

---

# Python Digitalocean Documentation

*Release 1.11*

**Lorenzo Setale**

**Apr 16, 2017**



---

## Contents:

---

<b>1</b>	<b>digitalocean package</b>	<b>1</b>
1.1	Submodules	1
1.2	digitalocean.Account module	1
1.3	digitalocean.Action module	1
1.4	digitalocean.Domain module	2
1.5	digitalocean.Droplet module	2
1.6	digitalocean.FloatingIP module	6
1.7	digitalocean.Image module	7
1.8	digitalocean.Kernel module	7
1.9	digitalocean.LoadBalancer module	7
1.10	digitalocean.Manager module	10
1.11	digitalocean.Metadata module	12
1.12	digitalocean.Record module	12
1.13	digitalocean.Region module	12
1.14	digitalocean.SSHKey module	12
1.15	digitalocean.Size module	13
1.16	digitalocean.Tag module	13
1.17	digitalocean.Volume module	13
1.18	digitalocean.baseapi module	14
1.19	Module contents	15
<b>2</b>	<b>Indices and tables</b>	<b>17</b>
	<b>Python Module Index</b>	<b>19</b>



## Submodules

### digitalocean.Account module

```
class digitalocean.Account.Account (*args, **kwargs)
    Bases: digitalocean.baseapi.BaseAPI
    classmethod get_object (api_token)
        Class method that will return an Account object.
    load ()
```

### digitalocean.Action module

```
class digitalocean.Action.Action (*args, **kwargs)
    Bases: digitalocean.baseapi.BaseAPI
    classmethod get_object (api_token, action_id)
        Class method that will return a Action object by ID.
    load ()
    load_directly ()
    wait (update_every_seconds=1)
        Wait until the action is marked as completed or with an error. It will return True in case of success,
        otherwise False.
    Optional Args:
        update_every_seconds - int [number of seconds to wait before] checking if the action is completed.
```

## digitalocean.Domain module

**class** digitalocean.Domain.**Domain** (\*args, \*\*kwargs)

Bases: *digitalocean.baseapi.BaseAPI*

**create** ()

Create new doamin

**create\_new\_domain\_record** (\*args, \*\*kwargs)

Create new domain record. <https://developers.digitalocean.com/#create-a-new-domain-record>

### Parameters

- **type** – The record type (A, MX, CNAME, etc).
- **name** – The host name, alias, or service being defined by the record
- **data** – Variable data depending on record type.

**Optional Args:** **priority:** The priority of the host port: The port that the service is accessible on **weight:** The weight of records with the same priority

**destroy** ()

Destroy the domain by name

**classmethod** **get\_object** (api\_token, domain\_name)

Class method that will return a Domain object by ID.

**get\_records** (params=None)

Returns a list of Record objects

**load** ()

## digitalocean.Droplet module

**exception** digitalocean.Droplet.**BadKernelObject**

Bases: *digitalocean.Droplet.DropletError*

**exception** digitalocean.Droplet.**BadSSHKeyFormat**

Bases: *digitalocean.Droplet.DropletError*

**class** digitalocean.Droplet.**Droplet** (\*args, \*\*kwargs)

Bases: *digitalocean.baseapi.BaseAPI*

“Droplet management

Attributes accepted at creation time:

### Parameters

- **name** (*str*) – name
- **size\_slug** (*str*) – droplet size
- **image** (*str*) – image name to use to create droplet
- **region** (*str*) – region
- **ssh\_keys** – (*str*, optional): list of ssh keys
- **backups** (*bool*) – True if backups enabled

- **ipv6** (*bool*) – True if ipv6 enabled
- **private\_networking** (*bool*) – True if private networking enabled
- **user\_data** (*str*) – arbitrary data to pass to droplet
- **volumes** (*str*, optional) – list of blockstorage volumes
- **monitoring** – (*bool*) - True if installing the DigitalOcean monitoring agent

**Attributes returned by API:** **id** (*int*): droplet id **memory** (*str*): memory size **vcpus** (*int*): number of vcpus **disk** (*int*): disk size in GB **status** (*str*): status **locked** (*bool*): True if locked **created\_at** (*str*): creation date in format u'2014-11-06T10:42:09Z' **status** (*str*): status, e.g. 'new', 'active', etc **networks** (*dict*): details of connected networks **kernel** (*dict*): details of kernel **backup\_ids** (*int*, optional): list of ids of backups of this droplet **snapshot\_ids** (*int*, optional): list of ids of snapshots of this droplet **action\_ids** (*int*, optional): list of ids of actions **features** (*str*, optional): list of enabled features. e.g.

```
[u'private_networking', u'virtio']
```

**image** (*dict*): details of image used to create this droplet **ip\_address** (*str*): public ip addresses **private\_ip\_address** (*str*): private ip address **ip\_v6\_address** (*str*, optional): list of ipv6 addresses assigned **end\_point** (*str*): url of api endpoint used **volume\_ids** (*str*, optional): list of blockstorage volumes

**change\_kernel** (*kernel*, *return\_dict=True*)

Change the kernel to a new one

**Parameters** **kernel** – instance of digitalocean.Kernel.Kernel

**Optional Args:**

**return\_dict** (*bool*): **Return a dict when True (default)**, otherwise return an Action.

Returns dict or Action

**create** (*\*args*, *\*\*kwargs*)

Create the droplet with object properties.

