

---

# Python Ironic Client Documentation

*Release 5.1.1.dev3*

**OpenStack LLC**

**Mar 19, 2025**



# CONTENTS

<b>1</b>	<b>Contents</b>	<b>3</b>
1.1	ironicclient Python API . . . . .	3
1.2	python-ironicclient User Documentation . . . . .	5
1.3	Creating the Bare Metal service resources from file . . . . .	77
1.4	python-ironicclient Contributor Documentation . . . . .	80
1.5	Full Ironic Client Python API Reference . . . . .	82
<b>2</b>	<b>Indices and tables</b>	<b>175</b>
	<b>Python Module Index</b>	<b>177</b>
	<b>Index</b>	<b>179</b>



This is a client for the OpenStack [Ironic](#) API. It provides:

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



## 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 doesn't 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 be 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 passthru 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 be 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 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 you're 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 OpenStack 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,
                                                                    de-
                                                                    tails=None,
                                                                    re-
                                                                    sponse=None,
                                                                    re-
                                                                    quest_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,
                                                                    re-
                                                                    sponse=None,
                                                                    re-
                                                                    quest_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,
                                                                    de-
                                                                    tails=None,
                                                                    re-
                                                                    sponse=None,
                                                                    re-
                                                                    quest_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,
                                                                    de-
                                                                    tails=None,
                                                                    re-
                                                                    sponse=None,
                                                                    re-
                                                                    quest_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,
                                                                    re-
                                                                    quest_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,
                                                                    de-
                                                                    tails=None,
                                                                    re-
                                                                    sponse=None,
                                                                    re-
                                                                    quest_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,
                                                                    re-
                                                                    sponse=None,
                                                                    re-
                                                                    quest_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,
                                                                    re-
                                                                    sponse=None,
                                                                    re-
                                                                    quest_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,
                                                                    de-
                                                                    tails=None,
                                                                    re-
                                                                    sponse=None,
                                                                    re-
                                                                    quest_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,
                                                                    de-
                                                                    tails=None,
                                                                    re-
                                                                    sponse=None,
                                                                    re-
                                                                    quest_id=None,
                                                                    url=None,
                                                                    method=None,
                                                                    http_status=None)
```

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,
    de-
    tails=None,
    re-
    sponse=None,
    re-
    quest_id=None,
    url=None,
    method=None,
    http_status=None)
```

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,
    de-
    tails=None,
    re-
    sponse=None,
    re-
    quest_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,
                                                                           de-
                                                                           tails=None,
                                                                           re-
                                                                           sponse=None,
                                                                           re-
                                                                           quest_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,
                                                                           de-
                                                                           tails=None,
                                                                           re-
                                                                           sponse=None,
                                                                           re-
                                                                           quest_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:
                                                                    Op-
                                                                    tional[argparse.Namespace]
                                                                    cmd_name:
                                                                    Op-
                                                                    tional[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:
                                                                    Op-
                                                                    tional[argparse.Namespace]
                                                                    cmd_name:
                                                                    Op-
                                                                    tional[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:
                                                                    Op-
                                                                    tional[argparse.Namespace],
                                                                    cmd_name:
                                                                    Op-
                                                                    tional[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:
                                                                    Op-
                                                                    tional[argparse.Namespace],
                                                                    cmd_name:
                                                                    Op-
                                                                    tional[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:
                                                                    Op-
                                                                    tional[argparse.Namespace]
                                                                    cmd_name:
                                                                    Op-
                                                                    tional[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:
                                                                    Op-
                                                                    tional[argparse.Namespace]
                                                                    cmd_name:
                                                                    Op-
                                                                    tional[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: Op-
    tional[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: Op-
    tional[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: Op-
    tional[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:
                                                                    Op-
                                                                    tional[argparse.Namespace],
                                                                    cmd_name:
                                                                    Op-
                                                                    tional[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: Op-
                                                            tional[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:
    Op-
    tional[argpa
    cmd_name:
    Op-
    tional[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:
    Op-
    tional[argpa
    cmd_name:
    Op-
    tional[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:
                                                                    Op-
                                                                    tional[argparse.
                                                                    cmd_name:
                                                                    Op-
                                                                    tional[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:
                                                                    Op-
                                                                    tional[argparse.
                                                                    cmd_name:
                                                                    Op-
                                                                    tional[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:
                                                                    Op-
                                                                    tional[argpars
                                                                    cmd_name:
                                                                    Op-
                                                                    tional[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:
                                                                    Op-
                                                                    tional[argpar
                                                                    cmd_name:
                                                                    Op-
                                                                    tional[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: Op-
                                                             tional[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:
                                                                    Op-
                                                                    tional[argparse.Namespace]
                                                                    cmd_name:
                                                                    Op-
                                                                    tional[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:
                                                                    Op-
                                                                    tional[argparse.Name
                                                                    cmd_name:
                                                                    Op-
                                                                    tional[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:
                                                                    Op-
                                                                    tional[argparse.Namespace
                                                                    cmd_name:
                                                                    Op-
                                                                    tional[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:
                                                                    Op-
                                                                    tional[argparse.Namespace]
                                                                    cmd_name:
                                                                    Op-
                                                                    tional[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: Op-
                                                                    tional[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: Op-
                                                                    tional[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: Op-
                                                                tional[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: Op-
                                                            tional[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:
                                                                    Op-
                                                                    tional[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:
                                                                    Op-
                                                                    tional[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:
                                                                    Op-
                                                                    tional[argparse.Namespace],
                                                                    cmd_name:
                                                                    Op-
                                                                    tional[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: Op-
                                                                    tional[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:
    Op-
    tional[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: Op-
    tional[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: Op-
    tional[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: Op-
    tional[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: Op-
    tional[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: Op-
    tional[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: Op-
                                                             tional[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:
                                                                    Op-
                                                                    tional[argparse.Namespace],
                                                                    cmd_name:
                                                                    Op-
                                                                    tional[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: Op-
                                                            tional[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: Op-
                                                                    tional[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:
                                                                    Op-
                                                                    tional[argparse.Namespace],
                                                                    cmd_name:
                                                                    Op-
                                                                    tional[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:
                                                                    Op-
                                                                    tional[argparse.Namespace],
                                                                    cmd_name:
                                                                    Op-
                                                                    tional[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: Op-
                                                            tional[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: Op-
                                                            tional[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: Op-
                                                         tional[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: Op-
                                                                    tional[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.PassthruCallBaremetalNode
(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: Op-
                                                                    tional[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: Op-
                                                                    tional[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: Op-
                                                             tional[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:
                                                                    Op-
                                                                    tional[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: Op-
                                                                    tional[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:
                                                                    Op-
                                                                    tional[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: Op-
                                                                    tional[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: Op-
                                                                tional[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: Op-
                                                                tional[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: Op-
                                                                tional[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: Op-
                                                             tional[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: Op-
                                                             tional[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:
                                                                    Op-
                                                                    tional[argparse.Namespace]
                                                                    cmd_name:
                                                                    Op-
                                                                    tional[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:
                                                                    Op-
                                                                    tional[argparse.Namespace]
                                                                    cmd_name:
                                                                    Op-
                                                                    tional[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:
                                                                    Op-
                                                                    tional[argparse.Namespace],
                                                                    cmd_name:
                                                                    Op-
                                                                    tional[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:
                                                                    Op-
                                                                    tional[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:
                                                                    Op-
                                                                    tional[argparse.Namespace],
                                                                    cmd_name:
                                                                    Op-
                                                                    tional[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*:

*Op-*

*tional[argparse.Namespace*

*cmd\_name*:

*Op-*

*tional[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*.

*Op-*

*tional[arg*

*cmd\_name*

*Op-*

*tional[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:
    Op-
    tional[argp
    cmd_name:
    Op-
    tional[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:
    Op-
    tional[argpa
    cmd_name:
    Op-
    tional[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:
    Op-
    tional[argpar
    tional[str]
    cmd_name:
    Op-
    tional[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:
    Op-
    tional[argpa
    tional[str]
    cmd_name:
    Op-
    tional[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:
                                                                    Op-
                                                                    tional[argp
                                                                    cmd_name
                                                                    Op-
                                                                    tional[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_connector.UnsetBaremetalVolumeConnector (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:
                                                                    Op-
                                                                    tional[argparse.N
                                                                    cmd_name:
                                                                    Op-
                                                                    tional[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:
    Op-
    tional[argparse.N
    cmd_name:
    Op-
    tional[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:
    Op-
    tional[argparse.Nam
    cmd_name:
    Op-
    tional[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:
    Op-
    tional[argparse.Name-
    cmd_name:
    Op-
    tional[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:
    Op-
    tional[argparse.Name-
    cmd_name:
    Op-
    tional[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:
    Op-
    tional[argparse.Namespace],
    cmd_name:
    Op-
    tional[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_target.UnsetBaremetalVolumeTarget`  
(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 occurred.

**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 port-groups 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 hisoty 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

```

```

'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.AmbiguousAuthSystem`

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

## INDICES AND TABLES

- genindex
- modindex
- search



## PYTHON MODULE INDEX

### C

`ironicclient.client`, 172

`ironicclient.common`, 102

`ironicclient.common.apiclient`, 97

`ironicclient.common.apiclient.base`, 83

`ironicclient.common.apiclient.exceptions`,  
85

`ironicclient.common.base`, 97

`ironicclient.common.filecache`, 98

`ironicclient.common.http`, 98

`ironicclient.common.i18n`, 99

`ironicclient.common.utils`, 99

### e

`ironicclient.exc`, 173

### i

`ironicclient`, 174

### O

`ironicclient.osc`, 143

`ironicclient.osc.plugin`, 142

`ironicclient.osc.v1`, 142

`ironicclient.osc.v1.baremetal_allocation`,  
102

`ironicclient.osc.v1.baremetal_chassis`,  
105

`ironicclient.osc.v1.baremetal_conductor`,  
107

`ironicclient.osc.v1.baremetal_create`,  
108

`ironicclient.osc.v1.baremetal_deploy_template`,  
109

`ironicclient.osc.v1.baremetal_driver`,  
112

`ironicclient.osc.v1.baremetal_node`, 114

`ironicclient.osc.v1.baremetal_port`, 131

`ironicclient.osc.v1.baremetal_portgroup`,  
134

`ironicclient.osc.v1.baremetal_volume_connector`,  
136

`ironicclient.osc.v1.baremetal_volume_target`,  
139

### S

`ironicclient.shell`, 174

`ironicclient.v1`, 172

`ironicclient.v1.allocation`, 143

`ironicclient.v1.chassis`, 145

`ironicclient.v1.client`, 147

`ironicclient.v1.conductor`, 147

`ironicclient.v1.create_resources`, 148

`ironicclient.v1.deploy_template`, 151

`ironicclient.v1.driver`, 152

`ironicclient.v1.events`, 153

`ironicclient.v1.node`, 153

`ironicclient.v1.port`, 164

`ironicclient.v1.portgroup`, 166

`ironicclient.v1.resource_fields`, 168

`ironicclient.v1.utils`, 170

`ironicclient.v1.volume_connector`, 170

`ironicclient.v1.volume_target`, 171



## Symbols

- address
  - openstack-baremetal-port-group-create
    - command line option, 59
  - openstack-baremetal-port-group-list
    - command line option, 61
  - openstack-baremetal-port-group-set
    - command line option, 61
  - openstack-baremetal-port-group-show
    - command line option, 62
  - openstack-baremetal-port-group-unset
    - command line option, 63
  - openstack-baremetal-port-list
    - command line option, 64
  - openstack-baremetal-port-set
    - command line option, 65
  - openstack-baremetal-port-show
    - command line option, 66
- all
  - openstack-baremetal-node-remove-trait
    - command line option, 43
- arg
  - openstack-baremetal-driver-passthru-call
    - command line option, 20
  - openstack-baremetal-node-passthru-call
    - command line option, 40
- associated
  - openstack-baremetal-node-list
    - command line option, 38
- automated-clean
  - openstack-baremetal-node-create
    - command line option, 33
  - openstack-baremetal-node-set
    - command line option, 47
  - openstack-baremetal-node-unset
    - command line option, 52
- bios-interface
  - openstack-baremetal-node-create
    - command line option, 32
  - openstack-baremetal-node-set
    - command line option, 45
- openstack-baremetal-node-unset
  - command line option, 52
- boot-index
  - openstack-baremetal-volume-target-create
    - command line option, 73
  - openstack-baremetal-volume-target-set
    - command line option, 75
- boot-interface
  - openstack-baremetal-node-create
    - command line option, 32
  - openstack-baremetal-node-set
    - command line option, 46
  - openstack-baremetal-node-unset
    - command line option, 52
- candidate-node
  - openstack-baremetal-allocation-create
    - command line option, 9
- chassis
  - openstack-baremetal-node-list
    - command line option, 39
- chassis-uuid
  - openstack-baremetal-node-create
    - command line option, 31
  - openstack-baremetal-node-set
    - command line option, 45
  - openstack-baremetal-node-unset
    - command line option, 52
- clean-steps
  - openstack-baremetal-node-clean
    - command line option, 29
- column
  - openstack-baremetal-allocation-create
    - command line option, 8
  - openstack-baremetal-allocation-list
    - command line option, 10
  - openstack-baremetal-allocation-show
    - command line option, 12
  - openstack-baremetal-chassis-create
    - command line option, 14
  - openstack-baremetal-chassis-list
    - command line option, 15

openstack-baremetal-chassis-show	command line option, 66
command line option, 17	openstack-baremetal-volume-connector-create
openstack-baremetal-driver-list	command line option, 68
command line option, 18	openstack-baremetal-volume-connector-list
openstack-baremetal-driver-passthru-call	command line option, 69
command line option, 19	openstack-baremetal-volume-connector-show
openstack-baremetal-driver-passthru-list	command line option, 71
command line option, 20	openstack-baremetal-volume-target-create
openstack-baremetal-driver-property-list	command line option, 73
command line option, 21	openstack-baremetal-volume-target-list
openstack-baremetal-driver-raid-property-list	command line option, 74
command line option, 22	openstack-baremetal-volume-target-show
openstack-baremetal-driver-show	command line option, 76
command line option, 23	--conductor
openstack-baremetal-node-bios-setting-list	openstack-baremetal-node-list
command line option, 25	command line option, 39
openstack-baremetal-node-bios-setting-show	conductor-group
command line option, 26	openstack-baremetal-node-create
openstack-baremetal-node-boot-device-show	command line option, 33
command line option, 28	openstack-baremetal-node-list
openstack-baremetal-node-console-show	command line option, 38
command line option, 30	openstack-baremetal-node-set
openstack-baremetal-node-create	command line option, 47
command line option, 31	openstack-baremetal-node-unset
openstack-baremetal-node-history-get	command line option, 52
command line option, 34	--config-drive
openstack-baremetal-node-history-list	openstack-baremetal-node-deploy
command line option, 35	command line option, 34
openstack-baremetal-node-list	openstack-baremetal-node-rebuild
command line option, 37	command line option, 43
openstack-baremetal-node-passthru-list-connector-id	
command line option, 41	openstack-baremetal-volume-connector-create
openstack-baremetal-node-show	command line option, 68
command line option, 48	openstack-baremetal-volume-connector-set
openstack-baremetal-node-trait-list	command line option, 71
command line option, 49	--console-interface
openstack-baremetal-node-validate	openstack-baremetal-node-create
command line option, 53	command line option, 32
openstack-baremetal-node-vif-list	openstack-baremetal-node-set
command line option, 55	command line option, 46
openstack-baremetal-port-create	openstack-baremetal-node-unset
command line option, 56	command line option, 52
openstack-baremetal-port-group-create	--deploy-interface
command line option, 58	openstack-baremetal-node-create
openstack-baremetal-port-group-list	command line option, 32
command line option, 60	openstack-baremetal-node-set
openstack-baremetal-port-group-show	command line option, 46
command line option, 62	openstack-baremetal-node-unset
openstack-baremetal-port-list	command line option, 52
command line option, 64	--deploy-steps
openstack-baremetal-port-show	openstack-baremetal-node-deploy



- command line option, 34
- openstack-baremetal-node-rebuild
  - command line option, 43
- description
  - openstack-baremetal-chassis-create
    - command line option, 14
  - openstack-baremetal-chassis-set
    - command line option, 16
  - openstack-baremetal-chassis-unset
    - command line option, 17
  - openstack-baremetal-node-create
    - command line option, 33
  - openstack-baremetal-node-set
    - command line option, 48
  - openstack-baremetal-node-unset
    - command line option, 53
- description-contains
  - openstack-baremetal-node-list
    - command line option, 39
- driver
  - openstack-baremetal-node-create
    - command line option, 32
  - openstack-baremetal-node-list
    - command line option, 38
  - openstack-baremetal-node-set
    - command line option, 45
- driver-info
  - openstack-baremetal-node-create
    - command line option, 32
  - openstack-baremetal-node-set
    - command line option, 48
  - openstack-baremetal-node-unset
    - command line option, 52
- extra
  - openstack-baremetal-allocation-create
    - command line option, 9
  - openstack-baremetal-allocation-set
    - command line option, 12
  - openstack-baremetal-allocation-unset
    - command line option, 13
  - openstack-baremetal-chassis-create
    - command line option, 14
  - openstack-baremetal-chassis-set
    - command line option, 16
  - openstack-baremetal-chassis-unset
    - command line option, 17
  - openstack-baremetal-node-create
    - command line option, 32
  - openstack-baremetal-node-set
    - command line option, 47
  - openstack-baremetal-node-unset
    - command line option, 51
- openstack-baremetal-port-create
  - command line option, 57
- openstack-baremetal-port-group-create
  - command line option, 59
- openstack-baremetal-port-group-set
  - command line option, 61
- openstack-baremetal-port-group-unset
  - command line option, 63
- openstack-baremetal-port-set
  - command line option, 65
- openstack-baremetal-port-unset
  - command line option, 67
- openstack-baremetal-volume-connector-create
  - command line option, 68
- openstack-baremetal-volume-connector-set
  - command line option, 71
- openstack-baremetal-volume-connector-unset
  - command line option, 72
- openstack-baremetal-volume-target-create
  - command line option, 73
- openstack-baremetal-volume-target-set
  - command line option, 76
- openstack-baremetal-volume-target-unset
  - command line option, 77
- fault
  - openstack-baremetal-node-list
    - command line option, 38
- fields
  - openstack-baremetal-allocation-list
    - command line option, 11
  - openstack-baremetal-allocation-show
    - command line option, 13
  - openstack-baremetal-chassis-list
    - command line option, 15
  - openstack-baremetal-chassis-show
    - command line option, 17
  - openstack-baremetal-driver-list
    - command line option, 19
  - openstack-baremetal-driver-show
    - command line option, 24
  - openstack-baremetal-node-bios-setting-list
    - command line option, 26
  - openstack-baremetal-node-list
    - command line option, 39
  - openstack-baremetal-node-show
    - command line option, 49
  - openstack-baremetal-port-group-list
    - command line option, 61
  - openstack-baremetal-port-group-show
    - command line option, 62

