during creation.

ironicclient.v1.create_resources.**create_nodes**(*client*, *node_list*, *chassis_uuid=None*)

Create nodes from dictionaries.

Parameters

- **client** ironic client instance.
- **node_list** list of dictionaries to be POSTed to /nodes endpoint, if some of them contain ports key, its content is POSTed separately to /ports endpoint.
- **chassis_uuid** UUID of a chassis the nodes should be associated with.

Returns array of exceptions encountered during creation.

ironicclient.v1.create_resources.**create_portgroups**(*client*, *portgroup_list*, *node_uuid*)

Create port groups from dictionaries.

Parameters

- **client** ironic client instance.
- **portgroup_list** list of dictionaries to be POSTed to /portgroups endpoint, if some of them contain ports key, its content is POSTed separately to /ports endpoint.
- **node_uuid** UUID of a node the port groups should be associated with.

Returns array of exceptions encountered during creation.

ironicclient.v1.create_resources.**create_ports**(*client*, *port_list*, *node_uuid*, *portgroup_uuid=None*)

Create ports from dictionaries.

Parameters

- **client** ironic client instance.
- **port_list** list of dictionaries to be POSTed to /ports endpoint.
- **node_uuid** UUID of a node the ports should be associated with.
- **portgroup_uuid** UUID of a port group the ports should be associated with, if they are its members.

Returns array of exceptions encountered during creation.

ironicclient.v1.create_resources.**create_resources**(*client*, *filenames*)

Create resources using their JSON or YAML descriptions.

Parameters

- **client** an instance of ironic client;
- **filenames** a list of filenames containing JSON or YAML resources definitions.

Raises ClientException if any operation during files processing/resource creation fails.

ironicclient.v1.create_resources.**create_single_chassis**(*client*, ***params*)

Call the client to create a chassis.

Parameters

- **client** ironic client instance.
- **params** dictionary to be POSTed to /chassis endpoint, excluding nodes key.

Returns UUID of the created chassis or None in case of exception, and an exception, if it appears.

Raises InvalidAttribute, if some parameters passed to clients create_method are invalid.

Raises ClientException, if the creation of the chassis fails.

ironicclient.v1.create_resources.**create_single_handler**(*resource_type*)

Catch errors of the creation of a single resource.

This decorator appends an error (which is an instance of some client exception class) to the return value of the create_method, changing the return value from just UUID to (UUID, error), and does some exception handling.

Parameters **resource_type** string value, the type of the resource being created, e.g. node, used purely for exception messages.

ironicclient.v1.create_resources.**create_single_node**(*client*, ***params*)

Call the client to create a node.

Parameters

- **client** ironic client instance.
- **params** dictionary to be POSTed to /nodes endpoint, excluding ports and portgroups keys.

Returns UUID of the created node or None in case of exception, and an exception, if it appears.

Raises InvalidAttribute, if some parameters passed to clients create_method are invalid.

Raises ClientException, if the creation of the node fails.

ironicclient.v1.create_resources.**create_single_port**(*client*, ***params*)

Call the client to create a port.

Parameters

- **client** ironic client instance.
- **params** dictionary to be POSTed to /ports endpoint.

Returns UUID of the created port or None in case of exception, and an exception, if it appears.

Raises InvalidAttribute, if some parameters passed to clients create_method are invalid.

Raises ClientException, if the creation of the port fails.

ironicclient.v1.create_resources.**create_single_portgroup**(*client*, ***params*)

Call the client to create a port group.

Parameters

- **client** ironic client instance.
- **params** dictionary to be POSTed to /portgroups endpoint, excluding ports key.

Returns UUID of the created port group or None in case of exception, and an exception, if it appears.

Raises InvalidAttribute, if some parameters passed to clients create_method are invalid.

Raises ClientException, if the creation of the portgroup fails.

ironicclient.v1.create_resources.**load_from_file**(*filename*)

Deserialize JSON or YAML from file.

Parameters **filename** name of the file containing JSON or YAML.

Returns a dictionary deserialized from JSON or YAML.

Raises ClientException if the file can not be loaded or if its contents is not a valid JSON or YAML, or if the file extension is not supported.

ironicclient.v1.deploy_template module

class ironicclient.v1.deploy_template.**DeployTemplate**(*manager*, *info*, *loaded=False*)

Bases: *ironicclient.common.base.Resource*

class ironicclient.v1.deploy_template.**DeployTemplateManager**(*api*)

Bases: *ironicclient.common.base.CreateManager*

delete(*template_id*, *os_ironic_api_version=None*, *global_request_id=None*)

get(*template_id*, *fields=None*, *os_ironic_api_version=None*, *global_request_id=None*)

list(*limit=None*, *marker=None*, *sort_key=None*, *sort_dir=None*, *detail=False*, *fields=None*, *os_ironic_api_version=None*, *global_request_id=None*)

Retrieve a list of deploy templates.

Parameters

- **marker** Optional, the UUID of a deploy template, eg the last template from a previous result set. Return the next result set.
- **limit**

  The maximum number of results to return per request, if:

  1) limit > 0, the maximum number of deploy templates to return.
  2) limit == 0, return the entire list of deploy templates.
  3) limit param is NOT specified (None), the number of items returned respect the maximum imposed by the Ironic API (see Ironics api.max_limit option).

- **sort_key** Optional, field used for sorting.
- **sort_dir** Optional, direction of sorting, either asc (the default) or desc.
- **detail** Optional, boolean whether to return detailed information about deploy templates.
- **fields** Optional, a list with a specified set of fields of the resource to be returned. Can not be used when detail is set.
- **os_ironic_api_version** String version (e.g. 1.35) to use for the request. If not specified, the clients default is used.
- **global_request_id** String containing global request ID header value (in form req-<UUID>) to use for the request.