Note: Every argument and parameter given to this method will be assigned to the object.

**classmethod create\_multiple** (*\*args*, *\*\*kwargs*)

**destroy** ()

Destroy the droplet

Returns dict

**disable\_backups** (*return\_dict=True*)

Disable automatic backups

**Optional Args:**

**return\_dict** (*bool*): **Return a dict when True (default)**, otherwise return an Action.

Returns dict or Action

**enable\_backups** (*return\_dict=True*)

Enable automatic backups

**Optional Args:**

**return\_dict** (*bool*): **Return a dict when True (default)**, otherwise return an Action.

Returns dict or Action

**enable\_ipv6** (*return\_dict=True*)

Enable IPv6 on an existing Droplet where available.

**Optional Args:**

**return\_dict (bool): Return a dict when True (default),** otherwise return an Action.

Returns dict or Action

**enable\_private\_networking** (*return\_dict=True*)

Enable private networking on an existing Droplet where available.

**Optional Args:**

**return\_dict (bool): Return a dict when True (default),** otherwise return an Action.

Returns dict or Action

**get\_action** (*action\_id*)

Returns a specific Action by its ID.

**Parameters** **action\_id** (*int*) – id of action

**get\_actions** ()

Returns a list of Action objects This actions can be used to check the droplet's status

**get\_data** (*\*args, \*\*kwargs*)

Customized version of get\_data to perform `__check_actions_in_data`

**get\_events** ()

A helper function for backwards compatability. Calls `get_actions()`

**get\_kernel\_available** ()

Get a list of kernels available

**classmethod get\_object** (*api\_token, droplet\_id*)

Class method that will return a Droplet object by ID.

**Parameters**

- **api\_token** (*str*) – token
- **droplet\_id** (*int*) – droplet id

**get\_snapshots** ()

This method will return the snapshots/images connected to that specific droplet.

**load** ()

Fetch data about droplet - use this instead of `get_data()`

**power\_cycle** (*return\_dict=True*)

restart the droplet

**Optional Args:**

**return\_dict (bool): Return a dict when True (default),** otherwise return an Action.

Returns dict or Action

**power\_off** (*return\_dict=True*)

restart the droplet

**Optional Args:**

**return\_dict (bool): Return a dict when True (default),** otherwise return an Action.

Returns dict or Action



**power\_on** (*return\_dict=True*)

Boot up the droplet

**Optional Args:**

**return\_dict (bool): Return a dict when True (default),** otherwise return an Action.

Returns dict or Action

**reboot** (*return\_dict=True*)

restart the droplet

**Optional Args:**

**return\_dict (bool): Return a dict when True (default),** otherwise return an Action.

Returns dict or Action

**rebuild** (*image\_id=None, return\_dict=True*)

Restore the droplet to an image ( snapshot or backup )

**Parameters** **image\_id** (*int*) – id of image

**Optional Args:**

**return\_dict (bool): Return a dict when True (default),** otherwise return an Action.

Returns dict or Action

**rename** (*name, return\_dict=True*)

Rename the droplet

**Parameters** **name** (*str*) – new name

**Optional Args:**

**return\_dict (bool): Return a dict when True (default),** otherwise return an Action.

Returns dict or Action

**reset\_root\_password** (*return\_dict=True*)

reset the root password

**Optional Args:**

**return\_dict (bool): Return a dict when True (default),** otherwise return an Action.

Returns dict or Action

**resize** (*new\_size\_slug, return\_dict=True, disk=True*)

Resize the droplet to a new size slug. <https://developers.digitalocean.com/documentation/v2/#resize-a-droplet>

**Parameters** **new\_size\_slug** (*str*) – name of new size

**Optional Args:**

**return\_dict (bool): Return a dict when True (default),** otherwise return an Action.

**disk (bool):** If a permanent resize, with disk changes included.

Returns dict or Action

**restore** (*image\_id, return\_dict=True*)

Restore the droplet to an image ( snapshot or backup )

**Parameters** `image_id` (*int*) – id of image

**Optional Args:**

**return\_dict** (**bool**): **Return a dict when True (default)**, otherwise return an Action.

Returns dict or Action

**shutdown** (*return\_dict=True*)

shutdown the droplet

**Optional Args:**

**return\_dict** (**bool**): **Return a dict when True (default)**, otherwise return an Action.

Returns dict or Action

**take\_snapshot** (*snapshot\_name, return\_dict=True, power\_off=False*)

Take a snapshot!