openstack-baremetal-port-list	openstack-baremetal-node-list
command line option, 65	command line option, 37
openstack-baremetal-port-show	openstack-baremetal-node-passthru-list
command line option, 67	command line option, 41
openstack-baremetal-volume-connector-list	openstack-baremetal-node-show
command line option, 70	command line option, 49
openstack-baremetal-volume-connector-show	openstack-baremetal-node-trait-list
command line option, 72	command line option, 49
openstack-baremetal-volume-target-list	openstack-baremetal-node-validate
command line option, 75	command line option, 54
openstack-baremetal-volume-target-show	openstack-baremetal-node-vif-list
command line option, 76	command line option, 55
--fit-width	openstack-baremetal-port-create
openstack-baremetal-allocation-create	command line option, 57
command line option, 9	openstack-baremetal-port-group-create
openstack-baremetal-allocation-list	command line option, 58
command line option, 10	openstack-baremetal-port-group-list
openstack-baremetal-allocation-show	command line option, 60
command line option, 12	openstack-baremetal-port-group-show
openstack-baremetal-chassis-create	command line option, 62
command line option, 14	openstack-baremetal-port-list
openstack-baremetal-chassis-list	command line option, 64
command line option, 15	openstack-baremetal-port-show
openstack-baremetal-chassis-show	command line option, 66
command line option, 17	openstack-baremetal-volume-connector-create
openstack-baremetal-driver-list	command line option, 68
command line option, 19	openstack-baremetal-volume-connector-list
openstack-baremetal-driver-passthru-call	command line option, 69
command line option, 20	openstack-baremetal-volume-connector-show
openstack-baremetal-driver-passthru-list	command line option, 71
command line option, 21	openstack-baremetal-volume-target-create
openstack-baremetal-driver-property-list	command line option, 73
command line option, 22	openstack-baremetal-volume-target-list
openstack-baremetal-driver-raid-property-list	command line option, 74
command line option, 23	openstack-baremetal-volume-target-show
openstack-baremetal-driver-show	command line option, 76
command line option, 24	--format
openstack-baremetal-node-bios-setting-list	openstack-baremetal-allocation-create
command line option, 25	command line option, 8
openstack-baremetal-node-bios-setting-show	openstack-baremetal-allocation-list
command line option, 27	command line option, 10
openstack-baremetal-node-boot-device-show	openstack-baremetal-allocation-show
command line option, 28	command line option, 12
openstack-baremetal-node-console-show	openstack-baremetal-chassis-create
command line option, 30	command line option, 13
openstack-baremetal-node-create	openstack-baremetal-chassis-list
command line option, 31	command line option, 15
openstack-baremetal-node-history-get	openstack-baremetal-chassis-show
command line option, 35	command line option, 17
openstack-baremetal-node-history-list	openstack-baremetal-driver-list
command line option, 36	command line option, 18

openstack-baremetal-driver-passthru-call      command line option, 69  
     command line option, 19      openstack-baremetal-volume-connector-show  
 openstack-baremetal-driver-passthru-list      command line option, 71  
     command line option, 20      openstack-baremetal-volume-target-create  
 openstack-baremetal-driver-property-list      command line option, 72  
     command line option, 21      openstack-baremetal-volume-target-list  
 openstack-baremetal-driver-raid-property-list      command line option, 74  
     command line option, 22      openstack-baremetal-volume-target-show  
 openstack-baremetal-driver-show      command line option, 76  
     command line option, 23      --http-method  
 openstack-baremetal-node-bios-setting-list      openstack-baremetal-driver-passthru-call  
     command line option, 25      command line option, 20  
 openstack-baremetal-node-bios-setting-show      openstack-baremetal-node-passthru-call  
     command line option, 26      command line option, 40  
 openstack-baremetal-node-boot-device-show      show-  
     command line option, 27      inspect-interface  
 openstack-baremetal-node-console-show      openstack-baremetal-node-create  
     command line option, 30      command line option, 32  
 openstack-baremetal-node-create      openstack-baremetal-node-set  
     command line option, 31      command line option, 46  
 openstack-baremetal-node-history-get      openstack-baremetal-node-unset  
     command line option, 34      command line option, 52  
     --instance  
 openstack-baremetal-node-history-list      openstack-baremetal-node-show  
     command line option, 35      command line option, 49  
 openstack-baremetal-node-list      --instance-info  
     command line option, 37      openstack-baremetal-node-set  
 openstack-baremetal-node-passthru-list      command line option, 48  
     command line option, 41      openstack-baremetal-node-unset  
 openstack-baremetal-node-show      command line option, 52  
     command line option, 48      --instance-uuid  
 openstack-baremetal-node-trait-list      openstack-baremetal-node-set  
     command line option, 49      command line option, 45  
 openstack-baremetal-node-validate      openstack-baremetal-node-unset  
     command line option, 53      command line option, 51  
 openstack-baremetal-node-vif-list      --is-smartnic  
     command line option, 55      openstack-baremetal-port-create  
 openstack-baremetal-port-create      command line option, 57  
     command line option, 56      openstack-baremetal-port-set  
 openstack-baremetal-port-group-create      command line option, 66  
     command line option, 58      openstack-baremetal-port-unset  
 openstack-baremetal-port-group-list      command line option, 67  
     command line option, 60      --lessee  
 openstack-baremetal-port-group-show      openstack-baremetal-node-create  
     command line option, 62      command line option, 33  
 openstack-baremetal-port-list      openstack-baremetal-node-list  
     command line option, 64      command line option, 39  
 openstack-baremetal-port-show      openstack-baremetal-node-set  
     command line option, 66      command line option, 48  
 openstack-baremetal-volume-connector-create      openstack-baremetal-node-unset  
     command line option, 68      command line option, 53  
 openstack-baremetal-volume-connector-list      limit

openstack-baremetal-allocation-list command line option, 11	openstack-baremetal-allocation-list command line option, 11
openstack-baremetal-chassis-list command line option, 16	openstack-baremetal-chassis-list command line option, 16
openstack-baremetal-node-list command line option, 38	openstack-baremetal-node-list command line option, 38
openstack-baremetal-port-group-list command line option, 60	openstack-baremetal-port-group-list command line option, 60
openstack-baremetal-port-list command line option, 64	openstack-baremetal-port-list command line option, 64
openstack-baremetal-volume-connector-list command line option, 70	openstack-baremetal-volume-connector-list command line option, 70
openstack-baremetal-volume-target-list command line option, 75	openstack-baremetal-volume-target-list command line option, 75
--local-link-connection	--max-width
openstack-baremetal-port-create command line option, 57	openstack-baremetal-allocation-create command line option, 9
openstack-baremetal-port-set command line option, 65	openstack-baremetal-allocation-list command line option, 10
--long	openstack-baremetal-allocation-show command line option, 12
openstack-baremetal-allocation-list command line option, 11	openstack-baremetal-chassis-create command line option, 14
openstack-baremetal-chassis-list command line option, 16	openstack-baremetal-chassis-list command line option, 15
openstack-baremetal-driver-list command line option, 19	openstack-baremetal-chassis-show command line option, 17
openstack-baremetal-node-bios-setting-list command line option, 26	openstack-baremetal-driver-list command line option, 18
openstack-baremetal-node-history-list command line option, 36	openstack-baremetal-driver-passthru-call command line option, 20
openstack-baremetal-node-list command line option, 39	openstack-baremetal-driver-passthru-list command line option, 21
openstack-baremetal-port-group-list command line option, 61	openstack-baremetal-driver-property-list command line option, 22
openstack-baremetal-port-list command line option, 65	openstack-baremetal-driver-raid-property-list command line option, 23
openstack-baremetal-volume-connector-list command line option, 70	openstack-baremetal-driver-show command line option, 24
openstack-baremetal-volume-target-list command line option, 75	openstack-baremetal-node-bios-setting-list command line option, 25
--maintenance	openstack-baremetal-node-bios-setting-show command line option, 26
openstack-baremetal-node-list command line option, 38	openstack-baremetal-node-boot-device-show command line option, 28
--management-interface	openstack-baremetal-node-console-show command line option, 30
openstack-baremetal-node-create command line option, 32	openstack-baremetal-node-create command line option, 31
openstack-baremetal-node-set command line option, 46	openstack-baremetal-node-history-get command line option, 35
openstack-baremetal-node-unset command line option, 52	
--marker	

openstack-baremetal-node-history-list command line option, 32  
 command line option, 35  
 openstack-baremetal-node-list command line option, 37  
 openstack-baremetal-node-passthru-list command line option, 41  
 openstack-baremetal-node-show command line option, 48  
 openstack-baremetal-node-trait-list command line option, 49  
 openstack-baremetal-node-validate command line option, 54  
 openstack-baremetal-node-vif-list command line option, 55  
 openstack-baremetal-port-create command line option, 56  
 openstack-baremetal-port-group-create command line option, 58  
 openstack-baremetal-port-group-list command line option, 60  
 openstack-baremetal-port-group-show command line option, 62  
 openstack-baremetal-port-list command line option, 64  
 openstack-baremetal-port-show command line option, 66  
 openstack-baremetal-volume-connector-create command line option, 68  
 openstack-baremetal-volume-connector-list command line option, 69  
 openstack-baremetal-volume-connector-show command line option, 71  
 openstack-baremetal-volume-target-create command line option, 73  
 openstack-baremetal-volume-target-list command line option, 74  
 openstack-baremetal-volume-target-show-node command line option, 76  
 --mode  
 openstack-baremetal-port-group-create command line option, 59  
 openstack-baremetal-port-group-set command line option, 61  
 --name  
 openstack-baremetal-allocation-create command line option, 9  
 openstack-baremetal-allocation-set command line option, 12  
 openstack-baremetal-allocation-unset command line option, 13  
 openstack-baremetal-node-create command line option, 32  
 openstack-baremetal-node-set command line option, 45  
 openstack-baremetal-node-unset command line option, 51  
 openstack-baremetal-port-group-create command line option, 59  
 openstack-baremetal-port-group-set command line option, 61  
 openstack-baremetal-port-group-unset command line option, 63  
 --network-data  
 openstack-baremetal-node-create command line option, 32  
 openstack-baremetal-node-set command line option, 46  
 openstack-baremetal-node-unset command line option, 52  
 --network-interface  
 openstack-baremetal-node-create command line option, 32  
 openstack-baremetal-node-set command line option, 46  
 openstack-baremetal-node-unset command line option, 52  
 --no-automated-clean  
 openstack-baremetal-node-create command line option, 33  
 openstack-baremetal-node-set command line option, 47  
 --no-maintenance  
 openstack-baremetal-node-list command line option, 38  
 --no-retired  
 openstack-baremetal-node-list command line option, 38  
 openstack-baremetal-allocation-create command line option, 9  
 openstack-baremetal-allocation-list command line option, 11  
 openstack-baremetal-port-create command line option, 57  
 openstack-baremetal-port-group-create command line option, 59  
 openstack-baremetal-port-group-list command line option, 61  
 openstack-baremetal-port-group-set command line option, 61  
 openstack-baremetal-port-list command line option, 64

```

openstack-baremetal-port-set
    command line option, 65
openstack-baremetal-volume-connector-create
    command line option, 68
openstack-baremetal-volume-connector-list
    command line option, 70
openstack-baremetal-volume-connector-set
    command line option, 70
openstack-baremetal-volume-target-create
    command line option, 73
openstack-baremetal-volume-target-list
    command line option, 75
openstack-baremetal-volume-target-set
    command line option, 75
--noindent
openstack-baremetal-allocation-create
    command line option, 8
openstack-baremetal-allocation-list
    command line option, 10
openstack-baremetal-allocation-show
    command line option, 12
openstack-baremetal-chassis-create
    command line option, 14
openstack-baremetal-chassis-list
    command line option, 15
openstack-baremetal-chassis-show
    command line option, 17
openstack-baremetal-driver-list
    command line option, 18
openstack-baremetal-driver-passthru-call
    command line option, 20
openstack-baremetal-driver-passthru-list
    command line option, 21
openstack-baremetal-driver-property-list
    command line option, 22
openstack-baremetal-driver-raid-property-list
    command line option, 23
openstack-baremetal-driver-show
    command line option, 24
openstack-baremetal-node-bios-setting-list
    command line option, 25
openstack-baremetal-node-bios-setting-show
    command line option, 26
openstack-baremetal-node-boot-device-show
    command line option, 28
openstack-baremetal-node-console-show
    command line option, 30
openstack-baremetal-node-create
    command line option, 31
openstack-baremetal-node-history-get
    command line option, 34
openstack-baremetal-node-history-list
    command line option, 35
openstack-baremetal-node-list
    command line option, 37
openstack-baremetal-node-passthru-list
    command line option, 41
openstack-baremetal-node-show
    command line option, 48
openstack-baremetal-node-trait-list
    command line option, 49
openstack-baremetal-node-validate
    command line option, 53
openstack-baremetal-node-vif-list
    command line option, 55
openstack-baremetal-port-create
    command line option, 56
openstack-baremetal-port-group-create
    command line option, 58
openstack-baremetal-port-group-list
    command line option, 60
openstack-baremetal-port-group-show
    command line option, 62
openstack-baremetal-port-list
    command line option, 64
openstack-baremetal-port-show
    command line option, 66
openstack-baremetal-volume-connector-create
    command line option, 68
openstack-baremetal-volume-connector-list
    command line option, 69
openstack-baremetal-volume-connector-show
    command line option, 71
openstack-baremetal-volume-target-create
    command line option, 73
openstack-baremetal-volume-target-list
    command line option, 74
openstack-baremetal-volume-target-show
    command line option, 76
--owner
openstack-baremetal-allocation-create
    command line option, 9
openstack-baremetal-allocation-list
    command line option, 11
openstack-baremetal-node-create
    command line option, 33
openstack-baremetal-node-list
    command line option, 39
openstack-baremetal-node-set
    command line option, 48
openstack-baremetal-node-unset
    command line option, 53

```

- persistent
  - openstack-baremetal-node-boot-device-set
    - command line option, 27
- physical-network
  - openstack-baremetal-port-create
    - command line option, 57
  - openstack-baremetal-port-set
    - command line option, 66
  - openstack-baremetal-port-unset
    - command line option, 67
- port-group
  - openstack-baremetal-port-create
    - command line option, 57
  - openstack-baremetal-port-list
    - command line option, 64
  - openstack-baremetal-port-set
    - command line option, 65
  - openstack-baremetal-port-unset
    - command line option, 67
- port-uuid
  - openstack-baremetal-node-vif-attach
    - command line option, 54
- power-interface
  - openstack-baremetal-node-create
    - command line option, 32
  - openstack-baremetal-node-set
    - command line option, 46
  - openstack-baremetal-node-unset
    - command line option, 52
- power-timeout
  - openstack-baremetal-node-power-off
    - command line option, 41
  - openstack-baremetal-node-power-on
    - command line option, 42
  - openstack-baremetal-node-reboot
    - command line option, 43
- prefix
  - openstack-baremetal-allocation-create
    - command line option, 9
  - openstack-baremetal-allocation-show
    - command line option, 12
  - openstack-baremetal-chassis-create
    - command line option, 14
  - openstack-baremetal-chassis-show
    - command line option, 17
  - openstack-baremetal-driver-passthru-call
    - command line option, 20
  - openstack-baremetal-driver-show
    - command line option, 24
  - openstack-baremetal-node-bios-setting-show
    - command line option, 26
  - openstack-baremetal-node-boot-device-show
    - command line option, 28
  - openstack-baremetal-node-console-show
    - command line option, 30
  - openstack-baremetal-node-create
    - command line option, 31
  - openstack-baremetal-node-history-get
    - command line option, 35
  - openstack-baremetal-node-show
    - command line option, 48
  - openstack-baremetal-port-create
    - command line option, 56
  - openstack-baremetal-port-group-create
    - command line option, 58
  - openstack-baremetal-port-group-show
    - command line option, 62
  - openstack-baremetal-port-show
    - command line option, 66
  - openstack-baremetal-volume-connector-create
    - command line option, 68
  - openstack-baremetal-volume-connector-show
    - command line option, 71
  - openstack-baremetal-volume-target-create
    - command line option, 73
  - openstack-baremetal-volume-target-show
    - command line option, 76
- print-empty
  - openstack-baremetal-allocation-create
    - command line option, 9
  - openstack-baremetal-allocation-list
    - command line option, 11
  - openstack-baremetal-allocation-show
    - command line option, 13
  - openstack-baremetal-chassis-create
    - command line option, 14
  - openstack-baremetal-chassis-list
    - command line option, 15
  - openstack-baremetal-chassis-show
    - command line option, 17
  - openstack-baremetal-driver-list
    - command line option, 19
  - openstack-baremetal-driver-passthru-call
    - command line option, 20
  - openstack-baremetal-driver-passthru-list
    - command line option, 21
  - openstack-baremetal-driver-property-list
    - command line option, 22
  - openstack-baremetal-driver-raid-property-list
    - command line option, 23
  - openstack-baremetal-driver-show
    - command line option, 24

openstack-baremetal-node-bios-setting-list command line option, 25  
 openstack-baremetal-node-bios-setting-show command line option, 27  
 openstack-baremetal-node-boot-device-show command line option, 28  
 openstack-baremetal-node-console-show command line option, 30  
 openstack-baremetal-node-create command line option, 31  
 openstack-baremetal-node-history-get command line option, 35  
 openstack-baremetal-node-history-list command line option, 36  
 openstack-baremetal-node-list command line option, 37  
 openstack-baremetal-node-passthru-list command line option, 41  
 openstack-baremetal-node-show command line option, 49  
 openstack-baremetal-node-trait-list command line option, 49  
 openstack-baremetal-node-validate command line option, 54  
 openstack-baremetal-node-vif-list command line option, 55  
 openstack-baremetal-port-group-create command line option, 57  
 openstack-baremetal-port-group-create command line option, 58  
 openstack-baremetal-port-group-list command line option, 60  
 openstack-baremetal-port-group-show command line option, 62  
 openstack-baremetal-port-list command line option, 64  
 openstack-baremetal-port-show command line option, 66  
 openstack-baremetal-volume-connector-create command line option, 68  
 openstack-baremetal-volume-connector-list command line option, 70  
 openstack-baremetal-volume-connector-show command line option, 71  
 openstack-baremetal-volume-target-create command line option, 73  
 openstack-baremetal-volume-target-list command line option, 74  
 openstack-baremetal-volume-target-show command line option, 76  
 --property  
 openstack-baremetal-node-create command line option, 32  
 openstack-baremetal-node-set command line option, 47  
 openstack-baremetal-node-unset command line option, 51  
 openstack-baremetal-port-group-create command line option, 59  
 openstack-baremetal-port-group-set command line option, 61  
 openstack-baremetal-port-group-unset command line option, 63  
 openstack-baremetal-volume-target-create command line option, 73  
 openstack-baremetal-volume-target-set command line option, 75  
 openstack-baremetal-volume-target-unset command line option, 77  
 --protected  
 openstack-baremetal-node-set command line option, 47  
 openstack-baremetal-node-unset command line option, 53  
 --protected-reason  
 openstack-baremetal-node-set command line option, 47  
 openstack-baremetal-node-unset command line option, 53  
 --provision-state  
 openstack-baremetal-node-list command line option, 38  
 --pxe-disabled  
 openstack-baremetal-port-set command line option, 65  
 --pxe-enabled  
 openstack-baremetal-port-create command line option, 57  
 openstack-baremetal-port-set command line option, 65  
 --quote  
 openstack-baremetal-allocation-list command line option, 10  
 openstack-baremetal-chassis-list command line option, 15  
 openstack-baremetal-driver-list command line option, 18  
 openstack-baremetal-driver-passthru-list command line option, 21  
 openstack-baremetal-driver-property-list command line option, 22  
 openstack-baremetal-driver-raid-property-list