Returns A list of deploy templates.

resource_class

alias of *ironicclient.v1.deploy_template.DeployTemplate*

update(*template_id*, *patch*, *os_ironic_api_version=None*, *global_request_id=None*)

ironicclient.v1.driver module

class ironicclient.v1.driver.**Driver**(*manager*, *info*, *loaded=False*)

Bases: *ironicclient.common.base.Resource*

class ironicclient.v1.driver.**DriverManager**(*api*)

Bases: *ironicclient.common.base.Manager*

delete(*driver_name*, *os_ironic_api_version=None*, *global_request_id=None*)

get(*driver_name*, *os_ironic_api_version=None*, *global_request_id=None*, *fields=None*)

get_vendor_passthru_methods(*driver_name*, *os_ironic_api_version=None*, *global_request_id=None*)

list(*driver_type=None*, *detail=None*, *os_ironic_api_version=None*, *global_request_id=None*, *fields=None*)

Retrieve a list of drivers.

Parameters

- **driver_type** Optional, string to filter the drivers by type. Value should be classic or dynamic.
- **detail** Optional, flag whether to return detailed information about drivers. Default is None means not to send the arg to the server due to older versions of the server cannot handle filtering on detail.
- **os_ironic_api_version** String version (e.g. 1.35) to use for the request. If not specified, the clients default is used.
- **global_request_id** String containing global request ID header value (in form req-<UUID>) to use for the request.
- **fields** Optional, a list with a specified set of fields of the resource to be returned. Can not be used when detail is set.

Returns A list of drivers.

properties(*driver_name*, *os_ironic_api_version=None*, *global_request_id=None*)

raid_logical_disk_properties(*driver_name*, *os_ironic_api_version=None*, *global_request_id=None*)

Returns the RAID logical disk properties for the driver.

Parameters

- **driver_name** Name of the driver.
- **os_ironic_api_version** String version (e.g. 1.35) to use for the request. If not specified, the clients default is used.

- **global_request_id** String containing global request ID header value (in form req-<UUID>) to use for the request.

Returns A dictionary containing the properties that can be mentioned for RAID logical disks and a textual description for them. It returns an empty dictionary on error.

resource_class

alias of *ironicclient.v1.driver.Driver*

update(*driver_name*, *patch*, *http_method='PATCH'*, *os_ironic_api_version=None*, *global_request_id=None*)

vendor_passthru(*driver_name*, *method*, *args=None*, *http_method=None*, *os_ironic_api_version=None*, *global_request_id=None*)

Issue requests for vendor-specific actions on a given driver.

Parameters

- **driver_name** The name of the driver.
- **method** Name of the vendor method.
- **args** Optional. The arguments to be passed to the method.
- **http_method** The HTTP method to use on the request. Defaults to POST.
- **os_ironic_api_version** String version (e.g. 1.35) to use for the request. If not specified, the clients default is used.
- **global_request_id** String containing global request ID header value (in form req-<UUID>) to use for the request.

ironicclient.v1.events module

class ironicclient.v1.events.**Event**(*manager*, *info*, *loaded=False*)

Bases: *ironicclient.common.base.Resource*

class ironicclient.v1.events.**EventManager**(*api*)

Bases: *ironicclient.common.base.CreateManager*

resource_class

alias of *ironicclient.v1.events.Event*

ironicclient.v1.node module

class ironicclient.v1.node.**Node**(*manager*, *info*, *loaded=False*)

Bases: *ironicclient.common.base.Resource*

class ironicclient.v1.node.**NodeManager**(*api*)

Bases: *ironicclient.common.base.CreateManager*

add_trait(*node_ident*, *trait*, *os_ironic_api_version=None*, *global_request_id=None*)

Add a trait to a node.

Parameters

- `node_ident` node UUID or name.
- `trait` trait to add to the node.
- `os_ironic_api_version` String version (e.g. 1.35) to use for the request. If not specified, the clients default is used.
- `global_request_id` String containing global request ID header value (in form req-<UUID>) to use for the request.

`delete`(*node_id*, *os_ironic_api_version=None*, *global_request_id=None*)

`get`(*node_id*, *fields=None*, *os_ironic_api_version=None*, *global_request_id=None*)

`get_bios_setting`(*node_ident*, *name*, *os_ironic_api_version=None*, *global_request_id=None*)

Get a BIOS setting from a node.

Parameters

- `node_ident` node UUID or name.
- `name` BIOS setting name to get from the node.
- `os_ironic_api_version` String version (e.g. 1.35) to use for the request. If not specified, the clients default is used.
- `global_request_id` String containing global request ID header value (in form req-<UUID>) to use for the request.

`get_boot_device`(*node_uuid*, *os_ironic_api_version=None*, *global_request_id=None*)

`get_by_instance_uuid`(*instance_uuid*, *fields=None*, *os_ironic_api_version=None*, *global_request_id=None*)

`get_console`(*node_uuid*, *os_ironic_api_version=None*, *global_request_id=None*)

`get_history_event`(*node_ident*, *event*, *os_ironic_api_version=None*, *global_request_id=None*)

Get a single event record for a node.

Provides the ability to request, and return a nodes single vent hisotyr entry.

Parameters

- `node_ident` The name or UUID of the node.
- `event` The UUID of the event entry as listed in the node event history list.
- `os_ironic_api_version` String version (e.g. 1.35) to use for the request. If not specified, the clients default is used.
- `global_request_id` String containing global request ID header value (in form req-<UUID>) to use for the request.

`get_history_list`(*node_ident*, *detail=False*, *os_ironic_api_version=None*, *global_request_id=None*)

Get node history event list.

Provides the ability to query a node event history list from the API and return the API response to the caller.

Requires API version 1.78.

Parameters

- `node_ident` The name or UUID of the node.
- `detail` If detailed data should be returned in the event list entry. Default False.
- `os_ironic_api_version` String version (e.g. 1.35) to use for the request. If not specified, the clients default is used.
- `global_request_id` String containing global request ID header value (in form req-<UUID>) to use for the request.

`get_supported_boot_devices`(*node_uuid*, *os_ironic_api_version=None*, *global_request_id=None*)

`get_traits`(*node_ident*, *os_ironic_api_version=None*, *global_request_id=None*)

Get traits for a node.

Parameters

- `node_ident` node UUID or name.
- `os_ironic_api_version` String version (e.g. 1.35) to use for the request. If not specified, the clients default is used.
- `global_request_id` String containing global request ID header value (in form req-<UUID>) to use for the request.