**Parameters** `snapshot_name` (*str*) – name of snapshot

**Optional Args:**

**return\_dict** (**bool**): **Return a dict when True (default)**, otherwise return an Action.

**power\_off** (**bool**): **Before taking the snapshot the droplet will be** turned off with another API call.  
It will wait until the droplet will be powered off.

Returns dict or Action

**exception** `digitalocean.Droplet.DropletError`

Bases: `digitalocean.baseapi.Error`

Base exception class for this module

## digitalocean.FloatingIP module

**class** `digitalocean.FloatingIP.FloatingIP` (*\*args, \*\*kwargs*)

Bases: `digitalocean.baseapi.BaseAPI`

**assign** (*droplet\_id*)

Assign a FloatingIP to a Droplet.

**Parameters** `droplet_id` – int - droplet id

**create** (*\*args, \*\*kwargs*)

Creates a FloatingIP and assigns it to a Droplet.

Note: Every argument and parameter given to this method will be assigned to the object.

**Parameters** `droplet_id` – int - droplet id

**destroy** ()

Destroy the FloatingIP

**classmethod** `get_object` (*api\_token, ip*)

Class method that will return a FloatingIP object by its IP.

**Parameters**

- `api_token` – str - token

- **ip** – str - floating ip address

**load()**

Load the FloatingIP object from DigitalOcean.

Requires self.ip to be set.

**reserve** (\*args, \*\*kwargs)

Creates a FloatingIP in a region without assigning it to a specific Droplet.

Note: Every argument and parameter given to this method will be assigned to the object.

**Parameters** **region\_slug** – str - region’s slug (e.g. ‘nyc3’)

**unassign()**

Unassign a FloatingIP.

## digitalocean.Image module

**class** digitalocean.Image.**Image** (\*args, \*\*kwargs)

Bases: *digitalocean.baseapi.BaseAPI*

**destroy()**

Destroy the image

**classmethod** **get\_object** (api\_token, image\_id)

Class method that will return an Image object by ID.

**load()**

**rename** (new\_name)

Rename an image

**transfer** (new\_region\_slug)

Transfer the image

## digitalocean.Kernel module

**class** digitalocean.Kernel.**Kernel** (\*args, \*\*kwargs)

Bases: *digitalocean.baseapi.BaseAPI*

## digitalocean.LoadBalancer module

**class** digitalocean.LoadBalancer.**ForwardingRule** (entry\_protocol=None, entry\_port=None, target\_protocol=None, target\_port=None, certificate\_id='', tls\_passthrough=False)

Bases: *object*

An object holding information about a LoadBalancer forwarding rule setting.

**Parameters**

- **entry\_protocol** (*str*) – The protocol used for traffic to a LoadBalancer. The possible values are: “http”, “https”, or “tcp”
- **entry\_port** (*int*) – The port the LoadBalancer instance will listen on

- **target\_protocol** (*str*) – The protocol used for traffic from a LoadBalancer to the backend Droplets. The possible values are: “http”, “https”, or “tcp”
- **target\_port** (*int*) – The port on the backend Droplets on which the LoadBalancer will send traffic
- **certificate\_id** (*str, optional*) – The ID of the TLS certificate used for SSL termination if enabled
- **tls\_passthrough** (*bool, optional*) – A boolean indicating if SSL encrypted traffic will be passed through to the backend Droplets

```
class digitalocean.LoadBalancer.HealthCheck (protocol='http', port=80, path='/',
                                             check_interval_seconds=10, response_timeout_seconds=5,
                                             healthy_threshold=5, unhealthy_threshold=3)
```

Bases: `object`

An object holding information about a LoadBalancer health check settings.

#### Parameters

- **protocol** (*str*) – The protocol used for health checks. The possible values are “http” or “tcp”.
- **port** (*int*) – The port on the backend Droplets for health checks
- **path** (*str*) – The path to send a health check request to
- **check\_interval\_seconds** (*int*) – The number of seconds between between two consecutive health checks
- **response\_timeout\_seconds** (*int*) – The number of seconds the Load Balancer instance will wait for a response until marking a check as failed
- **healthy\_threshold** (*int*) – The number of times a health check must fail for a backend Droplet to be removed from the pool
- **unhealthy\_threshold** (*int*) – The number of times a health check must pass for a backend Droplet to be re-added to the pool

```
class digitalocean.LoadBalancer.LoadBalancer (*args, **kwargs)
```

Bases: `digitalocean.baseapi.BaseAPI`

An object representing an DigitalOcean Load Balancer.