- command line option, 23
- openstack-baremetal-node-bios-setting-list
  - command line option, 25
- openstack-baremetal-node-history-list
  - command line option, 35
- openstack-baremetal-node-list
  - command line option, 37
- openstack-baremetal-node-passthru-list-reset-interfaces
  - command line option, 41
- openstack-baremetal-node-trait-list
  - command line option, 49
- openstack-baremetal-node-validate
  - command line option, 53
- openstack-baremetal-node-vif-list
  - command line option, 55
- openstack-baremetal-port-group-list
  - command line option, 60
- openstack-baremetal-port-list
  - command line option, 64
- openstack-baremetal-volume-connector-list
  - command line option, 69
- openstack-baremetal-volume-target-list
  - command line option, 74
- raid-interface
  - openstack-baremetal-node-create
    - command line option, 33
  - openstack-baremetal-node-set
    - command line option, 46
  - openstack-baremetal-node-unset
    - command line option, 52
- reason
  - openstack-baremetal-node-maintenance-set
    - command line option, 39
- rescue-interface
  - openstack-baremetal-node-create
    - command line option, 33
  - openstack-baremetal-node-set
    - command line option, 47
  - openstack-baremetal-node-unset
    - command line option, 52
- rescue-password
  - openstack-baremetal-node-rescue
    - command line option, 44
- reset-bios-interface
  - openstack-baremetal-node-set
    - command line option, 46
- reset-boot-interface
  - openstack-baremetal-node-set
    - command line option, 46
- reset-console-interface
  - openstack-baremetal-node-set
    - command line option, 46
- reset-deploy-interface
  - openstack-baremetal-node-set
    - command line option, 46
- reset-inspect-interface
  - openstack-baremetal-node-set
    - command line option, 46
- reset-management-interface
  - openstack-baremetal-node-set
    - command line option, 46
- reset-network-interface
  - openstack-baremetal-node-set
    - command line option, 46
- reset-power-interface
  - openstack-baremetal-node-set
    - command line option, 46
- reset-raid-interface
  - openstack-baremetal-node-set
    - command line option, 46
- reset-rescue-interface
  - openstack-baremetal-node-set
    - command line option, 47
- reset-storage-interface
  - openstack-baremetal-node-set
    - command line option, 47
- reset-vendor-interface
  - openstack-baremetal-node-set
    - command line option, 47
- resource-class
  - openstack-baremetal-allocation-create
    - command line option, 9
  - openstack-baremetal-allocation-list
    - command line option, 11
  - openstack-baremetal-node-create
    - command line option, 33
  - openstack-baremetal-node-list
    - command line option, 38
  - openstack-baremetal-node-set
    - command line option, 47
  - openstack-baremetal-node-unset
    - command line option, 51
- retired
  - openstack-baremetal-node-list
    - command line option, 38
  - openstack-baremetal-node-set
    - command line option, 47
  - openstack-baremetal-node-unset
    - command line option, 53

- retired-reason
  - openstack-baremetal-node-set
    - command line option, 47
  - openstack-baremetal-node-unset
    - command line option, 53
- soft
  - openstack-baremetal-node-power-off
    - command line option, 42
  - openstack-baremetal-node-reboot
    - command line option, 42
- sort
  - openstack-baremetal-allocation-list
    - command line option, 11
  - openstack-baremetal-chassis-list
    - command line option, 16
  - openstack-baremetal-node-list
    - command line option, 38
  - openstack-baremetal-port-group-list
    - command line option, 60
  - openstack-baremetal-port-list
    - command line option, 65
  - openstack-baremetal-volume-connector-list
    - command line option, 70
  - openstack-baremetal-volume-target-list
    - command line option, 75
- sort-ascending
  - openstack-baremetal-allocation-list
    - command line option, 11
  - openstack-baremetal-chassis-list
    - command line option, 15
  - openstack-baremetal-driver-list
    - command line option, 19
  - openstack-baremetal-driver-passthru-list
    - command line option, 21
  - openstack-baremetal-driver-property-list
    - command line option, 22
  - openstack-baremetal-driver-raid-property-list
    - command line option, 23
  - openstack-baremetal-node-bios-setting-list
    - command line option, 26
  - openstack-baremetal-node-history-list
    - command line option, 36
  - openstack-baremetal-node-list
    - command line option, 38
  - openstack-baremetal-node-passthru-list
    - command line option, 41
  - openstack-baremetal-node-trait-list
    - command line option, 50
  - openstack-baremetal-node-validate
    - command line option, 54
  - openstack-baremetal-node-vif-list
    - command line option, 55
- sort-column
  - openstack-baremetal-allocation-list
    - command line option, 11
  - openstack-baremetal-chassis-list
    - command line option, 15
  - openstack-baremetal-driver-list
    - command line option, 19
  - openstack-baremetal-driver-passthru-list
    - command line option, 21
  - openstack-baremetal-driver-property-list
    - command line option, 22
  - openstack-baremetal-driver-raid-property-list
    - command line option, 23
  - openstack-baremetal-node-bios-setting-list
    - command line option, 26
  - openstack-baremetal-node-history-list
    - command line option, 36
  - openstack-baremetal-node-list
    - command line option, 38
  - openstack-baremetal-node-passthru-list
    - command line option, 41
  - openstack-baremetal-node-trait-list
    - command line option, 50
  - openstack-baremetal-node-validate
    - command line option, 54
  - openstack-baremetal-node-vif-list
    - command line option, 55
- sort-descending
  - openstack-baremetal-allocation-list
    - command line option, 11
  - openstack-baremetal-chassis-list
    - command line option, 15
  - openstack-baremetal-driver-list
    - command line option, 19
- openstack-baremetal-port-group-list
  - command line option, 60
- openstack-baremetal-port-list
  - command line option, 64
- openstack-baremetal-volume-connector-list
  - command line option, 70
- openstack-baremetal-volume-target-list
  - command line option, 74
- openstack-baremetal-node-set
  - command line option, 56
- openstack-baremetal-port-group-list
  - command line option, 60
- openstack-baremetal-port-list
  - command line option, 64
- openstack-baremetal-volume-connector-list
  - command line option, 70
- openstack-baremetal-volume-target-list
  - command line option, 74

openstack-baremetal-driver-passthru-list `trait`  
     command line option, 21  
 openstack-baremetal-driver-property-list `command line option, 9`  
     command line option, 22  
     --type  
 openstack-baremetal-driver-raid-property-list `openstack-baremetal-driver-list`  
     command line option, 23  
     command line option, 19  
 openstack-baremetal-node-bios-setting-list `openstack-baremetal-volume-connector-create`  
     command line option, 26  
     command line option, 68  
 openstack-baremetal-node-history-list `openstack-baremetal-volume-connector-set`  
     command line option, 36  
     command line option, 71  
 openstack-baremetal-node-list `openstack-baremetal-volume-target-create`  
     command line option, 38  
     command line option, 73  
 openstack-baremetal-node-passthru-list `openstack-baremetal-volume-target-set`  
     command line option, 41  
     command line option, 75  
 openstack-baremetal-node-trait-list `--unassociated`  
     command line option, 50  
     openstack-baremetal-node-list  
     command line option, 38  
 openstack-baremetal-node-validate `--unsupport-standalone-ports`  
     command line option, 54  
     openstack-baremetal-port-group-create  
     command line option, 59  
 openstack-baremetal-node-vif-list `openstack-baremetal-port-group-set`  
     command line option, 56  
     command line option, 62  
 openstack-baremetal-port-group-list `--uuid`  
     command line option, 60  
     openstack-baremetal-allocation-create  
     command line option, 9  
 openstack-baremetal-port-list `openstack-baremetal-chassis-create`  
     command line option, 64  
     command line option, 14  
 openstack-baremetal-volume-connector-list `openstack-baremetal-node-create`  
     command line option, 70  
     command line option, 32  
 openstack-baremetal-volume-target-list `openstack-baremetal-port-create`  
     command line option, 75  
     command line option, 57  
 --state `openstack-baremetal-port-group-create`  
     openstack-baremetal-allocation-list `openstack-baremetal-volume-connector-create`  
         command line option, 11  
         command line option, 68  
 --storage-interface `openstack-baremetal-volume-target-create`  
     openstack-baremetal-node-create `openstack-baremetal-volume-target-set`  
         command line option, 33  
         command line option, 73  
     openstack-baremetal-node-set  
         command line option, 47  
     openstack-baremetal-node-unset  
         command line option, 52  
 --support-standalone-ports `--vendor-interface`  
     openstack-baremetal-port-group-create `openstack-baremetal-node-create`  
         command line option, 59  
         command line option, 33  
     openstack-baremetal-port-group-set `openstack-baremetal-node-set`  
         command line option, 61  
         command line option, 47  
     openstack-baremetal-node-unset `openstack-baremetal-node-unset`  
         command line option, 52  
         command line option, 52  
 --supported `openstack-baremetal-node-boot-device-show-info`  
     openstack-baremetal-node-boot-device-show-info `openstack-baremetal-node-vif-attach`  
         command line option, 28  
         command line option, 54  
 --target-raid-config `--volume-id`  
     openstack-baremetal-node-set `openstack-baremetal-volume-target-create`  
         command line option, 47  
         command line option, 73  
     openstack-baremetal-node-unset `openstack-baremetal-volume-target-set`  
         command line option, 51  
         command line option, 73

- command line option, 76
- wait
  - openstack-baremetal-allocation-create
    - command line option, 9
  - openstack-baremetal-node-adopt
    - command line option, 25
  - openstack-baremetal-node-clean
    - command line option, 29
  - openstack-baremetal-node-deploy
    - command line option, 34
  - openstack-baremetal-node-inspect
    - command line option, 36
  - openstack-baremetal-node-manage
    - command line option, 40
  - openstack-baremetal-node-provide
    - command line option, 42
  - openstack-baremetal-node-rebuild
    - command line option, 43
  - openstack-baremetal-node-rescue
    - command line option, 44
  - openstack-baremetal-node-undeploy
    - command line option, 50
  - openstack-baremetal-node-unrescue
    - command line option, 50
- c
  - openstack-baremetal-allocation-create
    - command line option, 8
  - openstack-baremetal-allocation-list
    - command line option, 10
  - openstack-baremetal-allocation-show
    - command line option, 12
  - openstack-baremetal-chassis-create
    - command line option, 14
  - openstack-baremetal-chassis-list
    - command line option, 15
  - openstack-baremetal-chassis-show
    - command line option, 17
  - openstack-baremetal-driver-list
    - command line option, 18
  - openstack-baremetal-driver-passthru-call
    - command line option, 19
  - openstack-baremetal-driver-passthru-list
    - command line option, 20
  - openstack-baremetal-driver-property-list
    - command line option, 21
  - openstack-baremetal-driver-raid-property-list
    - command line option, 22
  - openstack-baremetal-driver-show
    - command line option, 23
  - openstack-baremetal-node-bios-setting-list
    - command line option, 25
  - openstack-baremetal-node-bios-setting-show
    - command line option, 26
  - openstack-baremetal-node-boot-device-show
    - command line option, 28
  - openstack-baremetal-node-console-show
    - command line option, 30
  - openstack-baremetal-node-create
    - command line option, 31
  - openstack-baremetal-node-history-get
    - command line option, 34
  - openstack-baremetal-node-history-list
    - command line option, 35
  - openstack-baremetal-node-list
    - command line option, 37
  - openstack-baremetal-node-passthru-list
    - command line option, 41
  - openstack-baremetal-node-show
    - command line option, 48
  - openstack-baremetal-node-trait-list
    - command line option, 49
  - openstack-baremetal-node-validate
    - command line option, 53
  - openstack-baremetal-node-vif-list
    - command line option, 55
  - openstack-baremetal-port-create
    - command line option, 56
  - openstack-baremetal-port-group-create
    - command line option, 58
  - openstack-baremetal-port-group-list
    - command line option, 60
  - openstack-baremetal-port-group-show
    - command line option, 62
  - openstack-baremetal-port-list
    - command line option, 64
  - openstack-baremetal-port-show
    - command line option, 66
  - openstack-baremetal-volume-connector-create
    - command line option, 68
  - openstack-baremetal-volume-connector-list
    - command line option, 69
  - openstack-baremetal-volume-connector-show
    - command line option, 71
  - openstack-baremetal-volume-target-create
    - command line option, 73
  - openstack-baremetal-volume-target-list
    - command line option, 74
  - openstack-baremetal-volume-target-show
    - command line option, 76
- f
  - openstack-baremetal-allocation-create
    - command line option, 8

openstack-baremetal-allocation-list  
     command line option, 10  
 openstack-baremetal-allocation-show  
     command line option, 12  
 openstack-baremetal-chassis-create  
     command line option, 13  
 openstack-baremetal-chassis-list  
     command line option, 15  
 openstack-baremetal-chassis-show  
     command line option, 17  
 openstack-baremetal-driver-list  
     command line option, 18  
 openstack-baremetal-driver-passthru-call  
     command line option, 19  
 openstack-baremetal-driver-passthru-list  
     command line option, 20  
 openstack-baremetal-driver-property-list  
     command line option, 21  
 openstack-baremetal-driver-raid-property-list  
     command line option, 22  
 openstack-baremetal-driver-show  
     command line option, 23  
 openstack-baremetal-node-bios-setting-list  
     command line option, 25  
 openstack-baremetal-node-bios-setting-show  
     command line option, 26  
 openstack-baremetal-node-boot-device-show  
     command line option, 27  
 openstack-baremetal-node-console-show  
     command line option, 30  
 openstack-baremetal-node-create  
     command line option, 31  
 openstack-baremetal-node-history-get  
     command line option, 34  
 openstack-baremetal-node-history-list  
     command line option, 35  
 openstack-baremetal-node-list  
     command line option, 37  
 openstack-baremetal-node-passthru-list  
     command line option, 41  
 openstack-baremetal-node-show  
     command line option, 48  
 openstack-baremetal-node-trait-list  
     command line option, 49  
 openstack-baremetal-node-validate  
     command line option, 53  
 openstack-baremetal-node-vif-list  
     command line option, 55  
 openstack-baremetal-port-create  
     command line option, 56  
 openstack-baremetal-port-group-create  
     command line option, 58  
 openstack-baremetal-port-group-list  
     command line option, 60  
 openstack-baremetal-port-group-show  
     command line option, 62  
 openstack-baremetal-port-list  
     command line option, 64  
 openstack-baremetal-port-show  
     command line option, 66  
 openstack-baremetal-volume-connector-create  
     command line option, 68  
 openstack-baremetal-volume-connector-list  
     command line option, 69  
 openstack-baremetal-volume-connector-show  
     command line option, 71  
 openstack-baremetal-volume-target-create  
     command line option, 72  
 openstack-baremetal-volume-target-list  
     command line option, 74  
 openstack-baremetal-volume-target-show  
     command line option, 76  
 -l  
 openstack-baremetal-port-create  
     command line option, 57

## A

ShowBaremetalNode (*class in ironic-client.osc.v1.baremetal\_node*), 114  
 add\_hook() (*ironic-client.common.apiclient.base.HookableMixin class method*), 84  
 add\_trait() (*ironicclient.v1.node.NodeManager method*), 153  
 address  
 openstack-baremetal-port-create  
     command line option, 57  
 AddTraitBaremetalNode (*class in ironic-client.osc.v1.baremetal\_node*), 115  
 AdoptBaremetalNode (*class in ironic-client.osc.v1.baremetal\_node*), 115  
 allocation  
 openstack-baremetal-allocation-delete  
     command line option, 10  
 openstack-baremetal-allocation-set  
     command line option, 12  
 openstack-baremetal-allocation-unset  
     command line option, 13  
 Allocation (*class in ironicclient.v1.allocation*), 143  
 AllocationManager (*class in ironic-client.v1.allocation*), 143

- AmbiguousAuthSystem (in module *ironic-client.exc*), 173
- AmbiguousAuthSystem, 173
- AmbiguousEndpoints, 85
- App (class in *ironicclient.shell*), 174
- args\_array\_to\_dict() (in module *ironic-client.common.utils*), 99
- args\_array\_to\_patch() (in module *ironic-client.common.utils*), 99
- AuthorizationFailure, 86
- AuthPluginOptionsMissing, 85
- AuthSystemNotFound, 85
- B**
- BadGateway, 86
- BadRequest, 86
- baremetal (in *ironicclient.shell.ClientManager* property), 174
- baremetal\_introspection (in *ironic-client.shell.ClientManager* property), 174
- BaseManager (class in *ironic-client.common.apiclient.base*), 83
- BIOSSettingShowBaremetalNode (class in *ironicclient.osc.v1.baremetal\_node*), 115
- bool\_argument\_value() (in module *ironic-client.common.utils*), 99
- boot\_mode
  - openstack-baremetal-node-boot-mode-set command line option, 28
- BootdeviceSetBaremetalNode (class in *ironic-client.osc.v1.baremetal\_node*), 116
- BootdeviceShowBaremetalNode (class in *ironicclient.osc.v1.baremetal\_node*), 116
- BootmodeSetBaremetalNode (class in *ironic-client.osc.v1.baremetal\_node*), 116
- build\_option\_parser() (in module *ironic-client.osc.plugin*), 142
- build\_option\_parser() (*ironicclient.shell.App* method), 174
- build\_url() (in *ironic-client.common.apiclient.base.CrudManager* method), 83
- C**
- chassis
  - openstack-baremetal-chassis-delete command line option, 14
  - openstack-baremetal-chassis-set command line option, 16
  - openstack-baremetal-chassis-show command line option, 17
  - openstack-baremetal-chassis-unset command line option, 17
- Chassis (class in *ironicclient.v1.chassis*), 145
- ChassisManager (class in *ironic-client.v1.chassis*), 145
- check\_empty\_arg() (in module *ironic-client.common.utils*), 99
- check\_for\_invalid\_fields() (in module *ironicclient.common.utils*), 99
- CleanBaremetalNode (class in *ironic-client.osc.v1.baremetal\_node*), 117
- Client (class in *ironicclient.v1.client*), 147
- Client() (in module *ironicclient.client*), 172
- ClientException, 86
- ClientManager (class in *ironicclient.shell*), 174
- collection\_key (in *ironic-client.common.apiclient.base.CrudManager* attribute), 83
- CommandError, 86
- CommandManager (class in *ironicclient.shell*), 174
- common\_filters() (in module *ironic-client.common.utils*), 100
- common\_params\_for\_list() (in module *ironic-client.common.utils*), 100
- Conductor (class in *ironicclient.v1.conductor*), 147
- ConductorManager (class in *ironic-client.v1.conductor*), 147
- Conflict, 86
- ConnectionError, 87
- ConnectionRefused, 87
- ConsoleDisableBaremetalNode (class in *ironicclient.osc.v1.baremetal\_node*), 117
- ConsoleEnableBaremetalNode (class in *ironic-client.osc.v1.baremetal\_node*), 117
- ConsoleShowBaremetalNode (class in *ironic-client.osc.v1.baremetal\_node*), 118
- convert\_list\_props\_to\_comma\_separated() (in module *ironicclient.common.utils*), 100
- create() (in *ironic-client.common.apiclient.base.CrudManager* method), 83
- create() (in *ironic-client.common.base.CreateManager* method), 97
- create\_chassis() (in module *ironic-client.v1.create\_resources*), 148
- create\_nodes() (in module *ironic-client.v1.create\_resources*), 148
- create\_portgroups() (in module *ironic-*