`get_vendor_passthru_methods`(*node_ident*, *os_ironic_api_version=None*, *global_request_id=None*)

`inject_nmi`(*node_uuid*, *os_ironic_api_version=None*, *global_request_id=None*)

`list`(*associated=None*, *maintenance=None*, *marker=None*, *limit=None*, *detail=False*, *sort_key=None*, *sort_dir=None*, *fields=None*, *provision_state=None*, *driver=None*, *resource_class=None*, *chassis=None*, *fault=None*, *os_ironic_api_version=None*, *conductor_group=None*, *conductor=None*, *owner=None*, *retired=None*, *lessee=None*, *global_request_id=None*)

Retrieve a list of nodes.

Parameters

- `associated` Optional. Either a Boolean or a string representation of a Boolean that indicates whether to return a list of associated (True or True) or unassociated (False or False) nodes.
- `maintenance` Optional. Either a Boolean or a string representation of a Boolean that indicates whether to return nodes in maintenance mode (True or True), or not in maintenance mode (False or False).

- `retired` Optional. Either a Boolean or a string representation of a Boolean that indicates whether to return retired nodes (True or True).
- `provision_state` Optional. String value to get only nodes in that provision state.
- `marker` Optional, the UUID of a node, eg the last node from a previous result set. Return the next result set.
- `limit`

  The maximum number of results to return per request, if:

  1) limit > 0, the maximum number of nodes to return.
  2) limit == 0, return the entire list of nodes.
  3) limit param is NOT specified (None), the number of items returned respect the maximum imposed by the Ironic API (see Ironics api.max_limit option).
- `detail` Optional, boolean whether to return detailed information about nodes.
- `sort_key` Optional, field used for sorting.
- `sort_dir` Optional, direction of sorting, either asc (the default) or desc.
- `fields` Optional, a list with a specified set of fields of the resource to be returned. Can not be used when detail is set.
- `driver` Optional. String value to get only nodes using that driver.
- `resource_class` Optional. String value to get only nodes with the given resource class set.
- `chassis` Optional, the UUID of a chassis. Used to get only nodes of this chassis.
- `fault` Optional. String value to get only nodes with specified fault.
- `os_ironic_api_version` String version (e.g. 1.35) to use for the request. If not specified, the clients default is used.
- `global_request_id` String containing global request ID header value (in form req-<UUID>) to use for the request.
- `conductor_group` Optional. String value to get only nodes with the given conductor group set.
- `conductor` Optional. String value to get only nodes mapped to the given conductor.
- `owner` Optional. String value to get only nodes mapped to a specific owner.
- `lessee` Optional. String value to get only nodes mapped to a specific lessee.

Returns A list of nodes.

`list_bios_settings`(*node_ident*, *detail=False*, *fields=None*, *os_ironic_api_version=None*, *global_request_id=None*)

List all BIOS settings from a node.

Parameters

- `node_ident` node UUID or name.
- `os_ironic_api_version` String version (e.g. 1.35) to use for the request. If not specified, the clients default is used.
- `global_request_id` String containing global request ID header value (in form req-<UUID>) to use for the request.
- `detail` Optional, boolean whether to return detailed information about bios settings.
- `fields` Optional, a list with a specified set of fields of the resource to be returned. Can not be used when detail is set.

`list_ports`(*node_id*, *marker=None*, *limit=None*, *sort_key=None*, *sort_dir=None*, *detail=False*, *fields=None*, *os_ironic_api_version=None*, *global_request_id=None*)

List all the ports for a given node.

Parameters

- `node_id` Name or UUID of the node.
- `marker` Optional, the UUID of a port, eg the last port from a previous result set. Return the next result set.
- `limit`

  The maximum number of results to return per request, if:

  1) limit > 0, the maximum number of ports to return.
  2) limit == 0, return the entire list of ports.
  3) limit param is NOT specified (None), the number of items returned respect the maximum imposed by the Ironic API (see Ironics api.max_limit option).

- `sort_key` Optional, field used for sorting.
- `sort_dir` Optional, direction of sorting, either asc (the default) or desc.
- `detail` Optional, boolean whether to return detailed information about ports.
- `fields` Optional, a list with a specified set of fields of the resource to be returned. Can not be used when detail is set.
- `os_ironic_api_version` String version (e.g. 1.35) to use for the request. If not specified, the clients default is used.
- `global_request_id` String containing global request ID header value (in form req-<UUID>) to use for the request.

Returns A list of ports.

`list_volume_connectors`(*node_id*, *marker=None*, *limit=None*, *sort_key=None*, *sort_dir=None*, *detail=False*, *fields=None*, *os_ironic_api_version=None*, *global_request_id=None*)

List all the volume connectors for a given node.

Parameters

- `node_id` Name or UUID of the node.
- `marker` Optional, the UUID of a volume connector, eg the last volume connector from a previous result set. Return the next result set.
- `limit`

  The maximum number of results to return per request, if:

  1) limit > 0, the maximum number of volume connectors to return.
  2) limit == 0, return the entire list of volume connectors.
  3) limit param is NOT specified (None), the number of items returned respect the maximum imposed by the Ironic API (see Ironics api.max_limit option).

- `sort_key` Optional, field used for sorting.
- `sort_dir` Optional, direction of sorting, either asc (the default) or desc.
- `detail` Optional, boolean whether to return detailed information about volume connectors.
- `fields` Optional, a list with a specified set of fields of the resource to be returned. Can not be used when detail is set.
- `os_ironic_api_version` String version (e.g. 1.35) to use for the request. If not specified, the clients default is used.
- `global_request_id` String containing global request ID header value (in form req-<UUID>) to use for the request.

Returns A list of volume connectors.

`list_volume_targets`(*node_id*, *marker=None*, *limit=None*, *sort_key=None*, *sort_dir=None*, *detail=False*, *fields=None*, *os_ironic_api_version=None*, *global_request_id=None*)

List all the volume targets for a given node.

Parameters

- `node_id` Name or UUID of the node.
- `marker` Optional, the UUID of a volume target, eg the last volume target from a previous result set. Return the next result set.
- `limit`

  The maximum number of results to return per request, if:

  1) limit > 0, the maximum number of volume targets to return.
  2) limit == 0, return the entire list of volume targets.