Attributes accepted at creation time:

**Args:** **name** (*str*): The Load Balancer’s name **region** (*str*): The slug identifier for a DigitalOcean region **algorithm** (*str, optional*): The load balancing algorithm to be used. Currently, it must be either “round\_robin” or “least\_connections”

**forwarding\_rules** (*obj:list*): A list of *ForwardingRules* objects **health\_check** (*obj, optional*): A *HealthCheck* object **sticky\_sessions** (*obj, optional*): A *StickySessions* object **redirect\_http\_to\_https** (*bool, optional*): A boolean indicating

whether HTTP requests to the Load Balancer should be redirected to HTTPS

**droplet\_ids** (*obj:list of int*): A list of IDs representing Droplets to be added to the Load Balancer (mutually exclusive with ‘tag’)

**tag** (*str*): A string representing a DigitalOcean Droplet tag (mutually exclusive with ‘droplet\_ids’)

**Attributes returned by API:** `name` (`str`): The Load Balancer’s name `id` (`str`): An unique identifier for a Load-Balancer `ip` (`str`): Public IP address for a LoadBalancer `region` (`str`): The slug identifier for a DigitalOcean region `algorithm` (`str`, optional): The load balancing algorithm to be

used. Currently, it must be either “round\_robin” or “least\_connections”

`forwarding_rules` (`obj:list`): A list of *ForwrdingRules* objects `health_check` (`obj`, optional): A *HealthCheck* object `sticky_sessions` (`obj`, optional): A *StickySessions* object `redirect_http_to_https` (`bool`, optional): A boolean indicating

whether HTTP requests to the Load Balancer should be redirected to HTTPS

**droplet\_ids** (`obj:list of int`): A list of IDs representing Droplets to be added to the Load Balancer

`tag` (`str`): A string representing a DigitalOcean Droplet tag `status` (`string`): An indication the current state of the LoadBalancer `created_at` (`str`): The date and time when the LoadBalancer was created

**add\_droplets** (`droplet_ids`)

Assign a LoadBalancer to a Droplet.

**Parameters** (`obj` (`droplet_ids`) – list of `int`): A list of Droplet IDs

**add\_forwarding\_rules** (`forwarding_rules`)

Adds new forwarding rules to a LoadBalancer.

**Parameters** (`obj` (`forwarding_rules`) – list): A list of *ForwrdingRules* objects

**create** (`*args`, `**kwargs`)

Creates a new LoadBalancer.

Note: Every argument and parameter given to this method will be assigned to the object.

#### Parameters

- **name** (`str`) – The Load Balancer’s name
- **region** (`str`) – The slug identifier for a DigitalOcean region
- **algorithm** (`str`, `optional`) – The load balancing algorithm to be used. Currently, it must be either “round\_robin” or “least\_connections”
- (`obj` (`droplet_ids`) – list): A list of *ForwrdingRules* objects
- **health\_check** (`obj`, `optional`) – A *HealthCheck* object
- **sticky\_sessions** (`obj`, `optional`) – A *StickySessions* object
- **redirect\_http\_to\_https** (`bool`, `optional`) – A boolean indicating whether HTTP requests to the Load Balancer should be redirected to HTTPS
- (`obj` – list of `int`): A list of IDs representing Droplets to be added to the Load Balancer (mutually exclusive with ‘tag’)
- **tag** (`str`) – A string representing a DigitalOcean Droplet tag (mutually exclusive with ‘droplet\_ids’)

**destroy** ()

Destroy the LoadBalancer

**classmethod get\_object** (`api_token`, `id`)

Class method that will return a LoadBalancer object by its ID.

#### Parameters

- **api\_token** (`str`) – DigitalOcean API token

- **id** (*str*) – Load Balancer ID

**load** ()

Loads updated attributes for a LoadBalancer object.

Requires self.id to be set.

**remove\_droplets** (*droplet\_ids*)

Unassign a LoadBalancer.

**Parameters** (**obj** (*droplet\_ids*) – list of int): A list of Droplet IDs

**remove\_forwarding\_rules** (*forwarding\_rules*)

Removes existing forwarding rules from a LoadBalancer.

**Parameters** (**obj** (*forwarding\_rules*) – list): A list of *ForwardingRules* objects

**class** digitalocean.LoadBalancer.**StickySessions** (*type='none', cookie\_name='DO\_LB', cookie\_ttl\_seconds=300*)

Bases: `object`

An object holding information on a LoadBalancer's sticky sessions settings.