- `client.v1.create_resources`), 148
  - `create_ports()` (in module `ironic-client.v1.create_resources`), 149
  - `create_resources()` (in module `ironic-client.v1.create_resources`), 149
  - `create_single_chassis()` (in module `ironic-client.v1.create_resources`), 149
  - `create_single_handler()` (in module `ironic-client.v1.create_resources`), 149
  - `create_single_node()` (in module `ironic-client.v1.create_resources`), 150
  - `create_single_port()` (in module `ironic-client.v1.create_resources`), 150
  - `create_single_portgroup()` (in module `ironicclient.v1.create_resources`), 150
  - `CreateBaremetal` (class in `ironic-client.osc.v1.baremetal_create`), 108
  - `CreateBaremetalAllocation` (class in `ironic-client.osc.v1.baremetal_allocation`), 102
  - `CreateBaremetalChassis` (class in `ironic-client.osc.v1.baremetal_chassis`), 105
  - `CreateBaremetalDeployTemplate` (class in `ironic-client.osc.v1.baremetal_deploy_template`), 109
  - `CreateBaremetalNode` (class in `ironic-client.osc.v1.baremetal_node`), 118
  - `CreateBaremetalPort` (class in `ironic-client.osc.v1.baremetal_port`), 131
  - `CreateBaremetalPortGroup` (class in `ironic-client.osc.v1.baremetal_portgroup`), 134
  - `CreateBaremetalVolumeConnector` (class in `ironic-client.osc.v1.baremetal_volume_connector`), 136
  - `CreateBaremetalVolumeTarget` (class in `ironic-client.osc.v1.baremetal_volume_target`), 139
  - `CreateManager` (class in `ironic-client.common.base`), 97
  - `CrudManager` (class in `ironic-client.common.apiclient.base`), 83
  - `current_api_version` (`ironic-client.v1.client.Client` property), 147
- D**
- `define_command()` (in module `ironic-client.common.utils`), 100
  - `define_commands_from_module()` (in module `ironicclient.common.utils`), 100
  - `delete()` (`ironic-client.common.apiclient.base.CrudManager` method), 83
  - `delete()` (`ironic-client.v1.allocation.AllocationManager` method), 143
  - `delete()` (`ironic-client.v1.chassis.ChassisManager` method), 145
  - `delete()` (`ironic-client.v1.deploy_template.DeployTemplateManager` method), 151
  - `delete()` (`ironicclient.v1.driver.DriverManager` method), 152
  - `delete()` (`ironicclient.v1.node.NodeManager` method), 154
  - `delete()` (`ironicclient.v1.port.PortManager` method), 164
  - `delete()` (`ironic-client.v1.portgroup.PortgroupManager` method), 166
  - `delete()` (`ironic-client.v1.volume_connector.VolumeConnectorManager` method), 170
  - `delete()` (`ironic-client.v1.volume_target.VolumeTargetManager` method), 171
  - `DeleteBaremetalAllocation` (class in `ironic-client.osc.v1.baremetal_allocation`), 102
  - `DeleteBaremetalChassis` (class in `ironic-client.osc.v1.baremetal_chassis`), 105
  - `DeleteBaremetalDeployTemplate` (class in `ironic-client.osc.v1.baremetal_deploy_template`), 109
  - `DeleteBaremetalNode` (class in `ironic-client.osc.v1.baremetal_node`), 119
  - `DeleteBaremetalPort` (class in `ironic-client.osc.v1.baremetal_port`), 132
  - `DeleteBaremetalPortGroup` (class in `ironic-client.osc.v1.baremetal_portgroup`), 134
  - `DeleteBaremetalVolumeConnector` (class in `ironic-client.osc.v1.baremetal_volume_connector`), 137
  - `DeleteBaremetalVolumeTarget` (class in `ironic-client.osc.v1.baremetal_volume_target`), 139
  - `DeployBaremetalNode` (class in `ironic-client.osc.v1.baremetal_node`), 119

DeployTemplate (class in *ironic-client.v1.deploy\_template*), 151  
 DeployTemplateManager (class in *ironic-client.v1.deploy\_template*), 151  
 device  
     openstack-baremetal-node-boot-device-set command line option, 27  
 driver  
     openstack-baremetal-driver-passthru-get command line option, 20  
     openstack-baremetal-driver-passthru-list command line option, 21  
     openstack-baremetal-driver-property-get command line option, 22  
     openstack-baremetal-driver-raid-property-get command line option, 23  
     openstack-baremetal-driver-show command line option, 24  
 Driver (class in *ironicclient.v1.driver*), 152  
 DriverManager (class in *ironicclient.v1.driver*), 152  
**E**  
 EndpointException, 87  
 EndpointNotFound, 87  
 event  
     openstack-baremetal-node-history-get command line option, 35  
 Event (class in *ironicclient.v1.events*), 153  
 EventManager (class in *ironicclient.v1.events*), 153  
 ExpectationFailed, 87  
 Extension (class in *ironic-client.common.apiclient.base*), 84  
**F**  
 FIELDS (*ironicclient.v1.resource\_fields.Resource* attribute), 168  
 fields (*ironicclient.v1.resource\_fields.Resource* property), 170  
 file  
     openstack-baremetal-create command line option, 18  
 find() (*ironicclient.common.apiclient.base.CrudManager* method), 83  
 find() (*ironicclient.common.apiclient.base.ManagerWithFind* method), 84  
 findall() (*ironic-client.common.apiclient.base.ManagerWithFind* method), 85  
 Forbidden, 87  
 from\_response() (in module *ironic-client.common.apiclient.exceptions*), 96  
 from\_response() (in module *ironicclient.exc*), 173  
**G**  
 GatewayTimeout, 88  
 get() (*ironicclient.common.apiclient.base.CrudManager* method), 83  
 get() (*ironicclient.common.apiclient.base.Resource* method), 85  
 get() (*ironicclient.v1.allocation.AllocationManager* method), 143  
 get\_list() (*ironicclient.v1.chassis.ChassisManager* method), 145  
 get() (*ironicclient.v1.conductor.ConductorManager* method), 147  
 get() (*ironicclient.v1.deploy\_template.DeployTemplateManager* method), 151  
 get() (*ironicclient.v1.driver.DriverManager* method), 152  
 get() (*ironicclient.v1.node.NodeManager* method), 154  
 get() (*ironicclient.v1.port.PortManager* method), 164  
 get() (*ironicclient.v1.portgroup.PortgroupManager* method), 166  
 get() (*ironicclient.v1.volume\_connector.VolumeConnectorManager* method), 170  
 get() (*ironicclient.v1.volume\_target.VolumeTargetManager* method), 171  
 get\_bios\_setting() (*ironic-client.v1.node.NodeManager* method), 154  
 get\_boot\_device() (*ironic-client.v1.node.NodeManager* method), 154  
 get\_by\_address() (*ironic-client.v1.port.PortManager* method), 164  
 get\_by\_address() (*ironic-client.v1.portgroup.PortgroupManager* method), 166  
 get\_by\_instance\_uuid() (*ironic-client.v1.node.NodeManager* method), 154  
 get\_client() (in module *ironicclient.client*), 172  
 get\_console() (*ironic-client.v1.node.NodeManager* method), 154



get_from_stdin()	(in module <i>ironic-client.common.utils</i> ), 101	<i>client.osc.v1.baremetal_create.CreateBaremetal</i> method), 108
get_history_event()	( <i>ironic-client.v1.node.NodeManager</i> method), 154	get_parser() ( <i>ironic-client.osc.v1.baremetal_deploy_template.CreateBaremetal</i> method), 109
get_history_list()	( <i>ironic-client.v1.node.NodeManager</i> method), 154	get_parser() ( <i>ironic-client.osc.v1.baremetal_deploy_template.DeleteBaremetal</i> method), 109
get_parser()	( <i>ironic-client.osc.v1.baremetal_allocation.CreateBaremetalAllocation</i> method), 102	get_parser() ( <i>ironic-client.osc.v1.baremetal_deploy_template.ListBaremetalDe</i> method), 110
get_parser()	( <i>ironic-client.osc.v1.baremetal_allocation.DeleteBaremetalAllocation</i> method), 102	get_parser() ( <i>ironic-client.osc.v1.baremetal_deploy_template.SetBaremetalDe</i> method), 110
get_parser()	( <i>ironic-client.osc.v1.baremetal_allocation.ListBaremetalAllocation</i> method), 103	get_parser() ( <i>ironic-client.osc.v1.baremetal_deploy_template.ShowBaremetalL</i> method), 111
get_parser()	( <i>ironic-client.osc.v1.baremetal_allocation.SetBaremetalAllocation</i> method), 103	get_parser() ( <i>ironic-client.osc.v1.baremetal_deploy_template.UnsetBaremetalL</i> method), 111
get_parser()	( <i>ironic-client.osc.v1.baremetal_allocation.ShowBaremetalAllocation</i> method), 104	get_parser() ( <i>ironic-client.osc.v1.baremetal_driver.ListBaremetalDriver</i> method), 112
get_parser()	( <i>ironic-client.osc.v1.baremetal_allocation.UnsetBaremetalAllocation</i> method), 104	get_parser() ( <i>ironic-client.osc.v1.baremetal_driver.ListBaremetalDriverPropert</i> method), 112
get_parser()	( <i>ironic-client.osc.v1.baremetal_chassis.CreateBaremetalChassis</i> method), 105	get_parser() ( <i>ironic-client.osc.v1.baremetal_driver.ListBaremetalDriverRaidP</i> method), 113
get_parser()	( <i>ironic-client.osc.v1.baremetal_chassis.DeleteBaremetalChassis</i> method), 105	get_parser() ( <i>ironic-client.osc.v1.baremetal_driver.PassthruCallBaremetalDriv</i> method), 113
get_parser()	( <i>ironic-client.osc.v1.baremetal_chassis.ListBaremetalChassis</i> method), 105	get_parser() ( <i>ironic-client.osc.v1.baremetal_driver.PassthruListBaremetalDriv</i> method), 114
get_parser()	( <i>ironic-client.osc.v1.baremetal_chassis.SetBaremetalChassis</i> method), 106	get_parser() ( <i>ironic-client.osc.v1.baremetal_driver.ShowBaremetalDriver</i> method), 114
get_parser()	( <i>ironic-client.osc.v1.baremetal_chassis.ShowBaremetalChassis</i> method), 106	get_parser() ( <i>ironic-client.osc.v1.baremetal_node.AddTraitBaremetalNode</i> method), 115
get_parser()	( <i>ironic-client.osc.v1.baremetal_chassis.UnsetBaremetalChassis</i> method), 107	get_parser() ( <i>ironic-client.osc.v1.baremetal_node.BIOSSettingShowBaremetal</i> method), 115
get_parser()	( <i>ironic-client.osc.v1.baremetal_conductor.ListBaremetalConductors</i> method), 107	get_parser() ( <i>ironic-client.osc.v1.baremetal_node.BootdeviceSetBaremetalNo</i> method), 116
get_parser()	( <i>ironic-client.osc.v1.baremetal_conductor.ShowBaremetalConductors</i> method), 108	get_parser() ( <i>ironic-client.osc.v1.baremetal_node.BootdeviceShowBaremetalN</i> method), 116
get_parser()	( <i>ironic-client.osc.v1.baremetal_conductor.ShowBaremetalConductors</i> method), 108	get_parser() ( <i>ironic-client.osc.v1.baremetal_node.BootdeviceShowBaremetalN</i> method), 116

<i>client.osc.v1.baremetal_node.BootmodeSetBaremetalNode</i>	<i>client.osc.v1.baremetal_node.PassthruListBaremetalNode</i>
method), 117	method), 124
get_parser()	(ironic- get_parser()) (ironic-
<i>client.osc.v1.baremetal_node.CleanBaremetalNode</i>	<i>client.osc.v1.baremetal_node.PowerBaremetalNode</i>
method), 117	method), 124
get_parser()	(ironic- get_parser()) (ironic-
<i>client.osc.v1.baremetal_node.ConsoleDisableBaremetalNode</i>	<i>client.osc.v1.baremetal_node.PowerOffBaremetalNode</i>
method), 117	method), 124
get_parser()	(ironic- get_parser()) (ironic-
<i>client.osc.v1.baremetal_node.ConsoleEnableBaremetalNode</i>	<i>client.osc.v1.baremetal_node.ProvisionStateBaremetalNode</i>
method), 118	method), 125
get_parser()	(ironic- get_parser()) (ironic-
<i>client.osc.v1.baremetal_node.ConsoleShowBaremetalNode</i>	<i>client.osc.v1.baremetal_node.ProvisionStateWithWait</i>
method), 118	method), 126
get_parser()	(ironic- get_parser()) (ironic-
<i>client.osc.v1.baremetal_node.CreateBaremetalNode</i>	<i>client.osc.v1.baremetal_node.RebootBaremetalNode</i>
method), 118	method), 126
get_parser()	(ironic- get_parser()) (ironic-
<i>client.osc.v1.baremetal_node.DeleteBaremetalNode</i>	<i>client.osc.v1.baremetal_node.RebuildBaremetalNode</i>
method), 119	method), 126
get_parser()	(ironic- get_parser()) (ironic-
<i>client.osc.v1.baremetal_node.DeployBaremetalNode</i>	<i>client.osc.v1.baremetal_node.RemoveTraitBaremetalNode</i>
method), 119	method), 127
get_parser()	(ironic- get_parser()) (ironic-
<i>client.osc.v1.baremetal_node.InjectNmiBaremetalNode</i>	<i>client.osc.v1.baremetal_node.RescueBaremetalNode</i>
method), 119	method), 127
get_parser()	(ironic- get_parser()) (ironic-
<i>client.osc.v1.baremetal_node.ListBaremetalNode</i>	<i>client.osc.v1.baremetal_node.SecurebootOffBaremetalNode</i>
method), 121	method), 127
get_parser()	(ironic- get_parser()) (ironic-
<i>client.osc.v1.baremetal_node.ListBIOSSettingBaremetalNode</i>	<i>client.osc.v1.baremetal_node.SecurebootOnBaremetalNode</i>
method), 120	method), 128
get_parser()	(ironic- get_parser()) (ironic-
<i>client.osc.v1.baremetal_node.ListTraitsBaremetalNode</i>	<i>client.osc.v1.baremetal_node.SetBaremetalNode</i>
method), 121	method), 128
get_parser()	(ironic- get_parser()) (ironic-
<i>client.osc.v1.baremetal_node.MaintenanceSetBaremetalNode</i>	<i>client.osc.v1.baremetal_node.ShowBaremetalNode</i>
method), 121	method), 129
get_parser()	(ironic- get_parser()) (ironic-
<i>client.osc.v1.baremetal_node.MaintenanceUnsetBaremetalNode</i>	<i>client.osc.v1.baremetal_node.UnsetBaremetalNode</i>
method), 122	method), 129
get_parser()	(ironic- get_parser()) (ironic-
<i>client.osc.v1.baremetal_node.NodeHistoryEventGet</i>	<i>client.osc.v1.baremetal_node.ValidateBaremetalNode</i>
method), 122	method), 130
get_parser()	(ironic- get_parser()) (ironic-
<i>client.osc.v1.baremetal_node.NodeHistoryList</i>	<i>client.osc.v1.baremetal_node.VifAttachBaremetalNode</i>
method), 123	method), 130
get_parser()	(ironic- get_parser()) (ironic-
<i>client.osc.v1.baremetal_node.PassthruCallBaremetalNode</i>	<i>client.osc.v1.baremetal_node.VifDetachBaremetalNode</i>
method), 123	method), 131
get_parser()	(ironic- get_parser()) (ironic-

*client.osc.v1.baremetal\_node.VifListBaremetalNode* *client.osc.v1.baremetal\_volume\_connector.ShowBaremetalVolumeConnector* (method), 131 (method), 138

*get\_parser()* (ironic- *get\_parser()* (ironic- *client.osc.v1.baremetal\_port.CreateBaremetalPort* *client.osc.v1.baremetal\_volume\_connector.UnsetBaremetalVolumeConnector* method), 131 (method), 139

*get\_parser()* (ironic- *get\_parser()* (ironic- *client.osc.v1.baremetal\_port.DeleteBaremetalPort* *client.osc.v1.baremetal\_volume\_target.CreateBaremetalVolumeTarget* method), 132 (method), 139

*get\_parser()* (ironic- *get\_parser()* (ironic- *client.osc.v1.baremetal\_port.ListBaremetalPort* *client.osc.v1.baremetal\_volume\_target.DeleteBaremetalVolumeTarget* method), 132 (method), 140

*get\_parser()* (ironic- *get\_parser()* (ironic- *client.osc.v1.baremetal\_port.SetBaremetalPort* *client.osc.v1.baremetal\_volume\_target.ListBaremetalVolumeTargets* method), 132 (method), 140

*get\_parser()* (ironic- *get\_parser()* (ironic- *client.osc.v1.baremetal\_port.ShowBaremetalPort* *client.osc.v1.baremetal\_volume\_target.SetBaremetalVolumeTarget* method), 133 (method), 141

*get\_parser()* (ironic- *get\_parser()* (ironic- *client.osc.v1.baremetal\_port.UnsetBaremetalPort* *client.osc.v1.baremetal\_volume\_target.ShowBaremetalVolumeTargets* method), 133 (method), 141

*get\_parser()* (ironic- *get\_parser()* (ironic- *client.osc.v1.baremetal\_portgroup.CreateBaremetalPortGroup* *client.osc.v1.baremetal\_volume\_target.UnsetBaremetalVolumeTarget* method), 134 (method), 142

*get\_parser()* (ironic- *get\_server()* (in module *ironic-client.osc.v1.baremetal\_portgroup.DeleteBaremetalPortGroup* *client.common.http*), 98 (method), 134

*get\_parser()* (ironic- *get\_supported\_boot\_devices()* (ironic- *client.v1.node.NodeManager* method), *client.osc.v1.baremetal\_portgroup.ListBaremetalPortGroups* (method), 135

*get\_parser()* (ironic- *get\_traits()* (ironic- *client.v1.node.NodeManager* method), *client.osc.v1.baremetal\_portgroup.SetBaremetalPortGroup* (method), 135

*get\_parser()* (ironic- *get\_vendor\_passthru\_methods()* (ironic- *client.v1.driver.DriverManager* method), *client.osc.v1.baremetal\_portgroup.ShowBaremetalPortGroups* (method), 136