3) limit param is NOT specified (None), the number of items returned respect the maximum imposed by the Ironic API (see Ironics api.max_limit option).

- **sort_key** Optional, field used for sorting.
- **sort_dir** Optional, direction of sorting, either asc (the default) or desc.
- **detail** Optional, boolean whether to return detailed information about volume targets.
- **fields** Optional, a list with a specified set of fields of the resource to be returned. Can not be used when detail is set.
- **os_ironic_api_version** String version (e.g. 1.35) to use for the request. If not specified, the clients default is used.
- **global_request_id** String containing global request ID header value (in form req-<UUID>) to use for the request.

Returns A list of volume targets.

remove_all_traits(*node_ident*, *os_ironic_api_version=None*, *global_request_id=None*)

Remove all traits from a node.

Parameters

- **node_ident** node UUID or name.
- **os_ironic_api_version** String version (e.g. 1.35) to use for the request. If not specified, the clients default is used.
- **global_request_id** String containing global request ID header value (in form req-<UUID>) to use for the request.

remove_trait(*node_ident*, *trait*, *os_ironic_api_version=None*, *global_request_id=None*)

Remove a trait from a node.

Parameters

- **node_ident** node UUID or name.
- **trait** trait to remove from the node.
- **os_ironic_api_version** String version (e.g. 1.35) to use for the request. If not specified, the clients default is used.
- **global_request_id** String containing global request ID header value (in form req-<UUID>) to use for the request.

resource_class

alias of `ironicclient.v1.node.Node`

set_boot_device(*node_uuid*, *boot_device*, *persistent=False*, *os_ironic_api_version=None*, *global_request_id=None*)

set_boot_mode(*node_id*, *state*, *os_ironic_api_version=None*, *global_request_id=None*)

Sets boot mode for a node.

Parameters

- `node_id` Node identifier
- `state` One of target boot modes, uefi or bios
- `os_ironic_api_version` String version (e.g. 1.76) to use for the request. If not specified, the clients default is used.
- `global_request_id` String containing global request ID header value (in form req-<UUID>) to use for the request.

Raises ValueError if boot mode is not one of uefi / bios

Returns The status of the request

`set_console_mode`(*node_uuid*, *enabled*, *os_ironic_api_version=None*, *global_request_id=None*)

Set the console mode for the node.

Parameters

- `node_uuid` The UUID of the node.
- `enabled` Either a Boolean or a string representation of a Boolean. True to enable the console; False to disable.
- `os_ironic_api_version` String version (e.g. 1.35) to use for the request. If not specified, the clients default is used.
- `global_request_id` String containing global request ID header value (in form req-<UUID>) to use for the request.

`set_maintenance`(*node_id*, *state*, *maint_reason=None*, *os_ironic_api_version=None*, *global_request_id=None*)

Set the maintenance mode for the node.

Parameters

- `node_id` The UUID of the node.
- `state` the maintenance mode; either a Boolean or a string representation of a Boolean (eg, true, on, false, off). True to put the node in maintenance mode; False to take the node out of maintenance mode.
- `maint_reason` Optional string. Reason for putting node into maintenance mode.
- `os_ironic_api_version` String version (e.g. 1.35) to use for the request. If not specified, the clients default is used.
- `global_request_id` String containing global request ID header value (in form req-<UUID>) to use for the request.

Raises InvalidAttribute if state is an invalid string (that doesnt represent a Boolean).

`set_power_state`(*node_id*, *state*, *soft=False*, *timeout=None*, *os_ironic_api_version=None*, *global_request_id=None*)

Sets power state for a node.

Parameters

- `node_id` Node identifier
- `state` One of target power state, on, off, or reboot
- `soft` The flag for graceful power off or reboot
- `timeout` The timeout (in seconds) positive integer value (> 0)
- `os_ironic_api_version` String version (e.g. 1.35) to use for the request. If not specified, the clients default is used.
- `global_request_id` String containing global request ID header value (in form req-<UUID>) to use for the request.

Raises ValueError if soft or timeout option is invalid

Returns The status of the request

`set_provision_state`(*node_uuid*, *state*, *configdrive=None*, *cleansteps=None*, *rescue_password=None*, *os_ironic_api_version=None*, *global_request_id=None*, *deploysteps=None*)

Set the provision state for the node.

Parameters

- `node_uuid` The UUID or name of the node.
- `state` The desired provision state. One of active, deleted, rebuild, inspect, provide, manage, clean, abort, rescue, unrescue.
- `configdrive` A gzipped, base64-encoded configuration drive string OR the path to the configuration drive file OR the path to a directory containing the config drive files OR a dictionary to build config drive from. In case its a directory, a config drive will be generated from it. In case its a dictionary, a config drive will be generated on the server side (requires API version 1.56). This is only valid when setting state to active.
- `cleansteps` The clean steps as a list of clean-step dictionaries; each dictionary should have keys interface and step, and optional key args. This must be specified (and is only valid) when setting provision-state to clean.
- `rescue_password` A string to be used as the login password inside the rescue ramdisk once a node is rescued. This must be specified (and is only valid) when setting state to rescue.
- `os_ironic_api_version` String version (e.g. 1.35) to use for the request. If not specified, the clients default is used.
- `global_request_id` String containing global request ID header value (in form req-<UUID>) to use for the request.
- `deploysteps` The deploy steps as a list of deploy-step dictionaries; each dictionary should have keys interface, step, priority, and optional key args. This is optional and is only valid when setting provision-state to active or rebuild.

Raises InvalidAttribute if there was an error with the clean steps or deploy steps

Returns The status of the request

`set_secure_boot`(*node_id*, *state*, *os_ironic_api_version=None*, *global_request_id=None*)

Set the secure boot state for the node.

Parameters

- `node_id` The UUID of the node.
- `state` the secure boot state; either a Boolean or a string representation of a Boolean (eg, true, on, false, off). True to turn secure boot on; False to turn secure boot off.
- `os_ironic_api_version` String version (e.g. 1.76) to use for the request. If not specified, the clients default is used.
- `global_request_id` String containing global request ID header value (in form req-<UUID>) to use for the request.

Raises InvalidAttribute if state is an invalid string (that doesnt represent a Boolean).