**Parameters**

- **type** (*str*) – The type of sticky sessions used. Can be “cookies” or “none”
- **cookie\_name** (*str, optional*) – The name used for the client cookie when using cookies for sticky session
- **cookie\_ttl\_seconds** (*int, optional*) – The number of seconds until the cookie expires

## digitalocean.Manager module

**class** digitalocean.Manager.**Manager** (*\*args, \*\*kwargs*)

Bases: `digitalocean.baseapi.BaseAPI`

**get\_account** ()

Returns an Account object.

**get\_action** (*action\_id*)

Return an Action object by a specific ID.

**get\_all\_certificates** ()

This function returns a list of Certificate objects.

**get\_all\_domains** ()

This function returns a list of Domain object.

**get\_all\_droplets** (*tag\_name=None*)

This function returns a list of Droplet object.

**get\_all\_floating\_ips** ()

This function returns a list of FloatingIP objects.

**get\_all\_images** ()

This function returns a list of Image objects containing all available DigitalOcean images, both public and private.

**get\_all\_load\_balancers** ()

Returns a list of Load Balancer objects.

**get\_all\_regions** ()

This function returns a list of Region object.

**get\_all\_sizes** ()

This function returns a list of Size object.

**get\_all\_sshkeys** ()

This function returns a list of SSHKey object.

**get\_all\_volumes** ()

This function returns a list of Volume objects.

**get\_app\_images** ()

This function returns a list of Image objects representing public DigitalOcean ‘One-Click’ application images.

**get\_certificate** (*id*)

Returns a Certificate object by its ID.

**Parameters** *id* (*str*) – Certificate ID

**get\_distro\_images** ()

This function returns a list of Image objects representing public base distribution images.

**get\_domain** (*domain\_name*)

Return a Domain by its domain\_name

**get\_droplet** (*droplet\_id*)

Return a Droplet by its ID.

**get\_floating\_ip** (*ip*)

Returns a FloatingIP object by its IP address.

**get\_global\_images** ()

This function returns a list of Image objects representing public DigitalOcean images (e.g. base distribution images and ‘One-Click’ applications).

**get\_image** (*image\_id*)

Return a Image by its ID.

**get\_images** (*private=False*, *type=None*)

This function returns a list of Image object.

**get\_load\_balancer** (*id*)

Returns a Load Balancer object by its ID.

**Parameters** *id* (*str*) – Load Balancer ID

**get\_my\_images** ()

This function returns a list of Image objects representing private DigitalOcean images (e.g. snapshots and backups).

**get\_ssh\_key** (*ssh\_key\_id*)

Return a SSHKey object by its ID.

**get\_volume** (*volume\_id*)

Returns a Volume object by its ID.

## digitalocean.Metadata module

```
class digitalocean.Metadata.Metadata (*args, **kwargs)
    Bases: digitalocean.baseapi.BaseAPI

    Metadata API: Provide useful information about the current Droplet. See: https://developers.digitalocean.com/metadata/#introduction

    droplet_id = None

    end_point = 'http://169.254.169.254/metadata/v1'

    get_data (url, headers={}, params={}, render_json=True)
        Customized version of get_data to directly get the data without using the authentication method.

    load ()
```

## digitalocean.Record module

```
class digitalocean.Record.Record (domain_name=None, *args, **kwargs)
    Bases: digitalocean.baseapi.BaseAPI

    create ()
        Create a record for a domain

    destroy ()
        Destroy the record

    classmethod get_object (api_token, domain, record_id)
        Class method that will return a Record object by ID and the domain.

    load ()

    save ()
        Save existing record
```

## digitalocean.Region module

```
class digitalocean.Region.Region (*args, **kwargs)
    Bases: digitalocean.baseapi.BaseAPI
```

## digitalocean.SSHKey module

```
class digitalocean.SSHKey.SSHKey (*args, **kwargs)
    Bases: digitalocean.baseapi.BaseAPI

    create ()
        Create the SSH Key

    destroy ()
        Destroy the SSH Key

    edit ()
        Edit the SSH Key
```



**classmethod** `get_object` (*api\_token, ssh\_key\_id*)  
 Class method that will return a SSHKey object by ID.

**load** ()  
 Load the SSHKey object from DigitalOcean.  
 Requires either self.id or self.fingerprint to be set.

**load\_by\_pub\_key** (*public\_key*)  
 This method will load a SSHKey object from DigitalOcean from a public\_key. This method will avoid problem like uploading the same public\_key twice.