*get\_parser()* (ironic- *get\_vendor\_passthru\_methods()* (ironic- *client.v1.node.NodeManager* method), *client.osc.v1.baremetal\_portgroup.UnsetBaremetalPortGroup* (method), 136

*get\_parser()* (ironic- *getid()* (in module *ironic-client.osc.v1.baremetal\_volume\_connector.Getid()* *client.common.apiclient.base*), 85 (method), 136

*get\_parser()* (ironic- *Gone*, 88 *client.osc.v1.baremetal\_volume\_connector.DeleteBaremetalVolumeConnector* (method), 137

*get\_parser()* (ironic- *handle\_json\_arg()* (in module *ironic-client.osc.v1.baremetal\_volume\_connector.ListBaremetalVolumeConnectors*), 101 (method), 137

*get\_parser()* (ironic- *handle\_json\_or\_file\_arg()* (in module *ironic-client.common.utils*), 101 *client.osc.v1.baremetal\_volume\_connector.SetBaremetalVolumeConnector* (method), 138

*get\_parser()* (ironic- *HelpFormatter* (class in *ironic-client.common.apiclient.base.CrudManager* method), 83

`client.common.utils`), 99

HookableMixin (class in `ironic-client.common.apiclient.base`), 84

http\_status (ironic-`client.common.apiclient.exceptions.BadGateway` attribute), 86

http\_status (ironic-`client.common.apiclient.exceptions.BadRequest` attribute), 86

http\_status (ironic-`client.common.apiclient.exceptions.Conflict` attribute), 87

http\_status (ironic-`client.common.apiclient.exceptions.ExpectationFailed` attribute), 87

http\_status (ironic-`client.common.apiclient.exceptions.Forbidden` attribute), 88

http\_status (ironic-`client.common.apiclient.exceptions.GatewayTimeout` attribute), 88

http\_status (ironic-`client.common.apiclient.exceptions.Gone` attribute), 88

http\_status (ironic-`client.common.apiclient.exceptions.HttpError` attribute), 89

http\_status (ironic-`client.common.apiclient.exceptions.HttpNotImplemented` attribute), 90

http\_status (ironic-`client.common.apiclient.exceptions.HttpVersionNotSupported` attribute), 90

http\_status (ironic-`client.common.apiclient.exceptions.InternalServerError` attribute), 91

http\_status (ironic-`client.common.apiclient.exceptions.LengthRequired` attribute), 91

http\_status (ironic-`client.common.apiclient.exceptions.MethodNotAllowed` attribute), 91

http\_status (ironic-`client.common.apiclient.exceptions.MultipleChoices` attribute), 92

http\_status (ironic-`client.common.apiclient.exceptions.NotAcceptable` attribute), 92

http\_status (ironic-`client.common.apiclient.exceptions.NotFound` attribute), 93

http\_status (ironic-`client.common.apiclient.exceptions.PaymentRequired` attribute), 93

http\_status (ironic-`client.common.apiclient.exceptions.PreconditionFailed` attribute), 93

http\_status (ironic-`client.common.apiclient.exceptions.ProxyAuthenticationRequired` attribute), 94

http\_status (ironic-`client.common.apiclient.exceptions.RequestedRangeNotSatisfiable` attribute), 95

http\_status (ironic-`client.common.apiclient.exceptions.RequestEntityTooLarge` attribute), 94

http\_status (ironic-`client.common.apiclient.exceptions.RequestTimeout` attribute), 94

http\_status (ironic-`client.common.apiclient.exceptions.RequestUriTooLong` attribute), 94

http\_status (ironic-`client.common.apiclient.exceptions.ServiceUnavailable` attribute), 95

http\_status (ironic-`client.common.apiclient.exceptions.Unauthorized` attribute), 95

http\_status (ironic-`client.common.apiclient.exceptions.UnprocessableEntity` attribute), 96

http\_status (ironic-`client.common.apiclient.exceptions.UnsupportedMediaType` attribute), 96

HTTPClientError, 88

SetupError, 89

HttpNotImplemented, 89

HTTPRedirection, 89

HttpException, 90

HttpVersionNotSupported, 90

HUMAN\_ID (ironic-`client.common.apiclient.base.Resource` attribute), 85

human\_id (ironic-`client.common.apiclient.base.Resource` property), 85

id

openstack-baremetal-allocation-show command line option, 13

openstack-baremetal-port-group-show

command line option, 63  
 openstack-baremetal-port-show  
   command line option, 67  
 openstack-baremetal-volume-connector-show  
   command line option, 72  
 openstack-baremetal-volume-target-show  
   command line option, 77  
 initialize\_app() (*ironicclient.shell.App*  
   *method*), 174  
 inject\_nmi() (*ironic-*  
   *client.v1.node.NodeManager* *method*),  
   155  
 InjectNmiBaremetalNode (*class in ironic-*  
   *client.osc.v1.baremetal\_node*), 119  
 InspectBaremetalNode (*class in ironic-*  
   *client.osc.v1.baremetal\_node*), 120  
 InternalServerError, 91  
 InvalidAttribute, 173  
 ironicclient  
   module, 174  
 ironicclient.client  
   module, 172  
 ironicclient.common  
   module, 102  
 ironicclient.common.apiclient  
   module, 97  
 ironicclient.common.apiclient.base  
   module, 83  
 ironicclient.common.apiclient.exceptions  
   module, 85  
 ironicclient.common.base  
   module, 97  
 ironicclient.common.filecache  
   module, 98  
 ironicclient.common.http  
   module, 98  
 ironicclient.common.i18n  
   module, 99  
 ironicclient.common.utils  
   module, 99  
 ironicclient.exc  
   module, 173  
 ironicclient.osc  
   module, 143  
 ironicclient.osc.plugin  
   module, 142  
 ironicclient.osc.v1  
   module, 142  
 ironicclient.osc.v1.baremetal\_allocation  
   module, 102  
 ironicclient.osc.v1.baremetal\_chassis  
   module, 105  
 ironicclient.osc.v1.baremetal\_conductor  
   module, 107  
 ironicclient.osc.v1.baremetal\_create  
   module, 108  
 ironicclient.osc.v1.baremetal\_deploy\_template  
   module, 109  
 ironicclient.osc.v1.baremetal\_driver  
   module, 112  
 ironicclient.osc.v1.baremetal\_node  
   module, 114  
 ironicclient.osc.v1.baremetal\_port  
   module, 131  
 ironicclient.osc.v1.baremetal\_portgroup  
   module, 134  
 ironicclient.osc.v1.baremetal\_volume\_connector  
   module, 136  
 ironicclient.osc.v1.baremetal\_volume\_target  
   module, 139  
 ironicclient.shell  
   module, 174  
 ironicclient.v1  
   module, 172  
 ironicclient.v1.allocation  
   module, 143  
 ironicclient.v1.chassis  
   module, 145  
 ironicclient.v1.client  
   module, 147  
 ironicclient.v1.conductor  
   module, 147  
 ironicclient.v1.create\_resources  
   module, 148  
 ironicclient.v1.deploy\_template  
   module, 151  
 ironicclient.v1.driver  
   module, 152  
 ironicclient.v1.events  
   module, 153  
 ironicclient.v1.node  
   module, 153  
 ironicclient.v1.port  
   module, 164  
 ironicclient.v1.portgroup  
   module, 166  
 ironicclient.v1.resource\_fields  
   module, 168  
 ironicclient.v1.utils  
   module, 170  
 ironicclient.v1.volume\_connector  
   module, 170

ironicclient.v1.volume\_target module, 171  
 is\_api\_version\_negotiated (*ironic-client.v1.client.Client* property), 147  
 is\_loaded() (*ironic-client.common.apiclient.base.Resource* method), 85  
**J**  
 json\_request() (*ironic-client.common.http.SessionClient* method), 98  
**K**  
 key (*ironicclient.common.apiclient.base.CrudManager* attribute), 84  
 key\_value\_pairs\_to\_dict() (*in module ironicclient.common.utils*), 101  
**L**  
 labels (*ironicclient.v1.resource\_fields.Resource* property), 170  
 LengthRequired, 91  
 license agreement, 81  
 list() (*ironicclient.common.apiclient.base.CrudManager* method), 84  
 list() (*ironicclient.common.apiclient.base.ManagerWithFind* method), 85  
 list() (*ironicclient.v1.allocation.AllocationManager* method), 143  
 list() (*ironicclient.v1.chassis.ChassisManager* method), 145  
 list() (*ironicclient.v1.conductor.ConductorManager* method), 147  
 list() (*ironicclient.v1.deploy\_template.DeployTemplateManager* method), 151  
 list() (*ironicclient.v1.driver.DriverManager* method), 152  
 list() (*ironicclient.v1.node.NodeManager* method), 155  
 list() (*ironicclient.v1.port.PortManager* method), 164  
 list() (*ironicclient.v1.portgroup.PortgroupManager* method), 166  
 list() (*ironicclient.v1.volume\_connector.VolumeConnectorManager* method), 170  
 list() (*ironicclient.v1.volume\_target.VolumeTargetManager* method), 171  
 list\_bios\_settings() (*ironic-client.v1.node.NodeManager* method), 156  
 list\_nodes() (*ironic-client.v1.chassis.ChassisManager* method), 146  
 list\_ports() (*ironic-client.v1.node.NodeManager* method), 157  
 list\_ports() (*ironic-client.v1.portgroup.PortgroupManager* method), 167  
 list\_volume\_connectors() (*ironic-client.v1.node.NodeManager* method), 157  
 list\_volume\_targets() (*ironic-client.v1.node.NodeManager* method), 158  
 ListBaremetalAllocation (*class in ironic-client.osc.v1.baremetal\_allocation*), 103  
 ListBaremetalChassis (*class in ironic-client.osc.v1.baremetal\_chassis*), 105  
 ListBaremetalConductor (*class in ironic-client.osc.v1.baremetal\_conductor*), 107  
 ListBaremetalDeployTemplate (*class in ironic-client.osc.v1.baremetal\_deploy\_template*), 109  
 ListBaremetalDriver (*class in ironic-client.osc.v1.baremetal\_driver*), 112  
 ListBaremetalDriverProperty (*class in ironic-client.osc.v1.baremetal\_driver*), 112  
 ListBaremetalDriverRaidProperty (*class in ironic-client.osc.v1.baremetal\_driver*), 112  
 ListBaremetalNode (*class in ironic-client.osc.v1.baremetal\_node*), 120  
 ListBaremetalPort (*class in ironic-client.osc.v1.baremetal\_port*), 132  
 ListBaremetalPortGroup (*class in ironic-client.osc.v1.baremetal\_portgroup*), 134  
 ListBaremetalVolumeConnector (*class in ironic-client.osc.v1.baremetal\_volume\_connector*), 137  
 ListBaremetalVolumeTarget (*class in ironic-client.osc.v1.baremetal\_volume\_target*), 140  
 ListBIOSSettingBaremetalNode (*class in ironic-client.osc.v1.baremetal\_node*), 120  
 ListTraitsBaremetalNode (*class in ironic-client.osc.v1.baremetal\_node*), 121

`load_commands()` (*ironic-client.shell.CommandManager* method), 174  
`load_from_file()` (in module *ironic-client.v1.create\_resources*), 150  
`log(ironicclient.osc.v1.baremetal_allocation.CreateBaremetalAllocation)`, 114  
`log(ironicclient.osc.v1.baremetal_allocation.DeleteBaremetalAllocation)`, 114  
`log(ironicclient.osc.v1.baremetal_allocation.ListBaremetalAllocation)`, 115  
`log(ironicclient.osc.v1.baremetal_allocation.SetBaremetalAllocation)`, 115  
`log(ironicclient.osc.v1.baremetal_allocation.ShowBaremetalAllocation)`, 115  
`log(ironicclient.osc.v1.baremetal_allocation.UnsetBaremetalAllocation)`, 115  
`log(ironicclient.osc.v1.baremetal_chassis.CreateBaremetalChassis)`, 116  
`log(ironicclient.osc.v1.baremetal_chassis.DeleteBaremetalChassis)`, 116  
`log(ironicclient.osc.v1.baremetal_chassis.ListBaremetalChassis)`, 117  
`log(ironicclient.osc.v1.baremetal_chassis.SetBaremetalChassis)`, 117  
`log(ironicclient.osc.v1.baremetal_chassis.ShowBaremetalChassis)`, 117  
`log(ironicclient.osc.v1.baremetal_chassis.UnsetBaremetalChassis)`, 118  
`log(ironicclient.osc.v1.baremetal_conductor.ListBaremetalConductor)`, 118  
`log(ironicclient.osc.v1.baremetal_conductor.ShowBaremetalConductor)`, 118  
`log(ironicclient.osc.v1.baremetal_create.CreateBaremetal)`, 119  
`log(ironicclient.osc.v1.baremetal_deploy_template.CreateBaremetalDeployTemplate)`, 119  
`log(ironicclient.osc.v1.baremetal_deploy_template.DeleteBaremetalDeployTemplate)`, 119  
`log(ironicclient.osc.v1.baremetal_deploy_template.ListBaremetalDeployTemplate)`, 119  
`log(ironicclient.osc.v1.baremetal_deploy_template.SetBaremetalDeployTemplate)`, 119  
`log(ironicclient.osc.v1.baremetal_deploy_template.ShowBaremetalDeployTemplate)`, 119  
`log(ironicclient.osc.v1.baremetal_deploy_template.UnsetBaremetalDeployTemplate)`, 119  
`log(ironicclient.osc.v1.baremetal_driver.ListBaremetalDriver)`, 121  
`log(ironicclient.osc.v1.baremetal_driver.ListBaremetalDriver)`, 122  
`log(ironicclient.osc.v1.baremetal_driver.ListBaremetalDriverRaids)`, 113  
`log(ironicclient.osc.v1.baremetal_driver.PassthruCallBaremetalDriver)`, 113  
`log(ironicclient.osc.v1.baremetal_driver.PassthruListBaremetalDriver)`, 113  
`log(ironicclient.osc.v1.baremetal_driver.ShowBaremetalDriver)`, 114  
`log(ironicclient.osc.v1.baremetal_node.AbortBaremetalNode)`, 114  
`log(ironicclient.osc.v1.baremetal_node.AddTraitBaremetalNode)`, 115  
`log(ironicclient.osc.v1.baremetal_node.AdoptBaremetalNode)`, 115  
`log(ironicclient.osc.v1.baremetal_node.BIOSSettingShowBaremetalNode)`, 115  
`log(ironicclient.osc.v1.baremetal_node.BootdeviceSetBaremetalNode)`, 115  
`log(ironicclient.osc.v1.baremetal_node.BootdeviceShowBaremetalNode)`, 115  
`log(ironicclient.osc.v1.baremetal_node.BootmodeSetBaremetalNode)`, 115  
`log(ironicclient.osc.v1.baremetal_node.CleanBaremetalNode)`, 117  
`log(ironicclient.osc.v1.baremetal_node.ConsoleDisableBaremetalNode)`, 117  
`log(ironicclient.osc.v1.baremetal_node.ConsoleEnableBaremetalNode)`, 117  
`log(ironicclient.osc.v1.baremetal_node.ConsoleShowBaremetalNode)`, 117  
`log(ironicclient.osc.v1.baremetal_node.CreateBaremetalNode)`, 118  
`log(ironicclient.osc.v1.baremetal_node.DeleteBaremetalNode)`, 118  
`log(ironicclient.osc.v1.baremetal_node.DeployBaremetalNode)`, 119  
`log(ironicclient.osc.v1.baremetal_node.InjectNmiBaremetalNode)`, 119  
`log(ironicclient.osc.v1.baremetal_node.InspectBaremetalNode)`, 119  
`log(ironicclient.osc.v1.baremetal_node.ListBaremetalNode)`, 119  
`log(ironicclient.osc.v1.baremetal_node.ListBIOSSettingBaremetalNode)`, 119  
`log(ironicclient.osc.v1.baremetal_node.ListTraitsBaremetalNode)`, 119  
`log(ironicclient.osc.v1.baremetal_node.MaintenanceSetBaremetalNode)`, 121  
`log(ironicclient.osc.v1.baremetal_node.MaintenanceUnsetBaremetalNode)`, 121  
`log(ironicclient.osc.v1.baremetal_node.ManageBaremetalNode)`, 122

attribute), 122  
 log (ironicclient.osc.v1.baremetal\_node.NodeHistoryEventGetattribute), 131  
 attribute), 122  
 log (ironicclient.osc.v1.baremetal\_node.NodeHistoryList attribute), 132  
 attribute), 123  
 log (ironicclient.osc.v1.baremetal\_node.PassthruCallBaremetalNode), 132  
 attribute), 123  
 log (ironicclient.osc.v1.baremetal\_node.PassthruListBaremetalNode), 132  
 attribute), 124  
 log (ironicclient.osc.v1.baremetal\_node.PowerBaremetalNodeattribute), 133  
 attribute), 124  
 log (ironicclient.osc.v1.baremetal\_node.PowerOffBaremetalNodeattribute), 133  
 attribute), 124  
 log (ironicclient.osc.v1.baremetal\_node.PowerOnBaremetalNodeattribute), 134  
 attribute), 125  
 log (ironicclient.osc.v1.baremetal\_node.ProvideBaremetalNodeattribute), 134  
 attribute), 125  
 log (ironicclient.osc.v1.baremetal\_node.ProvisionStateBaremetalNode), 135  
 attribute), 125  
 log (ironicclient.osc.v1.baremetal\_node.ProvisionStateWithWaitattribute), 135  
 attribute), 126  
 log (ironicclient.osc.v1.baremetal\_node.RebootBaremetalNodeattribute), 136  
 attribute), 126  
 log (ironicclient.osc.v1.baremetal\_node.RebuildBaremetalNodeattribute), 136  
 attribute), 126  
 log (ironicclient.osc.v1.baremetal\_node.RemoveTraitBaremetalNode), 137  
 attribute), 127  
 log (ironicclient.osc.v1.baremetal\_node.RescueBaremetalNodeattribute), 137  
 attribute), 127  
 log (ironicclient.osc.v1.baremetal\_node.SecurebootOffBaremetalNode), 137  
 attribute), 127  
 log (ironicclient.osc.v1.baremetal\_node.SecurebootOnBaremetalNode), 138  
 attribute), 128  
 log (ironicclient.osc.v1.baremetal\_node.SetBaremetalNode attribute), 138  
 attribute), 128  
 log (ironicclient.osc.v1.baremetal\_node.ShowBaremetalNodeattribute), 139  
 attribute), 129  
 log (ironicclient.osc.v1.baremetal\_node.UndeployBaremetalNodeattribute), 139  
 attribute), 129  
 log (ironicclient.osc.v1.baremetal\_node.UnrescueBaremetalNodeattribute), 140  
 attribute), 129  
 log (ironicclient.osc.v1.baremetal\_node.UnsetBaremetalNodeattribute), 140  
 attribute), 129  
 log (ironicclient.osc.v1.baremetal\_node.ValidateBaremetalNodeattribute), 141  
 attribute), 130  
 log (ironicclient.osc.v1.baremetal\_node.VifAttachBaremetalNodeattribute), 141  
 attribute), 130  
 log (ironicclient.osc.v1.baremetal\_node.VifDetachBaremetalNodeattribute), 142  
 attribute), 131  
 log (ironicclient.osc.v1.baremetal\_node.VifListBaremetalNode  
 attribute), 131