`set_target_raid_config`(*node_ident*, *target_raid_config*, *os_ironic_api_version=None*, *global_request_id=None*)

Sets target_raid_config for a node.

Parameters

- `node_ident` Node identifier
- `target_raid_config` A dictionary with the target RAID configuration; may be empty.
- `os_ironic_api_version` String version (e.g. 1.35) to use for the request. If not specified, the clients default is used.
- `global_request_id` String containing global request ID header value (in form req-<UUID>) to use for the request.

Returns status of the request

`set_traits`(*node_ident*, *traits*, *os_ironic_api_version=None*, *global_request_id=None*)

Set traits for a node.

Removes any existing traits and adds the traits passed in to this method.

Parameters

- `node_ident` node UUID or name.
- `traits` list of traits to add to the node.
- `os_ironic_api_version` String version (e.g. 1.35) to use for the request. If not specified, the clients default is used.
- `global_request_id` String containing global request ID header value (in form req-<UUID>) to use for the request.

`states`(*node_uuid*, *os_ironic_api_version=None*, *global_request_id=None*)

`update`(*node_id*, *patch*, *http_method='PATCH'*, *os_ironic_api_version=None*, *reset_interfaces=None*, *global_request_id=None*) `validate`(*node_uuid*, *os_ironic_api_version=None*, *global_request_id=None*)

`vendor_passthru`(*node_id*, *method*, *args=None*, *http_method=None*, *os_ironic_api_version=None*, *global_request_id=None*)

Issue requests for vendor-specific actions on a given node.

Parameters

- `node_id` The UUID of the node.
- `method` Name of the vendor method.
- `args` Optional. The arguments to be passed to the method.
- `http_method` The HTTP method to use on the request. Defaults to POST.
- `os_ironic_api_version` String version (e.g. 1.35) to use for the request. If not specified, the clients default is used.
- `global_request_id` String containing global request ID header value (in form req-<UUID>) to use for the request.

`vif_attach`(*node_ident*, *vif_id*, *os_ironic_api_version=None*, *global_request_id=None*, ***kwargs*)

Attach VIF to a given node.

Parameters

- `node_ident` The UUID or Name of the node.
- `vif_id` The UUID or Name of the VIF to attach.
- `os_ironic_api_version` String version (e.g. 1.35) to use for the request. If not specified, the clients default is used.
- `global_request_id` String containing global request ID header value (in form req-<UUID>) to use for the request.
- `kwargs` A dictionary containing the attributes of the resource that will be created.

`vif_detach`(*node_ident*, *vif_id*, *os_ironic_api_version=None*, *global_request_id=None*)

Detach VIF from a given node.

Parameters

- `node_ident` The UUID or Name of the node.
- `vif_id` The UUID or Name of the VIF to detach.
- `os_ironic_api_version` String version (e.g. 1.35) to use for the request. If not specified, the clients default is used.
- `global_request_id` String containing global request ID header value (in form req-<UUID>) to use for the request.

`vif_list`(*node_ident*, *os_ironic_api_version=None*, *global_request_id=None*)

List VIFs attached to a given node.

Parameters

- `node_ident` The UUID or Name of the node.

- **os_ironic_api_version** String version (e.g. 1.35) to use for the request. If not specified, the clients default is used.
- **global_request_id** String containing global request ID header value (in form req-<UUID>) to use for the request.

wait_for_provision_state(*node_ident*, *expected_state*, *timeout=0*, *poll_interval=2*, *poll_delay_function=None*, *fail_on_unexpected_state=True*, *os_ironic_api_version=None*, *global_request_id=None*)

Helper function to wait for a node to reach a given state.

Polls Ironic API in a loop until node gets to a requested state.

Fails in the following cases: * Timeout (if provided) is reached * Nodes last_error gets set to a non-empty value * Unexpected stable state is reached and fail_on_unexpected_state is on * Error state is reached (if its not equal to expected_state)

Parameters

- **node_ident** node UUID or name
- **expected_state** expected final provision state
- **timeout** timeout in seconds, no timeout if 0
- **poll_interval** interval in seconds between 2 poll
- **poll_delay_function** function to use to wait between polls (defaults to time.sleep). Should take one argument - delay time in seconds. Any exceptions raised inside it will abort the wait.
- **fail_on_unexpected_state** whether to fail if the nodes reaches a different stable state.
- **os_ironic_api_version** String version (e.g. 1.35) to use for the request. If not specified, the clients default is used.
- **global_request_id** String containing global request ID header value (in form req-<UUID>) to use for the request.

Raises StateTransitionFailed if node reached an error state

Raises StateTransitionTimeout on timeout

ironicclient.v1.port module

class ironicclient.v1.port.**Port**(*manager*, *info*, *loaded=False*)

Bases: *ironicclient.common.base.Resource*

class ironicclient.v1.port.**PortManager**(*api*)

Bases: *ironicclient.common.base.CreateManager*

delete(*port_id*, *os_ironic_api_version=None*, *global_request_id=None*)

get(*port_id*, *fields=None*, *os_ironic_api_version=None*, *global_request_id=None*)

get_by_address(*address*, *fields=None*, *os_ironic_api_version=None*, *global_request_id=None*) `list`(*address=None*, *limit=None*, *marker=None*, *sort_key=None*, *sort_dir=None*, *detail=False*, *fields=None*, *node=None*, *portgroup=None*, *os_ironic_api_version=None*, *global_request_id=None*)

Retrieve a list of ports.

Parameters

- **`address`** Optional, MAC address of a port, to get the port which has this MAC address
- **`marker`** Optional, the UUID of a port, eg the last port from a previous result set. Return the next result set.
- **`limit`**

  The maximum number of results to return per request, if:

  1) limit > 0, the maximum number of ports to return.
  2) limit == 0, return the entire list of ports.
  3) limit param is NOT specified (None), the number of items returned respect the maximum imposed by the Ironic API (see Ironics api.max_limit option).