## digitalocean.Size module

**class** `digitalocean.Size.Size` (*\*args, \*\*kwargs*)  
 Bases: `digitalocean.baseapi.BaseAPI`

## digitalocean.Tag module

**class** `digitalocean.Tag.Tag` (*\*args, \*\*kwargs*)  
 Bases: `digitalocean.baseapi.BaseAPI`

**add\_droplets** (*droplet*)  
 Add the Tag to a Droplet.

**Attributes accepted at creation time:** droplet: array of string or array of int, or array of Droplets.

**create** (*\*\*kwargs*)  
 Create the tag.

**delete** ()

**classmethod** `get_object` (*api\_token, tag\_name*)

**load** ()  
 Fetch data about tag

**remove\_droplets** (*droplet*)  
 Remove the Tag from the Droplet.

**Attributes accepted at creation time:** droplet: array of string or array of int, or array of Droplets.

**update\_tag** (*name*)

## digitalocean.Volume module

**class** `digitalocean.Volume.Volume` (*\*args, \*\*kwargs*)  
 Bases: `digitalocean.baseapi.BaseAPI`

**attach** (*droplet\_id, region*)  
 Attach a Volume to a Droplet.

### Parameters

- **droplet\_id** – int - droplet id
- **region** – string - slug identifier for the region

**create** (\*args, \*\*kwargs)

Creates a Block Storage volume

Note: Every argument and parameter given to this method will be assigned to the object.

**Parameters**

- **name** – string - a name for the volume
- **region** – string - slug identifier for the region
- **size\_gigabytes** – int - size of the Block Storage volume in GiB

**Optional Args:** description: string - text field to describe a volume

**destroy** ()

Destroy a volume

**detach** (droplet\_id, region)

Detach a Volume to a Droplet.

**Parameters**

- **droplet\_id** – int - droplet id
- **region** – string - slug identifier for the region

**classmethod get\_object** (api\_token, volume\_id)

Class method that will return an Volume object by ID.

**load** ()

**resize** (size\_gigabytes, region)

Detach a Volume to a Droplet.

**Parameters**

- **size\_gigabytes** – int - size of the Block Storage volume in GiB
- **region** – string - slug identifier for the region

## digitalocean.baseapi module

**class** digitalocean.baseapi.**BaseAPI** (\*args, \*\*kwargs)

Bases: object

Basic api class for

**end\_point** = 'https://api.digitalocean.com/v2/'

**get\_data** (url, type='GET', params=None)

This method is a basic implementation of `__call_api` that checks errors too. In case of success the method will return True or the content of the response to the request.

Pagination is automatically detected and handled accordingly

**get\_timeout** ()

Checks if any timeout for the requests to DigitalOcean is required. To set a timeout, use the `REQUEST_TIMEOUT_ENV_VAR` environment variable.

**token** = ''

**exception** `digitalocean.baseapi.DataReadError`

Bases: `digitalocean.baseapi.Error`

**exception** `digitalocean.baseapi.Error`

Bases: `exceptions.Exception`

Base exception class for this module

**exception** `digitalocean.baseapi.JSONReadError`

Bases: `digitalocean.baseapi.Error`

**exception** `digitalocean.baseapi.NotFoundError`

Bases: `digitalocean.baseapi.Error`

**exception** `digitalocean.baseapi.TokenError`

Bases: `digitalocean.baseapi.Error`

## Module contents

digitalocean API to manage droplets



## CHAPTER 2

---

### Indices and tables

---

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



**d**

digitalocean, 15  
digitalocean.Account, 1  
digitalocean.Action, 1  
digitalocean.baseapi, 14  
digitalocean.Domain, 2  
digitalocean.Droplet, 2  
digitalocean.FloatingIP, 6  
digitalocean.Image, 7  
digitalocean.Kernel, 7  
digitalocean.LoadBalancer, 7  
digitalocean.Manager, 10  
digitalocean.Metadata, 12  
digitalocean.Record, 12  
digitalocean.Region, 12  
digitalocean.Size, 13  
digitalocean.SSHKey, 12  
digitalocean.Tag, 13  
digitalocean.Volume, 13





**A**

Account (class in digitalocean.Account), 1  
 Action (class in digitalocean.Action), 1  
 add\_droplets() (digitalocean.LoadBalancer.LoadBalancer method), 9  
 add\_droplets() (digitalocean.Tag.Tag method), 13  
 add\_forwarding\_rules() (digitalocean.LoadBalancer.LoadBalancer method), 9  
 assign() (digitalocean.FloatingIP.FloatingIP method), 6  
 attach() (digitalocean.Volume.Volume method), 13

**B**

BadKernelObject, 2  
 BadSSHKeyFormat, 2  
 BaseAPI (class in digitalocean.baseapi), 14