**M**  
 main() (in module ironicclient.shell), 174



MaintenanceSetBaremetalNode (class in *ironicclient.osc.v1.baremetal\_node*), 121

MaintenanceUnsetBaremetalNode (class in *ironicclient.osc.v1.baremetal\_node*), 122

make\_client() (in module *ironicclient.osc.plugin*), 142

make\_configdrive() (in module *ironicclient.common.utils*), 101

ManageBaremetalNode (class in *ironicclient.osc.v1.baremetal\_node*), 122

Manager (class in *ironicclient.common.base*), 97

manager\_class (ironic-client.common.apiclient.base.Extension attribute), 84

ManagerWithFind (class in *ironicclient.common.apiclient.base*), 84

message (ironic-client.common.apiclient.exceptions.BadGateway attribute), 86

message (ironic-client.common.apiclient.exceptions.BadRequest attribute), 86

message (ironic-client.common.apiclient.exceptions.Conflict attribute), 87

message (ironic-client.common.apiclient.exceptions.ExpectationFailed attribute), 87

message (ironic-client.common.apiclient.exceptions.Forbidden attribute), 88

message (ironic-client.common.apiclient.exceptions.GatewayTimeout attribute), 88

message (ironic-client.common.apiclient.exceptions.Gone attribute), 88

message (ironic-client.common.apiclient.exceptions.HTTPClientError attribute), 89

message (ironic-client.common.apiclient.exceptions.HttpError attribute), 89

message (ironic-client.common.apiclient.exceptions.HttpNotImplemented attribute), 90

message (ironic-client.common.apiclient.exceptions.HTTPRedirection attribute), 89

message (ironic-client.common.apiclient.exceptions.HttpServerError attribute), 90

message (ironic-client.common.apiclient.exceptions.HttpVersionNotSupported attribute), 91

message (ironic-client.common.apiclient.exceptions.InternalServerError attribute), 91

message (ironic-client.common.apiclient.exceptions.LengthRequired attribute), 91

message (ironic-client.common.apiclient.exceptions.MethodNotAllowed attribute), 92

message (ironic-client.common.apiclient.exceptions.MultipleChoices attribute), 92

message (ironic-client.common.apiclient.exceptions.NotAcceptable attribute), 92

message (ironic-client.common.apiclient.exceptions.NotFound attribute), 93

message (ironic-client.common.apiclient.exceptions.PaymentRequired attribute), 93

message (ironic-client.common.apiclient.exceptions.PreconditionFailed attribute), 93

message (ironic-client.common.apiclient.exceptions.ProxyAuthenticationRequired attribute), 94

message (ironic-client.common.apiclient.exceptions.RequestedRangeNotSatisfiable attribute), 95

message (ironic-client.common.apiclient.exceptions.RequestEntityTooLarge attribute), 94

message (ironic-client.common.apiclient.exceptions.RequestTimeout attribute), 94

message (ironic-client.common.apiclient.exceptions.RequestUriTooLong attribute), 95

message (ironic-client.common.apiclient.exceptions.ServiceUnavailable attribute), 95

message (ironic-client.common.apiclient.exceptions.Unauthorized attribute), 96

message (ironic-client.common.apiclient.exceptions.UnprocessableEntity attribute), 96

- attribute*), 96
  - message
    - (*ironic-client.common.apiclient.exceptions.UnsupportedMethod* attribute), 96
  - method
    - openstack-baremetal-driver-passthru-call command line option, 20
    - openstack-baremetal-node-passthru-call command line option, 40
  - MethodNotAllowed, 91
  - module
    - ironicclient, 174
    - ironicclient.client, 172
    - ironicclient.common, 102
    - ironicclient.common.apiclient, 97
    - ironicclient.common.apiclient.base, 83
    - ironicclient.common.apiclient.exceptions, 85
    - ironicclient.common.base, 97
    - ironicclient.common.filecache, 98
    - ironicclient.common.http, 98
    - ironicclient.common.i18n, 99
    - ironicclient.common.utils, 99
    - ironicclient.exc, 173
    - ironicclient.osc, 143
    - ironicclient.osc.plugin, 142
    - ironicclient.osc.v1, 142
    - ironicclient.osc.v1.baremetal\_allocation, 102
    - ironicclient.osc.v1.baremetal\_chassis, 105
    - ironicclient.osc.v1.baremetal\_conductor, 107
    - ironicclient.osc.v1.baremetal\_create, 108
    - ironicclient.osc.v1.baremetal\_deploy\_template, 109
    - ironicclient.osc.v1.baremetal\_driver, 112
    - ironicclient.osc.v1.baremetal\_node, 114
    - ironicclient.osc.v1.baremetal\_port, 131
    - ironicclient.osc.v1.baremetal\_portgroup, 134
    - ironicclient.osc.v1.baremetal\_volume\_connector, 136
    - ironicclient.osc.v1.baremetal\_volume\_target, 139
    - ironicclient.shell, 174
    - ironicclient.v1, 172
    - ironicclient.v1.allocation, 143
    - ironicclient.v1.chassis, 145
    - ironicclient.v1.client, 147
    - ironicclient.v1.conductor, 147
    - ironicclient.v1.create\_resources, 148
    - ironicclient.v1.deploy\_template, 151
    - ironicclient.v1.driver, 152
    - ironicclient.v1.events, 153
    - ironicclient.v1.node, 153
    - ironicclient.v1.port, 164
    - ironicclient.v1.portgroup, 166
    - ironicclient.v1.resource\_fields, 168
    - ironicclient.v1.utils, 170
    - ironicclient.v1.volume\_connector, 170
    - ironicclient.v1.volume\_target, 171
  - MultipleChoices, 92
- ## N
- NAME\_ATTR (*ironic-client.common.apiclient.base.Resource* attribute), 85
  - negotiate\_api\_version() (*ironic-client.v1.client.Client* method), 147
  - negotiate\_version() (*ironic-client.common.http.VersionNegotiationMixin* method), 98
  - node
    - openstack-baremetal-node-abort command line option, 24
    - openstack-baremetal-node-add-trait command line option, 24
    - openstack-baremetal-node-adopt command line option, 25
    - openstack-baremetal-node-bios-setting-list command line option, 26
    - openstack-baremetal-node-bios-setting-show command line option, 27
    - openstack-baremetal-node-boot-device-set command line option, 27
    - openstack-baremetal-node-boot-device-show command line option, 28
    - openstack-baremetal-node-boot-mode-set command line option, 28
    - openstack-baremetal-node-clean command line option, 29
    - openstack-baremetal-node-console-disable command line option, 29
    - openstack-baremetal-node-console-enable

- command line option, 29
  - openstack-baremetal-node-console-show
    - command line option, 30
  - openstack-baremetal-node-delete
    - command line option, 33
  - openstack-baremetal-node-deploy
    - command line option, 34
  - openstack-baremetal-node-history-get
    - command line option, 35
  - openstack-baremetal-node-history-list
    - command line option, 36
  - openstack-baremetal-node-inject-nmi
    - command line option, 36
  - openstack-baremetal-node-inspect
    - command line option, 36
  - openstack-baremetal-node-maintenance-mode
    - command line option, 39
  - openstack-baremetal-node-maintenance-mode-set
    - command line option, 39
  - openstack-baremetal-node-manage
    - command line option, 40
  - openstack-baremetal-node-passthru-call
    - command line option, 40
  - openstack-baremetal-node-passthru-list
    - command line option, 41
  - openstack-baremetal-node-power-off
    - command line option, 42
  - openstack-baremetal-node-power-on
    - command line option, 42
  - openstack-baremetal-node-provide
    - command line option, 42
  - openstack-baremetal-node-reboot
    - command line option, 43
  - openstack-baremetal-node-rebuild
    - command line option, 43
  - openstack-baremetal-node-remove-trait
    - command line option, 43
  - openstack-baremetal-node-rescue
    - command line option, 44
  - openstack-baremetal-node-secure-boot-off
    - command line option, 44
  - openstack-baremetal-node-secure-boot-on
    - command line option, 44
  - openstack-baremetal-node-set
    - command line option, 48
  - openstack-baremetal-node-show
    - command line option, 49
  - openstack-baremetal-node-trait-list
    - command line option, 50
  - openstack-baremetal-node-undeploy
    - command line option, 50
  - openstack-baremetal-node-unrescue
    - command line option, 50
  - openstack-baremetal-node-unset
    - command line option, 53
  - openstack-baremetal-node-validate
    - command line option, 54
  - openstack-baremetal-node-vif-attach
    - command line option, 54
  - openstack-baremetal-node-vif-detach
    - command line option, 55
  - openstack-baremetal-node-vif-list
    - command line option, 56
  - Node (class in *ironicclient.v1.node*), 153
  - NodeHistoryEventGet (class in *ironicclient.osc.v1.baremetal\_node*), 122
  - NodeHistoryList (class in *ironicclient.osc.v1.baremetal\_node*), 123
  - NodeManager (class in *ironicclient.v1.node*), 153
  - NotAcceptable, 92
  - NotFound, 92
  - NoUniqueMatch, 92
- O
- openstack-baremetal-allocation-create
    - command line option
    - candidate-node, 9
    - column, 8
    - extra, 9
    - fit-width, 9
    - format, 8
    - max-width, 9
    - name, 9
    - node, 9
    - noindent, 8
    - owner, 9
    - prefix, 9
    - print-empty, 9
    - resource-class, 9
    - trait, 9
    - uuid, 9
    - wait, 9
    - c, 8
    - f, 8
  - openstack-baremetal-allocation-delete
    - command line option
    - allocation, 10
  - openstack-baremetal-allocation-list
    - command line option
    - column, 10
    - fields, 11
    - fit-width, 10

```

--format, 10
--limit, 11
--long, 11
--marker, 11
--max-width, 10
--node, 11
--noindent, 10
--owner, 11
--print-empty, 11
--quote, 10
--resource-class, 11
--sort, 11
--sort-ascending, 11
--sort-column, 11
--sort-descending, 11
--state, 11
-c, 10
-f, 10
openstack-baremetal-allocation-set
    command line option
--extra, 12
--name, 12
allocation, 12
openstack-baremetal-allocation-show
    command line option
--column, 12
--fields, 13
--fit-width, 12
--format, 12
--max-width, 12
--noindent, 12
--prefix, 12
--print-empty, 13
-c, 12
-f, 12
id, 13
openstack-baremetal-allocation-unset
    command line option
--extra, 13
--name, 13
allocation, 13
openstack-baremetal-chassis-create
    command line option
--column, 14
--description, 14
--extra, 14
--fit-width, 14
--format, 13
--max-width, 14
--noindent, 14
--prefix, 14
--print-empty, 14
--uuid, 14
-c, 14
-f, 13
openstack-baremetal-chassis-delete
    command line option
chassis, 14
openstack-baremetal-chassis-list
    command line option
--column, 15
--fields, 15
--fit-width, 15
--format, 15
--limit, 16
--long, 16
--marker, 16
--max-width, 15
--noindent, 15
--print-empty, 15
--quote, 15
--sort, 16
--sort-ascending, 15
--sort-column, 15
--sort-descending, 15
-c, 15
-f, 15
openstack-baremetal-chassis-set
    command line option
--description, 16
--extra, 16
chassis, 16
openstack-baremetal-chassis-show
    command line option
--column, 17
--fields, 17
--fit-width, 17
--format, 17
--max-width, 17
--noindent, 17
--prefix, 17
--print-empty, 17
-c, 17
-f, 17
chassis, 17
openstack-baremetal-chassis-unset
    command line option
--description, 17
--extra, 17
chassis, 17
openstack-baremetal-create command
line option

```

file, 18

openstack-baremetal-driver-list  
 command line option

- column, 18
- fields, 19
- fit-width, 19
- format, 18
- long, 19
- max-width, 18
- noindent, 18
- print-empty, 19
- quote, 18
- sort-ascending, 19
- sort-column, 19
- sort-descending, 19
- type, 19
- c, 18
- f, 18

openstack-baremetal-driver-passthru-call  
 command line option

- arg, 20
- column, 19
- fit-width, 20
- format, 19
- http-method, 20
- max-width, 20
- noindent, 20
- prefix, 20
- print-empty, 20
- c, 19
- f, 19

driver, 20

method, 20

openstack-baremetal-driver-passthru-list  
 command line option

- column, 20
- fit-width, 21
- format, 20
- max-width, 21
- noindent, 21
- print-empty, 21
- quote, 21
- sort-ascending, 21
- sort-column, 21
- sort-descending, 21
- c, 20
- f, 20

driver, 21

openstack-baremetal-driver-property-list  
 command line option

- column, 21
- fit-width, 22
- format, 21
- max-width, 22
- noindent, 22
- print-empty, 22
- quote, 22
- sort-ascending, 22
- sort-column, 22
- sort-descending, 22
- c, 21
- f, 21

driver, 22

openstack-baremetal-driver-raid-property-list  
 command line option

- column, 22
- fit-width, 23
- format, 22
- max-width, 23
- noindent, 23
- print-empty, 23
- quote, 23
- sort-ascending, 23
- sort-column, 23
- sort-descending, 23
- c, 22
- f, 22

driver, 23

openstack-baremetal-driver-show  
 command line option

- column, 23
- fields, 24
- fit-width, 24
- format, 23
- max-width, 24
- noindent, 24
- prefix, 24
- print-empty, 24
- c, 23
- f, 23

driver, 24

openstack-baremetal-node-abort command  
 line option

node, 24

openstack-baremetal-node-add-trait  
 command line option

node, 24

trait, 24

openstack-baremetal-node-adopt command  
 line option

- wait, 25

node, 25

**openstack-baremetal-node-bios-setting-list** node, 28  
     command line option  
     --column, 25  
     --fields, 26  
     --fit-width, 25  
     --format, 25  
     --long, 26  
     --max-width, 25  
     --noindent, 25  
     --print-empty, 25  
     --quote, 25  
     --sort-ascending, 26  
     --sort-column, 26  
     --sort-descending, 26  
     -c, 25  
     -f, 25  
     node, 26

**openstack-baremetal-node-bios-setting-show**  
     command line option  
     --column, 26  
     --fit-width, 27  
     --format, 26  
     --max-width, 26  
     --noindent, 26  
     --prefix, 26  
     --print-empty, 27  
     -c, 26  
     -f, 26  
     node, 27  
     setting, 27

**openstack-baremetal-node-boot-device-set**  
     command line option  
     --persistent, 27  
     device, 27  
     node, 27

**openstack-baremetal-node-boot-device-show**  
     command line option  
     --column, 28  
     --fit-width, 28  
     --format, 27  
     --max-width, 28  
     --noindent, 28  
     --prefix, 28  
     --print-empty, 28  
     --supported, 28  
     -c, 28  
     -f, 27  
     node, 28

**openstack-baremetal-node-boot-mode-set**  
     command line option  
     boot\_mode, 28

**openstack-baremetal-node-clean** command  
     line option  
     --clean-steps, 29  
     --wait, 29  
     node, 29

**openstack-baremetal-node-console-disable**  
     command line option  
     node, 29

**openstack-baremetal-node-console-enable**  
     command line option  
     node, 29

**openstack-baremetal-node-console-show**  
     command line option  
     --column, 30  
     --fit-width, 30  
     --format, 30  
     --max-width, 30  
     --noindent, 30  
     --prefix, 30  
     --print-empty, 30  
     -c, 30  
     -f, 30  
     node, 30

**openstack-baremetal-node-create**  
     command line option  
     --automated-clean, 33  
     --bios-interface, 32  
     --boot-interface, 32  
     --chassis-uuid, 31  
     --column, 31  
     --conductor-group, 33  
     --console-interface, 32  
     --deploy-interface, 32  
     --description, 33  
     --driver, 32  
     --driver-info, 32  
     --extra, 32  
     --fit-width, 31  
     --format, 31  
     --inspect-interface, 32  
     --lessee, 33  
     --management-interface, 32  
     --max-width, 31  
     --name, 32  
     --network-data, 32  
     --network-interface, 32  
     --no-automated-clean, 33  
     --noindent, 31  
     --owner, 33  
     --power-interface, 32

- prefix, 31
- print-empty, 31
- property, 32
- raid-interface, 33
- rescue-interface, 33
- resource-class, 33
- storage-interface, 33
- uuid, 32
- vendor-interface, 33
- c, 31
- f, 31
- openstack-baremetal-node-delete
  - command line option
  - node, 33
- openstack-baremetal-node-deploy
  - command line option
  - config-drive, 34
  - deploy-steps, 34
  - wait, 34
  - node, 34
- openstack-baremetal-node-history-get
  - command line option
  - column, 34
  - fit-width, 35
  - format, 34
  - max-width, 35
  - noindent, 34
  - prefix, 35
  - print-empty, 35
  - c, 34
  - f, 34
  - event, 35
  - node, 35
- openstack-baremetal-node-history-list
  - command line option
  - column, 35
  - fit-width, 36
  - format, 35
  - long, 36
  - max-width, 35
  - noindent, 35
  - print-empty, 36
  - quote, 35
  - sort-ascending, 36
  - sort-column, 36
  - sort-descending, 36
  - c, 35
  - f, 35
  - node, 36
- openstack-baremetal-node-inject-nmi
  - command line option
- node, 36
- openstack-baremetal-node-inspect
  - command line option
  - wait, 36
  - node, 36
- openstack-baremetal-node-list command
  - line option
  - associated, 38
  - chassis, 39
  - column, 37
  - conductor, 39
  - conductor-group, 38
  - description-contains, 39
  - driver, 38
  - fault, 38
  - fields, 39
  - fit-width, 37
  - format, 37
  - lessee, 39
  - limit, 38
  - long, 39
  - maintenance, 38
  - marker, 38
  - max-width, 37
  - no-maintenance, 38
  - no-retired, 38
  - noindent, 37
  - owner, 39
  - print-empty, 37
  - provision-state, 38
  - quote, 37
  - resource-class, 38
  - retired, 38
  - sort, 38
  - sort-ascending, 38
  - sort-column, 38
  - sort-descending, 38
  - unassociated, 38
  - c, 37
  - f, 37
- openstack-baremetal-node-maintenance-set
  - command line option
  - reason, 39
  - node, 39
- openstack-baremetal-node-maintenance-unset
  - command line option
  - node, 39
- openstack-baremetal-node-manage
  - command line option
  - wait, 40
  - node, 40

**openstack-baremetal-node-passthru-call**  
 command line option  
 --arg, 40  
 --http-method, 40  
 method, 40  
 node, 40

**openstack-baremetal-node-passthru-list**  
 command line option  
 --column, 41  
 --fit-width, 41  
 --format, 41  
 --max-width, 41  
 --noindent, 41  
 --print-empty, 41  
 --quote, 41  
 --sort-ascending, 41  
 --sort-column, 41  
 --sort-descending, 41  
 -c, 41  
 -f, 41  
 node, 41