- **`sort_key`** Optional, field used for sorting.
- **`sort_dir`** Optional, direction of sorting, either asc (the default) or desc.
- **`detail`** Optional, boolean whether to return detailed information about ports.
- **`fields`** Optional, a list with a specified set of fields of the resource to be returned. Can not be used when detail is set.
- **`node`** Optional, name or UUID of a node. Used to get ports of this node.
- **`portgroup`** Optional, name or UUID of a portgroup. Used to get ports of this portgroup.
- **`os_ironic_api_version`** String version (e.g. 1.35) to use for the request. If not specified, the clients default is used.
- **`global_request_id`** String containing global request ID header value (in form req-<UUID>) to use for the request.

Returns A list of ports.

`resource_class`

alias of *`ironicclient.v1.port.Port`*

`update`(*port_id*, *patch*, *os_ironic_api_version=None*, *global_request_id=None*)

ironicclient.v1.portgroup module

class ironicclient.v1.portgroup.**Portgroup**(*manager*, *info*, *loaded=False*)

Bases: *ironicclient.common.base.Resource*

class ironicclient.v1.portgroup.**PortgroupManager**(*api*)

Bases: *ironicclient.common.base.CreateManager*

delete(*portgroup_id*, *os_ironic_api_version=None*, *global_request_id=None*)

Delete the Portgroup from the DB.

Parameters

- **portgroup_id** The UUID or name of a portgroup.
- **os_ironic_api_version** String version (e.g. 1.35) to use for the request. If not specified, the clients default is used.
- **global_request_id** String containing global request ID header value (in form req-<UUID>) to use for the request.

get(*portgroup_id*, *fields=None*, *os_ironic_api_version=None*, *global_request_id=None*)

Get a port group with the specified identifier.

Parameters

- **portgroup_id** The UUID or name of a portgroup.
- **fields** Optional, a list with a specified set of fields of the resource to be returned. Can not be used when detail is set.
- **os_ironic_api_version** String version (e.g. 1.35) to use for the request. If not specified, the clients default is used.
- **global_request_id** String containing global request ID header value (in form req-<UUID>) to use for the request.

Returns a *Portgroup* object.

get_by_address(*address*, *fields=None*, *os_ironic_api_version=None*, *global_request_id=None*)

Get a port group with the specified MAC address.

Parameters

- **address** The MAC address of a portgroup.
- **fields** Optional, a list with a specified set of fields of the resource to be returned. Can not be used when detail is set.
- **os_ironic_api_version** String version (e.g. 1.35) to use for the request. If not specified, the clients default is used.
- **global_request_id** String containing global request ID header value (in form req-<UUID>) to use for the request.

Returns a *Portgroup* object.

`list`(*node=None*, *address=None*, *limit=None*, *marker=None*, *sort_key=None*, *sort_dir=None*, *detail=False*, *fields=None*, *os_ironic_api_version=None*, *global_request_id=None*)

Retrieve a list of portgroups.

Parameters

- `node` Optional, UUID or name of a node, to get the portgroups for that node.
- `address` Optional, MAC address of a portgroup, to get the portgroup which has this MAC address.
- `marker` Optional, the UUID of a portgroup, eg the last portgroup from a previous result set. Return the next result set.
- `limit`

  The maximum number of results to return per request, if:

  1) limit > 0, the maximum number of portgroups to return.
  2) limit == 0, return the entire list of portgroups.
  3) limit == None, the number of items returned respect the maximum imposed by the Ironic API (see Ironics api.max_limit option).

- `sort_key` Optional, field used for sorting.
- `sort_dir` Optional, direction of sorting, either asc (the default) or desc.
- `detail` Optional, boolean whether to return detailed information about portgroups.
- `fields` Optional, a list with a specified set of fields of the resource to be returned. Can not be used when detail is set.
- `os_ironic_api_version` String version (e.g. 1.35) to use for the request. If not specified, the clients default is used.
- `global_request_id` String containing global request ID header value (in form req-<UUID>) to use for the request.

Returns A list of portgroups.

Raises InvalidAttribute if a subset of fields is requested with detail option set.

`list_ports`(*portgroup_id*, *marker=None*, *limit=None*, *sort_key=None*, *sort_dir=None*, *detail=False*, *fields=None*, *os_ironic_api_version=None*, *global_request_id=None*)

List all the ports for a given portgroup.

Parameters

- `portgroup_id` Name or UUID of the portgroup.
- `marker` Optional, the UUID of a port, eg the last port from a previous result set. Return the next result set.
- `limit`

  The maximum number of results to return per request, if:

1) limit > 0, the maximum number of ports to return.
2) limit == 0, return the entire list of ports.
3) limit param is NOT specified (None), the number of items returned respect the maximum imposed by the Ironic API (see Ironics api.max_limit option).

- **sort_key** Optional, field used for sorting.
- **sort_dir** Optional, direction of sorting, either asc (the default) or desc.
- **detail** Optional, boolean whether to return detailed information about ports.
- **fields** Optional, a list with a specified set of fields of the resource to be returned. Can not be used when detail is set.
- **os_ironic_api_version** String version (e.g. 1.35) to use for the request. If not specified, the clients default is used.
- **global_request_id** String containing global request ID header value (in form req-<UUID>) to use for the request.

Returns A list of ports.

`resource_class`

alias of *ironicclient.v1.portgroup.Portgroup*

`update`(*portgroup_id*, *patch*, *os_ironic_api_version=None*, *global_request_id=None*)

Update the Portgroup.

Parameters

- **portgroup_id** The UUID or name of a portgroup.
- **patch** The patch request with updates.
- **os_ironic_api_version** String version (e.g. 1.35) to use for the request. If not specified, the clients default is used.
- **global_request_id** String containing global request ID header value (in form req-<UUID>) to use for the request.

ironicclient.v1.resource_fields module

`class ironicclient.v1.resource_fields.Resource`(*field_ids*, *sort_excluded=None*, *override_labels=None*)

Bases: `object`

Resource class

This class is used to manage the various fields that a resource (e.g. Chassis, Node, Port) contains. An individual field consists of a field_id (key) and a label (value). The caller only provides the field_ids when instantiating the object.

Ordering of the field_ids will be preserved as specified by the caller.