**C**

change\_kernel() (digitalocean.Droplet.Droplet method), 3  
 create() (digitalocean.Domain.Domain method), 2  
 create() (digitalocean.Droplet.Droplet method), 3  
 create() (digitalocean.FloatingIP.FloatingIP method), 6  
 create() (digitalocean.LoadBalancer.LoadBalancer method), 9  
 create() (digitalocean.Record.Record method), 12  
 create() (digitalocean.SSHKey.SSHKey method), 12  
 create() (digitalocean.Tag.Tag method), 13  
 create() (digitalocean.Volume.Volume method), 13  
 create\_multiple() (digitalocean.Droplet.Droplet class method), 3  
 create\_new\_domain\_record() (digitalocean.Domain.Domain method), 2

**D**

DataReadError, 14  
 delete() (digitalocean.Tag.Tag method), 13  
 destroy() (digitalocean.Domain.Domain method), 2  
 destroy() (digitalocean.Droplet.Droplet method), 3  
 destroy() (digitalocean.FloatingIP.FloatingIP method), 6

destroy() (digitalocean.Image.Image method), 7  
 destroy() (digitalocean.LoadBalancer.LoadBalancer method), 9  
 destroy() (digitalocean.Record.Record method), 12  
 destroy() (digitalocean.SSHKey.SSHKey method), 12  
 destroy() (digitalocean.Volume.Volume method), 14  
 detach() (digitalocean.Volume.Volume method), 14  
 digitalocean (module), 15  
 digitalocean.Account (module), 1  
 digitalocean.Action (module), 1  
 digitalocean.baseapi (module), 14  
 digitalocean.Domain (module), 2  
 digitalocean.Droplet (module), 2  
 digitalocean.FloatingIP (module), 6  
 digitalocean.Image (module), 7  
 digitalocean.Kernel (module), 7  
 digitalocean.LoadBalancer (module), 7  
 digitalocean.Manager (module), 10  
 digitalocean.Metadata (module), 12  
 digitalocean.Record (module), 12  
 digitalocean.Region (module), 12  
 digitalocean.Size (module), 13  
 digitalocean.SSHKey (module), 12  
 digitalocean.Tag (module), 13  
 digitalocean.Volume (module), 13  
 disable\_backups() (digitalocean.Droplet.Droplet method), 3  
 Domain (class in digitalocean.Domain), 2  
 Droplet (class in digitalocean.Droplet), 2  
 droplet\_id (digitalocean.Metadata.Metadata attribute), 12  
 DropletError, 6

**E**

edit() (digitalocean.SSHKey.SSHKey method), 12  
 enable\_backups() (digitalocean.Droplet.Droplet method), 3  
 enable\_ipv6() (digitalocean.Droplet.Droplet method), 3  
 enable\_private\_networking() (digitalocean.Droplet.Droplet method), 4  
 end\_point (digitalocean.baseapi.BaseAPI attribute), 14

end\_point (digitalocean.Metadata.Metadata attribute), 12  
 Error, 15

## F

FloatingIP (class in digitalocean.FloatingIP), 6  
 ForwardingRule (class in digitalocean.LoadBalancer), 7

## G

get\_account() (digitalocean.Manager.Manager method), 10  
 get\_action() (digitalocean.Droplet.Droplet method), 4  
 get\_action() (digitalocean.Manager.Manager method), 10  
 get\_actions() (digitalocean.Droplet.Droplet method), 4  
 get\_all\_certificates() (digitalocean.Manager.Manager method), 10  
 get\_all\_domains() (digitalocean.Manager.Manager method), 10  
 get\_all\_droplets() (digitalocean.Manager.Manager method), 10  
 get\_all\_floating\_ips() (digitalocean.Manager.Manager method), 10  
 get\_all\_images() (digitalocean.Manager.Manager method), 10  
 get\_all\_load\_balancers() (digitalocean.Manager.Manager method), 10  
 get\_all\_regions() (digitalocean.Manager.Manager method), 10  
 get\_all\_sizes() (digitalocean.Manager.Manager method), 11  
 get\_all\_sshkeys() (digitalocean.Manager.Manager method), 11  
 get\_all\_volumes() (digitalocean.Manager.Manager method), 11  
 get\_app\_images() (digitalocean.Manager.Manager method), 11  
 get\_certificate() (digitalocean.Manager.Manager method), 11  
 get\_data() (digitalocean.baseapi.BaseAPI method), 14  
 get\_data() (digitalocean.Droplet.Droplet method), 4  
 get\_data() (digitalocean.Metadata.Metadata method), 12  
 get\_distro\_images() (digitalocean.Manager.Manager method), 11  
 get\_domain() (digitalocean.Manager.Manager method), 11  
 get\_droplet() (digitalocean.Manager.Manager method), 11  
 get\_events() (digitalocean.Droplet.Droplet method), 4  
 get\_floating\_ip() (digitalocean.Manager.Manager method), 11  
 get\_global\_images() (digitalocean.Manager.Manager method), 11  
 get\_image() (digitalocean.Manager.Manager method), 11  
 get\_images() (digitalocean.Manager.Manager method), 11