**openstack-baremetal-node-power-off**  
 command line option  
 --power-timeout, 41  
 --soft, 42  
 node, 42

**openstack-baremetal-node-power-on**  
 command line option  
 --power-timeout, 42  
 node, 42

**openstack-baremetal-node-provide**  
 command line option  
 --wait, 42  
 node, 42

**openstack-baremetal-node-reboot**  
 command line option  
 --power-timeout, 43  
 --soft, 42  
 node, 43

**openstack-baremetal-node-rebuild**  
 command line option  
 --config-drive, 43  
 --deploy-steps, 43  
 --wait, 43  
 node, 43

**openstack-baremetal-node-remove-trait**  
 command line option  
 --all, 43  
 node, 43  
 trait, 44

**openstack-baremetal-node-rescue**  
 command line option  
 --rescue-password, 44  
 --wait, 44  
 node, 44

**openstack-baremetal-node-secure-boot-off**  
 command line option  
 node, 44

**openstack-baremetal-node-secure-boot-on**  
 command line option  
 node, 44

**openstack-baremetal-node-set** command line option  
 --automated-clean, 47  
 --bios-interface, 45  
 --boot-interface, 46  
 --chassis-uuid, 45  
 --conductor-group, 47  
 --console-interface, 46  
 --deploy-interface, 46  
 --description, 48  
 --driver, 45  
 --driver-info, 48  
 --extra, 47  
 --inspect-interface, 46  
 --instance-info, 48  
 --instance-uuid, 45  
 --lessee, 48  
 --management-interface, 46  
 --name, 45  
 --network-data, 46  
 --network-interface, 46  
 --no-automated-clean, 47  
 --owner, 48  
 --power-interface, 46  
 --property, 47  
 --protected, 47  
 --protected-reason, 47  
 --raid-interface, 46  
 --rescue-interface, 47  
 --reset-bios-interface, 46  
 --reset-boot-interface, 46  
 --reset-console-interface, 46  
 --reset-deploy-interface, 46  
 --reset-inspect-interface, 46  
 --reset-interfaces, 47  
 --reset-management-interface, 46  
 --reset-network-interface, 46  
 --reset-power-interface, 46  
 --reset-raid-interface, 46  
 --reset-rescue-interface, 47  
 --reset-storage-interface, 47



```

--reset-vendor-interface, 47
--resource-class, 47
--retired, 47
--retired-reason, 47
--storage-interface, 47
--target-raid-config, 47
--vendor-interface, 47
node, 48
openstack-baremetal-node-show command
    line option
--column, 48
--fields, 49
--fit-width, 49
--format, 48
--instance, 49
--max-width, 48
--noindent, 48
--prefix, 48
--print-empty, 49
-c, 48
-f, 48
node, 49
openstack-baremetal-node-trait-list
    command line option
--column, 49
--fit-width, 49
--format, 49
--max-width, 49
--noindent, 49
--print-empty, 49
--quote, 49
--sort-ascending, 50
--sort-column, 50
--sort-descending, 50
-c, 49
-f, 49
node, 50
openstack-baremetal-node-undeploy
    command line option
--wait, 50
node, 50
openstack-baremetal-node-unrescue
    command line option
--wait, 50
node, 50
openstack-baremetal-node-unset command
    line option
--automated-clean, 52
--bios-interface, 52
--boot-interface, 52
--chassis-uuid, 52
--conductor-group, 52
--console-interface, 52
--deploy-interface, 52
--description, 53
--driver-info, 52
--extra, 51
--inspect-interface, 52
--instance-info, 52
--instance-uuid, 51
--lessee, 53
--management-interface, 52
--name, 51
--network-data, 52
--network-interface, 52
--owner, 53
--power-interface, 52
--property, 51
--protected, 53
--protected-reason, 53
--raid-interface, 52
--rescue-interface, 52
--resource-class, 51
--retired, 53
--retired-reason, 53
--storage-interface, 52
--target-raid-config, 51
--vendor-interface, 52
node, 53
openstack-baremetal-node-validate
    command line option
--column, 53
--fit-width, 54
--format, 53
--max-width, 54
--noindent, 53
--print-empty, 54
--quote, 53
--sort-ascending, 54
--sort-column, 54
--sort-descending, 54
-c, 53
-f, 53
node, 54
openstack-baremetal-node-vif-attach
    command line option
--port-uuid, 54
--vif-info, 54
node, 54
vif-id, 54
openstack-baremetal-node-vif-detach
    command line option

```

node, 55  
vif-id, 55  
openstack-baremetal-node-vif-list  
    command line option  
    --column, 55  
    --fit-width, 55  
    --format, 55  
    --max-width, 55  
    --noindent, 55  
    --print-empty, 55  
    --quote, 55  
    --sort-ascending, 56  
    --sort-column, 55  
    --sort-descending, 56  
    -c, 55  
    -f, 55  
node, 56  
openstack-baremetal-port-create  
    command line option  
    --column, 56  
    --extra, 57  
    --fit-width, 57  
    --format, 56  
    --is-smartnic, 57  
    --local-link-connection, 57  
    --max-width, 56  
    --node, 57  
    --noindent, 56  
    --physical-network, 57  
    --port-group, 57  
    --prefix, 56  
    --print-empty, 57  
    --pxe-enabled, 57  
    --uuid, 57  
    -c, 56  
    -f, 56  
    -l, 57  
address, 57  
openstack-baremetal-port-delete  
    command line option  
port, 58  
openstack-baremetal-port-group-create  
    command line option  
    --address, 59  
    --column, 58  
    --extra, 59  
    --fit-width, 58  
    --format, 58  
    --max-width, 58  
    --mode, 59  
    --name, 59  
    --node, 59  
    --noindent, 58  
    --prefix, 58  
    --print-empty, 58  
    --property, 59  
    --support-standalone-ports, 59  
    --unsupport-standalone-ports, 59  
    --uuid, 59  
-c, 58  
-f, 58  
openstack-baremetal-port-group-delete  
    command line option  
port, 59  
openstack-baremetal-port-group-list  
    command line option  
    --address, 61  
    --column, 60  
    --fields, 61  
    --fit-width, 60  
    --format, 60  
    --limit, 60  
    --long, 61  
    --marker, 60  
    --max-width, 60  
    --node, 61  
    --noindent, 60  
    --print-empty, 60  
    --quote, 60  
    --sort, 60  
    --sort-ascending, 60  
    --sort-column, 60  
    --sort-descending, 60  
    -c, 60  
    -f, 60  
openstack-baremetal-port-group-set  
    command line option  
    --address, 61  
    --extra, 61  
    --mode, 61  
    --name, 61  
    --node, 61  
    --property, 61  
    --support-standalone-ports, 61  
    --unsupport-standalone-ports, 62  
port, 62  
openstack-baremetal-port-group-show  
    command line option  
    --address, 62  
    --column, 62  
    --fields, 62  
    --fit-width, 62

```

--format, 62
--max-width, 62
--noindent, 62
--prefix, 62
--print-empty, 62
-c, 62
-f, 62
id, 63
openstack-baremetal-port-group-unset
    command line option
--address, 63
--extra, 63
--name, 63
--property, 63
port, 63
openstack-baremetal-port-list command
    line option
--address, 64
--column, 64
--fields, 65
--fit-width, 64
--format, 64
--limit, 64
--long, 65
--marker, 64
--max-width, 64
--node, 64
--noindent, 64
--port-group, 64
--print-empty, 64
--quote, 64
--sort, 65
--sort-ascending, 64
--sort-column, 64
--sort-descending, 64
-c, 64
-f, 64
openstack-baremetal-port-set command
    line option
--address, 65
--extra, 65
--is-smartnic, 66
--local-link-connection, 65
--node, 65
--physical-network, 66
--port-group, 65
--pxe-disabled, 65
--pxe-enabled, 65
port, 66
openstack-baremetal-port-show command
    line option
--address, 66
--column, 66
--fields, 67
--fit-width, 66
--format, 66
--max-width, 66
--noindent, 66
--prefix, 66
--print-empty, 66
-c, 66
-f, 66
id, 67
openstack-baremetal-port-unset command
    line option
--extra, 67
--is-smartnic, 67
--physical-network, 67
--port-group, 67
port, 67
openstack-baremetal-volume-connector-create
    command line option
--column, 68
--connector-id, 68
--extra, 68
--fit-width, 68
--format, 68
--max-width, 68
--node, 68
--noindent, 68
--prefix, 68
--print-empty, 68
--type, 68
--uuid, 68
-c, 68
-f, 68
openstack-baremetal-volume-connector-delete
    command line option
volume, 69
openstack-baremetal-volume-connector-list
    command line option
--column, 69
--fields, 70
--fit-width, 69
--format, 69
--limit, 70
--long, 70
--marker, 70
--max-width, 69
--node, 70
--noindent, 69
--print-empty, 70

```

- quote, 69
- sort, 70
- sort-ascending, 70
- sort-column, 70
- sort-descending, 70
- c, 69
- f, 69
- openstack-baremetal-volume-connector-set
  - command line option
  - connector-id, 71
  - extra, 71
  - node, 70
  - type, 71
  - volume, 71
- openstack-baremetal-volume-connector-show
  - command line option
  - column, 71
  - fields, 72
  - fit-width, 71
  - format, 71
  - max-width, 71
  - noindent, 71
  - prefix, 71
  - print-empty, 71
  - c, 71
  - f, 71
  - id, 72
- openstack-baremetal-volume-connector-unset
  - command line option
  - extra, 72
  - volume, 72
- openstack-baremetal-volume-target-create
  - command line option
  - boot-index, 73
  - column, 73
  - extra, 73
  - fit-width, 73
  - format, 72
  - max-width, 73
  - node, 73
  - noindent, 73
  - prefix, 73
  - print-empty, 73
  - property, 73
  - type, 73
  - uuid, 73
  - volume-id, 73
  - c, 73
  - f, 72
- openstack-baremetal-volume-target-delete
  - command line option
  - volume, 73
- openstack-baremetal-volume-target-list
  - command line option
  - column, 74
  - fields, 75
  - fit-width, 74
  - format, 74
  - limit, 75
  - long, 75
  - marker, 75
  - max-width, 74
  - node, 75
  - noindent, 74
  - print-empty, 74
  - quote, 74
  - sort, 75
  - sort-ascending, 74
  - sort-column, 74
  - sort-descending, 75
  - c, 74
  - f, 74
- openstack-baremetal-volume-target-set
  - command line option
  - boot-index, 75
  - extra, 76
  - node, 75
  - property, 75
  - type, 75
  - volume-id, 76
  - volume, 76
- openstack-baremetal-volume-target-show
  - command line option
  - column, 76
  - fields, 76
  - fit-width, 76
  - format, 76
  - max-width, 76
  - noindent, 76
  - prefix, 76
  - print-empty, 76
  - c, 76
  - f, 76
  - id, 77
- openstack-baremetal-volume-target-unset
  - command line option
  - extra, 77
  - property, 77
  - volume, 77

## P

PassthruCallBaremetalDriver (*class in iron-*

- icclient.osc.v1.baremetal\_driver*), 113
  - PassthruCallBaremetalNode* (class in *ironic-client.osc.v1.baremetal\_node*), 123
  - PassthruListBaremetalDriver* (class in *ironic-client.osc.v1.baremetal\_driver*), 113
  - PassthruListBaremetalNode* (class in *ironic-client.osc.v1.baremetal\_node*), 123
  - PaymentRequired, 93
  - poll()* (in module *ironicclient.common.utils*), 101
  - port
    - openstack-baremetal-port-delete* command line option, 58
    - openstack-baremetal-port-group-delete* command line option, 59
    - openstack-baremetal-port-group-set* command line option, 62
    - openstack-baremetal-port-group-unset* command line option, 63
    - openstack-baremetal-port-set* command line option, 66
    - openstack-baremetal-port-unset* command line option, 67
  - Port* (class in *ironicclient.v1.port*), 164
  - Portgroup* (class in *ironicclient.v1.portgroup*), 166
  - PortgroupManager* (class in *ironic-client.v1.portgroup*), 166
  - PortManager* (class in *ironicclient.v1.port*), 164
  - POWER\_STATE (ironic-client.osc.v1.baremetal\_node.*PowerOffBaremetalNode* attribute), 124
  - POWER\_STATE (ironic-client.osc.v1.baremetal\_node.*PowerOnBaremetalNode* attribute), 125
  - PowerBaremetalNode* (class in *ironic-client.osc.v1.baremetal\_node*), 124
  - PowerOffBaremetalNode* (class in *ironic-client.osc.v1.baremetal\_node*), 124
  - PowerOnBaremetalNode* (class in *ironic-client.osc.v1.baremetal\_node*), 124
  - PreconditionFailed, 93
  - properties()* (ironic-client.v1.driver.*DriverManager* method), 152
  - ProvideBaremetalNode* (class in *ironic-client.osc.v1.baremetal\_node*), 125
  - PROVISION\_STATE (ironic-client.osc.v1.baremetal\_node.*AbortBaremetalNode* attribute), 114
  - PROVISION\_STATE (ironic-client.osc.v1.baremetal\_node.*AdoptBaremetalNode* attribute), 115
  - PROVISION\_STATE (ironic-client.osc.v1.baremetal\_node.*CleanBaremetalNode* attribute), 117
  - PROVISION\_STATE (ironic-client.osc.v1.baremetal\_node.*DeployBaremetalNode* attribute), 119
  - PROVISION\_STATE (ironic-client.osc.v1.baremetal\_node.*InspectBaremetalNode* attribute), 120
  - PROVISION\_STATE (ironic-client.osc.v1.baremetal\_node.*ManageBaremetalNode* attribute), 122
  - PROVISION\_STATE (ironic-client.osc.v1.baremetal\_node.*ProvideBaremetalNode* attribute), 125
  - PROVISION\_STATE (ironic-client.osc.v1.baremetal\_node.*RebuildBaremetalNode* attribute), 126
  - PROVISION\_STATE (ironic-client.osc.v1.baremetal\_node.*RescueBaremetalNode* attribute), 127
  - PROVISION\_STATE (ironic-client.osc.v1.baremetal\_node.*UndeployBaremetalNode* attribute), 129
  - PROVISION\_STATE (ironic-client.osc.v1.baremetal\_node.*UnrescueBaremetalNode* attribute), 129
  - PROVISION\_STATES (ironic-client.osc.v1.baremetal\_node.*ListBaremetalNode* attribute), 120
  - ProvisionStateBaremetalNode* (class in *ironic-client.osc.v1.baremetal\_node*), 125
  - ProvisionStateWithWait* (class in *ironic-client.osc.v1.baremetal\_node*), 125
  - ProxyAuthenticationRequired, 93
  - put()* (*ironicclient.common.apiclient.base.CrudManager* method), 84
- ## R
- raid\_logical\_disk\_properties()* (ironic-client.v1.driver.*DriverManager* method), 152
  - raw\_request()* (ironic-client.common.http.*SessionClient* method), 98
  - RebootBaremetalNode* (class in *ironic-client.osc.v1.baremetal\_node*), 126
  - RebuildBaremetalNode* (class in *ironic-client.osc.v1.baremetal\_node*), 126
  - remove\_all\_traits()* (ironic-

*client.v1.node.NodeManager* method), 159

`remove_trait()` (*ironic-client.v1.node.NodeManager* method), 159

`RemoveTraitBaremetalNode` (class in *ironic-client.osc.v1.baremetal\_node*), 127

`ReplaceLatestVersion` (class in *ironic-client.osc.plugin*), 142

`RequestedRangeNotSatisfiable`, 95

`RequestEntityTooLarge`, 94

`RequestTimeout`, 94

`RequestUriTooLong`, 94

`RescueBaremetalNode` (class in *ironic-client.osc.v1.baremetal\_node*), 127

`Resource` (class in *ironic-client.common.apiclient.base*), 85

`Resource` (class in *ironicclient.common.base*), 97

`Resource` (class in *ironic-client.v1.resource\_fields*), 168

`resource_class` (*ironic-client.common.apiclient.base.BaseManager* attribute), 83

`resource_class` (*ironic-client.common.base.Manager* property), 97

`resource_class` (*ironic-client.v1.allocation.AllocationManager* attribute), 144

`resource_class` (*ironic-client.v1.chassis.ChassisManager* attribute), 146

`resource_class` (*ironic-client.v1.conductor.ConductorManager* attribute), 148

`resource_class` (*ironic-client.v1.deploy\_template.DeployTemplateManager* attribute), 152

`resource_class` (*ironic-client.v1.driver.DriverManager* attribute), 153

`resource_class` (*ironic-client.v1.events.EventManager* attribute), 153

`resource_class` (*ironic-client.v1.node.NodeManager* attribute), 159

`resource_class` (*ironic-client.v1.port.PortManager* attribute), 165

`resource_class` (*ironic-client.v1.portgroup.PortgroupManager* attribute), 168

`resource_class` (*ironic-client.v1.volume\_connector.VolumeConnectorManager* attribute), 171

`resource_class` (*ironic-client.v1.volume\_target.VolumeTargetManager* attribute), 172

`retrieve_data()` (in module *ironic-client.common.filecache*), 98

`run_hooks()` (*ironic-client.common.apiclient.base.HookableMixin* class method), 84

## S

`save_data()` (in module *ironic-client.common.filecache*), 98

`SecurebootOffBaremetalNode` (class in *ironic-client.osc.v1.baremetal\_node*), 127

`SecurebootOnBaremetalNode` (class in *ironic-client.osc.v1.baremetal\_node*), 128