It also provides the ability to exclude some of these fields when they are being used for sorting.

```
FIELDS = {'address': 'Address', 'alive': 'Alive', 'allocation_uuid':
'Allocation UUID', 'allowable_values': 'Allowable Values', 'async':
'Async', 'attach': 'Response is attachment', 'attribute_type':
'Attribute Type', 'automated_clean': 'Automated Clean', 'bios_interface':
'BIOS Interface', 'boot_index': 'Boot Index', 'boot_interface': 'Boot
Interface', 'boot_mode': 'Boot Mode', 'candidate_nodes': 'Candidate
Nodes', 'chassis_uuid': 'Chassis UUID', 'clean_step': 'Clean Step',
'conductor': 'Conductor', 'conductor_group': 'Conductor Group',
'connector_id': 'Connector ID', 'console_enabled': 'Console Enabled',
'console_interface': 'Console Interface', 'created_at': 'Created At',
'default_bios_interface': 'Default BIOS Interface',
'default_boot_interface': 'Default Boot Interface',
'default_console_interface': 'Default Console Interface',
'default_deploy_interface': 'Default Deploy Interface',
'default_inspect_interface': 'Default Inspect Interface',
'default_management_interface': 'Default Management Interface',
'default_network_interface': 'Default Network Interface',
'default_power_interface': 'Default Power Interface',
'default_raid_interface': 'Default RAID Interface',
'default_rescue_interface': 'Default Rescue Interface',
'default_storage_interface': 'Default Storage Interface',
'default_vendor_interface': 'Default Vendor Interface',
'deploy_interface': 'Deploy Interface', 'deploy_step': 'Deploy Step',
'description': 'Description', 'driver': 'Driver', 'driver_info':
'Driver Info', 'driver_internal_info': 'Driver Internal Info', 'drivers':
'Drivers', 'enabled_bios_interfaces': 'Enabled BIOS Interfaces',
'enabled_boot_interfaces': 'Enabled Boot Interfaces',
'enabled_console_interfaces': 'Enabled Console Interfaces',
'enabled_deploy_interfaces': 'Enabled Deploy Interfaces',
'enabled_inspect_interfaces': 'Enabled Inspect Interfaces',
'enabled_management_interfaces': 'Enabled Management Interfaces',
'enabled_network_interfaces': 'Enabled Network Interfaces',
'enabled_power_interfaces': 'Enabled Power Interfaces',
'enabled_raid_interfaces': 'Enabled RAID Interfaces',
'enabled_rescue_interfaces': 'Enabled Rescue Interfaces',
'enabled_storage_interfaces': 'Enabled Storage Interfaces',
'enabled_vendor_interfaces': 'Enabled Vendor Interfaces', 'event':
'Description of the event', 'event_type': 'Event Origin Type', 'extra':
'Extra', 'fault': 'Fault', 'hostname': 'Hostname', 'hosts': 'Active
host(s)', 'http_methods': 'Supported HTTP methods', 'id': 'ID',
'inspect_interface': 'Inspect Interface', 'inspection_finished_at':
'Inspection Finished At', 'inspection_started_at': 'Inspection Started
At', 'instance_info': 'Instance Info', 'instance_uuid': 'Instance UUID',
'internal_info': 'Internal Info', 'is_smartnic': 'Is Smart NIC port',
'last_error': 'Last Error', 'lessee': 'Lessee', 'local_link_connection':
'Local Link Connection', 'lower_bound': 'Lower Bound', 'maintenance':
'Maintenance', 'maintenance_reason': 'Maintenance Reason',
'management_interface': 'Management Interface', 'max_length': 'Maximum
Length', 'min_length': 'Minimum Length', 'mode': 'Mode', 'name':
'Name', 'network_data': 'Network Configuration', 'network_interface':
'Network Interface', 'node_uuid': 'Node UUID', 'owner': 'Owner',
'physical_network': 'Physical Network', 'portgroup_uuid': 'Portgroup
UUID', 'power_interface': 'Power Interface', 'power_state': 'Power
State', 'properties': 'Properties', 'protected': 'Protected',
'protected_reason': 'Protected Reason', 'provision_state': 'Provisioning
State', 'provision_updated_at': 'Provision Updated At', 'pxe_enabled':
'PXE boot enabled', 'raid_config': 'Current RAID configuration',
```

property fields

property labels

property sort_fields

property sort_labels

ironicclient.v1.utils module

ironicclient.v1.volume_connector module

class ironicclient.v1.volume_connector.**VolumeConnector**(*manager*, *info*, *loaded=False*)

Bases: *ironicclient.common.base.Resource*

class ironicclient.v1.volume_connector.**VolumeConnectorManager**(*api*)

Bases: *ironicclient.common.base.CreateManager*

delete(*volume_connector_id*, *os_ironic_api_version=None*, *global_request_id=None*)

get(*volume_connector_id*, *fields=None*, *os_ironic_api_version=None*, *global_request_id=None*)

list(*node=None*, *limit=None*, *marker=None*, *sort_key=None*, *sort_dir=None*, *detail=False*, *fields=None*, *os_ironic_api_version=None*, *global_request_id=None*)

Retrieve a list of volume connector.

Parameters

- **node** Optional, UUID or name of a node, to get volume connectors for this node only.
- **marker** Optional, the UUID of a volume connector, eg the last volume connector from a previous result set. Return the next result set.
- **limit**

  The maximum number of results to return per request, if:

  1) limit > 0, the maximum number of volume connectors to return.
  2) limit == 0, return the entire list of volume connectors.
  3) limit param is NOT specified (None), the number of items returned respect the maximum imposed by the Ironic API (see Ironics api.max_limit option).

- **sort_key** Optional, field used for sorting.
- **sort_dir** Optional, direction of sorting, either asc (the default) or desc.
- **detail** Optional, boolean whether to return detailed information about volume connectors.

- **fields** Optional, a list with a specified set of fields of the resource to be returned. Can not be used when detail is set.
- **os_ironic_api_version** String version (e.g. 1.35) to use for the request. If not specified, the clients default is used.
- **global_request_id** String containing global request ID header value (in form req-<UUID>) to use for the request.