`ServiceUnavailable`, 95

`SessionClient` (class in *ironic-client.common.http*), 98

`set_boot_device()` (*ironic-client.v1.node.NodeManager* method), 159

`set_boot_mode()` (*ironic-client.v1.node.NodeManager* method), 159

`set_console_mode()` (*ironic-client.v1.node.NodeManager* method), 160

`set_loaded()` (*ironic-client.common.apiclient.base.Resource* method), 85

`set_maintenance()` (*ironic-client.v1.node.NodeManager* method), 160

`set_power_state()` (*ironic-client.v1.node.NodeManager* method), 160

`set_provision_state()` (*ironic-client.v1.node.NodeManager* method), 161

`set_secure_boot()` (*ironic-client.v1.node.NodeManager* method), 161

`set_target_raid_config()` (*ironic-client.v1.node.NodeManager* method), 162

<code>set_traits()</code> ( <i>ironic-client.v1.node.NodeManager</i> method), 162	<code>ShowBaremetalVolumeTarget</code> ( <i>class in ironic-client.osc.v1.baremetal_volume_target</i> ), 141
<code>SetBaremetalAllocation</code> ( <i>class in ironic-client.osc.v1.baremetal_allocation</i> ), 103	<code>sort_fields</code> ( <i>ironic-client.v1.resource_fields.Resource</i> property), 170
<code>SetBaremetalChassis</code> ( <i>class in ironic-client.osc.v1.baremetal_chassis</i> ), 106	<code>sort_labels</code> ( <i>ironic-client.v1.resource_fields.Resource</i> property), 170
<code>SetBaremetalDeployTemplate</code> ( <i>class in ironic-client.osc.v1.baremetal_deploy_template</i> ), 110	<code>split_and_deserialize()</code> ( <i>in module ironic-client.common.utils</i> ), 101
<code>SetBaremetalNode</code> ( <i>class in ironic-client.osc.v1.baremetal_node</i> ), 128	<code>start_section()</code> ( <i>ironic-client.common.utils.HelpFormatter</i> method), 99
<code>SetBaremetalPort</code> ( <i>class in ironic-client.osc.v1.baremetal_port</i> ), 132	<code>states()</code> ( <i>ironicclient.v1.node.NodeManager</i> method), 162
<code>SetBaremetalPortGroup</code> ( <i>class in ironic-client.osc.v1.baremetal_portgroup</i> ), 135	<code>StateTransitionFailed</code> , 173
<code>SetBaremetalVolumeConnector</code> ( <i>class in ironic-client.osc.v1.baremetal_volume_connector</i> ), 138	<code>StateTransitionTimeout</code> , 173
<code>SetBaremetalVolumeTarget</code> ( <i>class in ironic-client.osc.v1.baremetal_volume_target</i> ), 140	<code>SUPPORTED_HOOKS</code> ( <i>ironic-client.common.apiclient.base.Extension</i> attribute), 84
<code>setting</code>	<b>T</b>
<code>openstack-baremetal-node-bios-setting-show</code> command line option, 27	<code>take_action()</code> ( <i>ironic-client.osc.v1.baremetal_allocation.CreateBaremetalAllocation</i> method), 102
<code>ShowBaremetalAllocation</code> ( <i>class in ironic-client.osc.v1.baremetal_allocation</i> ), 104	<code>take_action()</code> ( <i>ironic-client.osc.v1.baremetal_allocation.DeleteBaremetalAllocation</i> method), 103
<code>ShowBaremetalChassis</code> ( <i>class in ironic-client.osc.v1.baremetal_chassis</i> ), 106	<code>take_action()</code> ( <i>ironic-client.osc.v1.baremetal_allocation.ListBaremetalAllocation</i> method), 103
<code>ShowBaremetalConductor</code> ( <i>class in ironic-client.osc.v1.baremetal_conductor</i> ), 107	<code>take_action()</code> ( <i>ironic-client.osc.v1.baremetal_allocation.SetBaremetalAllocation</i> method), 103
<code>ShowBaremetalDeployTemplate</code> ( <i>class in ironic-client.osc.v1.baremetal_deploy_template</i> ), 110	<code>take_action()</code> ( <i>ironic-client.osc.v1.baremetal_allocation.ShowBaremetalAllocation</i> method), 104
<code>ShowBaremetalDriver</code> ( <i>class in ironic-client.osc.v1.baremetal_driver</i> ), 114	<code>take_action()</code> ( <i>ironic-client.osc.v1.baremetal_allocation.UnsetBaremetalAllocation</i> method), 104
<code>ShowBaremetalNode</code> ( <i>class in ironic-client.osc.v1.baremetal_node</i> ), 128	<code>take_action()</code> ( <i>ironic-client.osc.v1.baremetal_chassis.CreateBaremetalChassis</i> method), 105
<code>ShowBaremetalPort</code> ( <i>class in ironic-client.osc.v1.baremetal_port</i> ), 133	<code>take_action()</code> ( <i>ironic-client.osc.v1.baremetal_chassis.DeleteBaremetalChassis</i> method), 105
<code>ShowBaremetalPortGroup</code> ( <i>class in ironic-client.osc.v1.baremetal_portgroup</i> ), 135	<code>take_action()</code> ( <i>ironic-client.osc.v1.baremetal_chassis.ListBaremetalChassis</i> method), 106
<code>ShowBaremetalVolumeConnector</code> ( <i>class in ironic-client.osc.v1.baremetal_volume_connector</i> ),	

take\_action() (ironic- take\_action() (ironic-  
client.osc.v1.baremetal\_chassis.SetBaremetalChassisclient.osc.v1.baremetal\_driver.ShowBaremetalDriver  
method), 106 method), 114

take\_action() (ironic- take\_action() (ironic-  
client.osc.v1.baremetal\_chassis.ShowBaremetalChassisclient.osc.v1.baremetal\_node.AddTraitBaremetalNode  
method), 106 method), 115

take\_action() (ironic- take\_action() (ironic-  
client.osc.v1.baremetal\_chassis.UnsetBaremetalChassisclient.osc.v1.baremetal\_node.BIOSSettingShowBaremetal  
method), 107 method), 116

take\_action() (ironic- take\_action() (ironic-  
client.osc.v1.baremetal\_conductor.ListBaremetalConductorsclient.osc.v1.baremetal\_node.BootdeviceSetBaremetalNode  
method), 107 method), 116

take\_action() (ironic- take\_action() (ironic-  
client.osc.v1.baremetal\_conductor.ShowBaremetalConductorclient.osc.v1.baremetal\_node.BootdeviceShowBaremetalNode  
method), 108 method), 116

take\_action() (ironic- take\_action() (ironic-  
client.osc.v1.baremetal\_create.CreateBaremetalclient.osc.v1.baremetal\_node.BootmodeSetBaremetalNode  
method), 108 method), 117

take\_action() (ironic- take\_action() (ironic-  
client.osc.v1.baremetal\_deploy\_template.CreateBaremetalDeployTemplateclient.osc.v1.baremetal\_node.ConsoleDisableBaremetalNode  
method), 109 method), 117

take\_action() (ironic- take\_action() (ironic-  
client.osc.v1.baremetal\_deploy\_template.DeleteBaremetalDeployTemplateclient.osc.v1.baremetal\_node.ConsoleEnableBaremetalNode  
method), 109 method), 118

take\_action() (ironic- take\_action() (ironic-  
client.osc.v1.baremetal\_deploy\_template.ListBaremetalDeployTemplatesclient.osc.v1.baremetal\_node.ConsoleShowBaremetalNode  
method), 110 method), 118

take\_action() (ironic- take\_action() (ironic-  
client.osc.v1.baremetal\_deploy\_template.SetBaremetalDeployTemplateclient.osc.v1.baremetal\_node.CreateBaremetalNode  
method), 110 method), 119

take\_action() (ironic- take\_action() (ironic-  
client.osc.v1.baremetal\_deploy\_template.ShowBaremetalDeployTemplateclient.osc.v1.baremetal\_node.DeleteBaremetalNode  
method), 111 method), 119

take\_action() (ironic- take\_action() (ironic-  
client.osc.v1.baremetal\_deploy\_template.UnsetBaremetalDeployTemplateclient.osc.v1.baremetal\_node.InjectNmiBaremetalNode  
method), 111 method), 120

take\_action() (ironic- take\_action() (ironic-  
client.osc.v1.baremetal\_driver.ListBaremetalDriverclient.osc.v1.baremetal\_node.ListBaremetalNode  
method), 112 method), 121

take\_action() (ironic- take\_action() (ironic-  
client.osc.v1.baremetal\_driver.ListBaremetalDriverPartitionsclient.osc.v1.baremetal\_node.ListBIOSSettingBaremetalNode  
method), 112 method), 120

take\_action() (ironic- take\_action() (ironic-  
client.osc.v1.baremetal\_driver.ListBaremetalDriverPropertiesclient.osc.v1.baremetal\_node.ListTraitsBaremetalNode  
method), 113 method), 121

take\_action() (ironic- take\_action() (ironic-  
client.osc.v1.baremetal\_driver.PassthruCallBaremetalDriverclient.osc.v1.baremetal\_node.MaintenanceSetBaremetalNode  
method), 113 method), 121

take\_action() (ironic- take\_action() (ironic-  
client.osc.v1.baremetal\_driver.PassthruListBaremetalDriverclient.osc.v1.baremetal\_node.MaintenanceUnsetBaremetalNode  
method), 114 method), 122



<code>take_action()</code>	( <i>ironic-</i> <code>take_action()</code>	( <i>ironic-</i>
<i>client.osc.v1.baremetal_node.NodeHistoryEventGet</i>	<i>client.osc.v1.baremetal_node.VifListBaremetalNode</i>	
<i>method</i> ), 123	<i>method</i> ), 131	
<code>take_action()</code>	( <i>ironic-</i> <code>take_action()</code>	( <i>ironic-</i>
<i>client.osc.v1.baremetal_node.NodeHistoryList</i>	<i>client.osc.v1.baremetal_port.CreateBaremetalPort</i>	
<i>method</i> ), 123	<i>method</i> ), 131	
<code>take_action()</code>	( <i>ironic-</i> <code>take_action()</code>	( <i>ironic-</i>
<i>client.osc.v1.baremetal_node.PassthruCallBaremetalNode</i>	<i>client.osc.v1.baremetal_port.DeleteBaremetalPort</i>	
<i>method</i> ), 123	<i>method</i> ), 132	
<code>take_action()</code>	( <i>ironic-</i> <code>take_action()</code>	( <i>ironic-</i>
<i>client.osc.v1.baremetal_node.PassthruListBaremetalNode</i>	<i>client.osc.v1.baremetal_port.ListBaremetalPort</i>	
<i>method</i> ), 124	<i>method</i> ), 132	
<code>take_action()</code>	( <i>ironic-</i> <code>take_action()</code>	( <i>ironic-</i>
<i>client.osc.v1.baremetal_node.PowerBaremetalNode</i>	<i>client.osc.v1.baremetal_port.SetBaremetalPort</i>	
<i>method</i> ), 124	<i>method</i> ), 133	
<code>take_action()</code>	( <i>ironic-</i> <code>take_action()</code>	( <i>ironic-</i>
<i>client.osc.v1.baremetal_node.ProvisionStateBaremetalNode</i>	<i>client.osc.v1.baremetal_port.ShowBaremetalPort</i>	
<i>method</i> ), 125	<i>method</i> ), 133	
<code>take_action()</code>	( <i>ironic-</i> <code>take_action()</code>	( <i>ironic-</i>
<i>client.osc.v1.baremetal_node.ProvisionStateWithWait</i>	<i>client.osc.v1.baremetal_port.UnsetBaremetalPort</i>	
<i>method</i> ), 126	<i>method</i> ), 133	
<code>take_action()</code>	( <i>ironic-</i> <code>take_action()</code>	( <i>ironic-</i>
<i>client.osc.v1.baremetal_node.RebootBaremetalNode</i>	<i>client.osc.v1.baremetal_portgroup.CreateBaremetalPortG</i>	
<i>method</i> ), 126	<i>method</i> ), 134	
<code>take_action()</code>	( <i>ironic-</i> <code>take_action()</code>	( <i>ironic-</i>
<i>client.osc.v1.baremetal_node.RemoveTraitBaremetalNode</i>	<i>client.osc.v1.baremetal_portgroup.DeleteBaremetalPortG</i>	
<i>method</i> ), 127	<i>method</i> ), 134	
<code>take_action()</code>	( <i>ironic-</i> <code>take_action()</code>	( <i>ironic-</i>
<i>client.osc.v1.baremetal_node.SecurebootOffBaremetalNode</i>	<i>client.osc.v1.baremetal_portgroup.ListBaremetalPortGrou</i>	
<i>method</i> ), 128	<i>method</i> ), 135	
<code>take_action()</code>	( <i>ironic-</i> <code>take_action()</code>	( <i>ironic-</i>
<i>client.osc.v1.baremetal_node.SecurebootOnBaremetalNode</i>	<i>client.osc.v1.baremetal_portgroup.SetBaremetalPortGrou</i>	
<i>method</i> ), 128	<i>method</i> ), 135	
<code>take_action()</code>	( <i>ironic-</i> <code>take_action()</code>	( <i>ironic-</i>
<i>client.osc.v1.baremetal_node.SetBaremetalNode</i>	<i>client.osc.v1.baremetal_portgroup.ShowBaremetalPortGrou</i>	
<i>method</i> ), 128	<i>method</i> ), 136	
<code>take_action()</code>	( <i>ironic-</i> <code>take_action()</code>	( <i>ironic-</i>
<i>client.osc.v1.baremetal_node.ShowBaremetalNode</i>	<i>client.osc.v1.baremetal_portgroup.UnsetBaremetalPortGrou</i>	
<i>method</i> ), 129	<i>method</i> ), 136	
<code>take_action()</code>	( <i>ironic-</i> <code>take_action()</code>	( <i>ironic-</i>
<i>client.osc.v1.baremetal_node.UnsetBaremetalNode</i>	<i>client.osc.v1.baremetal_volume_connector.CreateBaremet</i>	
<i>method</i> ), 130	<i>method</i> ), 137	
<code>take_action()</code>	( <i>ironic-</i> <code>take_action()</code>	( <i>ironic-</i>
<i>client.osc.v1.baremetal_node.ValidateBaremetalNode</i>	<i>client.osc.v1.baremetal_volume_connector.DeleteBaremet</i>	
<i>method</i> ), 130	<i>method</i> ), 137	
<code>take_action()</code>	( <i>ironic-</i> <code>take_action()</code>	( <i>ironic-</i>
<i>client.osc.v1.baremetal_node.VifAttachBaremetalNode</i>	<i>client.osc.v1.baremetal_volume_connector.ListBaremetalV</i>	
<i>method</i> ), 130	<i>method</i> ), 137	
<code>take_action()</code>	( <i>ironic-</i> <code>take_action()</code>	( <i>ironic-</i>
<i>client.osc.v1.baremetal_node.VifDetachBaremetalNode</i>	<i>client.osc.v1.baremetal_volume_connector.SetBaremetalV</i>	
<i>method</i> ), 131	<i>method</i> ), 138	

take\_action() (ironic- 111  
*client.osc.v1.baremetal\_volume\_connector.UnsetBaremetalVolumeConnector* (class in ironic-  
*method*), 138 *client.osc.v1.baremetal\_node*), 129

take\_action() (ironic- UnsetBaremetalPort (class in ironic-  
*client.osc.v1.baremetal\_volume\_connector.UnsetBaremetalVolumeConnector*), 133  
*method*), 139 *UnsetBaremetalPortGroup* (class in ironic-  
*UnsetBaremetalVolumeConnector*  
*method*), 139 (class in ironic-  
*client.osc.v1.baremetal\_portgroup*), 136

take\_action() (ironic- *client.osc.v1.baremetal\_volume\_connector*),  
*client.osc.v1.baremetal\_volume\_target.DeleteBaremetalVolumeTarget*  
*method*), 140 *UnsetBaremetalVolumeTarget* (class in ironic-  
*UnsetBaremetalVolumeTarget* (class in ironic-  
*method*), 140 *client.osc.v1.baremetal\_volume\_target*),

take\_action() (ironic- *client.osc.v1.baremetal\_volume\_target*,  
*client.osc.v1.baremetal\_volume\_target.ListBaremetalVolumeTarget*  
*method*), 140 *UnsupportedMediaType*, 96

take\_action() (ironic- *UnsupportedVersion*, 96  
*client.osc.v1.baremetal\_volume\_target.SetBaremetalVolumeTarget* (ironic-  
*method*), 141 *client.common.apiclient.base.CrudManager*  
*method*), 84

take\_action() (ironic- *client.osc.v1.baremetal\_volume\_target.ShowBaremetalVolumeTarget* (ironic-  
*method*), 141 *client.v1.allocation.AllocationManager*  
*method*), 144

take\_action() (ironic- *client.osc.v1.baremetal\_volume\_target.UnsetBaremetalVolumeTarget* (ironic-  
*method*), 142 *client.v1.chassis.ChassisManager*  
*method*), 147

tempdir() (in module *ironicclient.common.utils*), 101

to\_dict() (ironic- *client.v1.deploy\_template.DeployTemplateManager*  
*client.common.apiclient.base.Resource*  
*method*), 85 *method*), 152

to\_dict() (*ironicclient.common.base.Resource*  
*method*), 97 *update()* (*ironicclient.v1.driver.DriverManager*  
*method*), 153

trait *update()* (*ironicclient.v1.node.NodeManager*  
*method*), 162

openstack-baremetal-node-add-trait *update()* (*ironicclient.v1.port.PortManager*  
*command line option*, 24 *method*), 165

openstack-baremetal-node-remove-trait *update()* (ironic-  
*command line option*, 44 *client.v1.portgroup.PortgroupManager*  
*method*), 168

U *update()* (ironic-  
*client.v1.volume\_connector.VolumeConnectorManager*  
*method*), 171

Unauthorized, 95 *update()* (ironic-  
*client.v1.volume\_target.VolumeTargetManager*  
*method*), 172

UndeployBaremetalNode (class in *ironic-*  
*client.osc.v1.baremetal\_node*), 129

UnprocessableEntity, 96

UnrescueBaremetalNode (class in *ironic-*  
*client.osc.v1.baremetal\_node*), 129

UnsetBaremetalAllocation (class in *ironic-*  
*client.osc.v1.baremetal\_allocation*), 104

UnsetBaremetalChassis (class in *ironic-*  
*client.osc.v1.baremetal\_chassis*), 106

UnsetBaremetalDeployTemplate (class in *ironic-*  
*client.osc.v1.baremetal\_deploy\_template*),

V

validate() (*ironicclient.v1.node.NodeManager*  
*method*), 162

ValidateBaremetalNode (class in *ironic-*  
*client.osc.v1.baremetal\_node*), 130

ValidationError, 96

vendor\_passthru() (ironic-

`client.v1.driver.DriverManager` method), 153

`wait_for_provision_state()` (`ironic-client.v1.node.NodeManager` method), 164

`vendor_passthru()` (`ironic-client.v1.node.NodeManager` method), 163

`with_retries()` (in module `ironic-client.common.http`), 98

`VersionNegotiationMixin` (class in `ironic-client.common.http`), 98

`vif_attach()` (`ironic-client.v1.node.NodeManager` method), 163

`vif_detach()` (`ironic-client.v1.node.NodeManager` method), 163

`vif_list()` (`ironicclient.v1.node.NodeManager` method), 163

`vif-id`

- `openstack-baremetal-node-vif-attach` command line option, 54
- `openstack-baremetal-node-vif-detach` command line option, 55

`VifAttachBaremetalNode` (class in `ironic-client.osc.v1.baremetal_node`), 130

`VifDetachBaremetalNode` (class in `ironic-client.osc.v1.baremetal_node`), 130

`VifListBaremetalNode` (class in `ironic-client.osc.v1.baremetal_node`), 131

`volume`

- `openstack-baremetal-volume-connector-delete` command line option, 69
- `openstack-baremetal-volume-connector-set` command line option, 71
- `openstack-baremetal-volume-connector-unset` command line option, 72
- `openstack-baremetal-volume-target-delete` command line option, 73
- `openstack-baremetal-volume-target-set` command line option, 76
- `openstack-baremetal-volume-target-unset` command line option, 77

`VolumeConnector` (class in `ironic-client.v1.volume_connector`), 170

`VolumeConnectorManager` (class in `ironic-client.v1.volume_connector`), 170

`VolumeTarget` (class in `ironic-client.v1.volume_target`), 171

`VolumeTargetManager` (class in `ironic-client.v1.volume_target`), 171

## W

`wait()` (`ironicclient.v1.allocation.AllocationManager` method), 144