Returns A list of volume connectors.

resource_class

alias of *ironicclient.v1.volume_connector.VolumeConnector*

update(*volume_connector_id*, *patch*, *os_ironic_api_version=None*, *global_request_id=None*)

ironicclient.v1.volume_target module

class ironicclient.v1.volume_target.**VolumeTarget**(*manager*, *info*, *loaded=False*)

Bases: *ironicclient.common.base.Resource*

class ironicclient.v1.volume_target.**VolumeTargetManager**(*api*)

Bases: *ironicclient.common.base.CreateManager*

delete(*volume_target_id*, *os_ironic_api_version=None*, *global_request_id=None*)

get(*volume_target_id*, *fields=None*, *os_ironic_api_version=None*, *global_request_id=None*)

list(*node=None*, *limit=None*, *marker=None*, *sort_key=None*, *sort_dir=None*, *detail=False*, *fields=None*, *os_ironic_api_version=None*, *global_request_id=None*)

Retrieve a list of volume target.

Parameters

- **node** Optional, UUID or name of a node, to get volume targets for this node only.
- **marker** Optional, the UUID of a volume target, eg the last volume target from a previous result set. Return the next result set.
- **limit**

  The maximum number of results to return per request, if:

  1) limit > 0, the maximum number of volume targets to return.
  2) limit == 0, return the entire list of volume targets.
  3) limit param is NOT specified (None), the number of items returned respect the maximum imposed by the Ironic API (see Ironics api.max_limit option).

- **sort_key** Optional, field used for sorting.
- **sort_dir** Optional, direction of sorting, either asc (the default) or desc.
- **detail** Optional, boolean whether to return detailed information about volume targets.

- **fields** Optional, a list with a specified set of fields of the resource to be returned. Can not be used when detail is set.
- **os_ironic_api_version** String version (e.g. 1.35) to use for the request. If not specified, the clients default is used.
- **global_request_id** String containing global request ID header value (in form req-<UUID>) to use for the request.

Returns A list of volume targets.

resource_class

alias of *ironicclient.v1.volume_target.VolumeTarget*

update(*volume_target_id*, *patch*, *os_ironic_api_version=None*, *global_request_id=None*)

Module contents

Submodules

ironicclient.client module

ironicclient.client.**Client**(*version*, *endpoint_override=None*, *session=None*, **args*, ***kwargs*)

Create a client of an appropriate version.

This call requires a session. If you want it to be created, use `get_client` instead.

Parameters

- **endpoint_override** A bare metal endpoint to use.
- **session** A keystoneauth session to use. This argument is actually required and is marked optional only for backward compatibility.
- **args** Other arguments to pass to the HTTP client. Not recommended, use kwargs instead.
- **kwargs** Other keyword arguments to pass to the HTTP client (e.g. `insecure`).

ironicclient.client.**get_client**(*api_version*, *auth_type=None*, *os_ironic_api_version=None*, *max_retries=None*, *retry_interval=None*, *session=None*, *valid_interfaces=None*, *interface=None*, *service_type=None*, *region_name=None*, *additional_headers=None*, *global_request_id=None*, ***kwargs*)

Get an authenticated client, based on the credentials.

Parameters

- **api_version** the API version to use. Valid value: 1.
- **auth_type** type of keystoneauth auth plugin loader to use.
- **os_ironic_api_version** ironic API version to use.
- **max_retries** Maximum number of retries in case of conflict error

- **retry_interval** Amount of time (in seconds) between retries in case of conflict error.
- **session** An existing keystoneauth session. Will be created from kwargs if not provided.
- **valid_interfaces** List of valid endpoint interfaces to use if the bare metal endpoint is not provided.
- **interface** An alias for valid_interfaces.
- **service_type** Bare metal endpoint service type.
- **region_name** Name of the region to use when searching the bare metal endpoint.
- **additional_headers** Additional headers that should be attached to every request passing through the client. Headers of the same name specified per request will take priority.
- **global_request_id** A header (in the form of `req-$uuid`) that will be passed on all requests. Enables cross project request id tracking.
- **kwargs** all the other params that are passed to keystoneauth for session construction.

ironicclient.exc module

`ironicclient.exc.`**`AmbigiousAuthSystem`**

alias of *`ironicclient.exc.AmbiguousAuthSystem`*

exception `ironicclient.exc.`**`AmbiguousAuthSystem`**

Bases: *`ironicclient.common.apiclient.exceptions.ClientException`*

Could not obtain token and endpoint using provided credentials.

exception `ironicclient.exc.`**`InvalidAttribute`**

Bases: *`ironicclient.common.apiclient.exceptions.ClientException`*

exception `ironicclient.exc.`**`StateTransitionFailed`**

Bases: *`ironicclient.common.apiclient.exceptions.ClientException`*

Failed to reach a requested provision state.

exception `ironicclient.exc.`**`StateTransitionTimeout`**

Bases: *`ironicclient.common.apiclient.exceptions.ClientException`*

Timed out while waiting for a requested provision state.

`ironicclient.exc.`**`from_response`**(*response*, *message=None*, *traceback=None*, *method=None*, *url=None*)

Return an HttpError instance based on response from httplib/requests.

ironicclient.shell module

class ironicclient.shell.**App**

Bases: `cliff.app.App`

build_option_parser(*description*, *version*, *argparse_kwargs=None*)

Return an argparse option parser for this application.

Subclasses may override this method to extend the parser with more global options.

Parameters

- **description** (*str*) full description of the application
- **version** (*str*) version number for the application
- **argparse_kwargs** extra keyword argument passed to the ArgumentParser constructor

initialize_app(*argv*)

Hook for subclasses to take global initialization action after the arguments are parsed but before a command is run. Invoked only once, even in interactive mode.

Parameters **argv** List of arguments, including the subcommand to run. Empty for interactive mode.

class ironicclient.shell.**ClientManager**(*cloud_region*, *options*)

Bases: `object`

property baremetal

property baremetal_introspection

class ironicclient.shell.**CommandManager**(*namespace: str*, *convert_underscores: bool = True*)

Bases: `cliff.commandmanager.CommandManager`

load_commands(*namespace*)

Load all the commands from an entrypoint

ironicclient.shell.**main**(*argv=['-W', '-b', 'latex', 'doc/source', 'doc/build/pdf']*)

Module contents

CHAPTER

TWO

INDICES AND TABLES

- genindex
- modindex
- search

PYTHON MODULE INDEX

INDEX