get\_kernel\_available() (digitalocean.Droplet.Droplet method), 4  
 get\_load\_balancer() (digitalocean.Manager.Manager method), 11  
 get\_my\_images() (digitalocean.Manager.Manager method), 11  
 get\_object() (digitalocean.Account.Account class method), 1  
 get\_object() (digitalocean.Action.Action class method), 1  
 get\_object() (digitalocean.Domain.Domain class method), 2  
 get\_object() (digitalocean.Droplet.Droplet class method), 4  
 get\_object() (digitalocean.FloatingIP.FloatingIP class method), 6  
 get\_object() (digitalocean.Image.Image class method), 7  
 get\_object() (digitalocean.LoadBalancer.LoadBalancer class method), 9  
 get\_object() (digitalocean.Record.Record class method), 12  
 get\_object() (digitalocean.SSHKey.SSHKey class method), 12  
 get\_object() (digitalocean.Tag.Tag class method), 13  
 get\_object() (digitalocean.Volume.Volume class method), 14  
 get\_records() (digitalocean.Domain.Domain method), 2  
 get\_snapshots() (digitalocean.Droplet.Droplet method), 4  
 get\_ssh\_key() (digitalocean.Manager.Manager method), 11  
 get\_timeout() (digitalocean.baseapi.BaseAPI method), 14  
 get\_volume() (digitalocean.Manager.Manager method), 11

## H

HealthCheck (class in digitalocean.LoadBalancer), 8

## I

Image (class in digitalocean.Image), 7

## J

JSONReadError, 15

## K

Kernel (class in digitalocean.Kernel), 7

## L

load() (digitalocean.Account.Account method), 1  
 load() (digitalocean.Action.Action method), 1  
 load() (digitalocean.Domain.Domain method), 2  
 load() (digitalocean.Droplet.Droplet method), 4  
 load() (digitalocean.FloatingIP.FloatingIP method), 7  
 load() (digitalocean.Image.Image method), 7  
 load() (digitalocean.LoadBalancer.LoadBalancer method), 10

load() (digitalocean.Metadata.Metadata method), 12  
 load() (digitalocean.Record.Record method), 12  
 load() (digitalocean.SSHKey.SSHKey method), 13  
 load() (digitalocean.Tag.Tag method), 13  
 load() (digitalocean.Volume.Volume method), 14  
 load\_by\_pub\_key() (digitalocean.SSHKey.SSHKey method), 13  
 load\_directly() (digitalocean.Action.Action method), 1  
 LoadBalancer (class in digitalocean.LoadBalancer), 8

## M

Manager (class in digitalocean.Manager), 10  
 Metadata (class in digitalocean.Metadata), 12

## N

NotFoundError, 15

## P

power\_cycle() (digitalocean.Droplet.Droplet method), 4  
 power\_off() (digitalocean.Droplet.Droplet method), 4  
 power\_on() (digitalocean.Droplet.Droplet method), 4

## R

reboot() (digitalocean.Droplet.Droplet method), 5  
 rebuild() (digitalocean.Droplet.Droplet method), 5  
 Record (class in digitalocean.Record), 12  
 Region (class in digitalocean.Region), 12  
 remove\_droplets() (digitalocean.LoadBalancer.LoadBalancer method), 10  
 remove\_droplets() (digitalocean.Tag.Tag method), 13  
 remove\_forwarding\_rules() (digitalocean.LoadBalancer.LoadBalancer method), 10  
 rename() (digitalocean.Droplet.Droplet method), 5  
 rename() (digitalocean.Image.Image method), 7  
 reserve() (digitalocean.FloatingIP.FloatingIP method), 7  
 reset\_root\_password() (digitalocean.Droplet.Droplet method), 5  
 resize() (digitalocean.Droplet.Droplet method), 5  
 resize() (digitalocean.Volume.Volume method), 14  
 restore() (digitalocean.Droplet.Droplet method), 5

## S

save() (digitalocean.Record.Record method), 12  
 shutdown() (digitalocean.Droplet.Droplet method), 6  
 Size (class in digitalocean.Size), 13  
 SSHKey (class in digitalocean.SSHKey), 12  
 StickySessions (class in digitalocean.LoadBalancer), 10

## T

Tag (class in digitalocean.Tag), 13  
 take\_snapshot() (digitalocean.Droplet.Droplet method), 6

token (digitalocean.baseapi.BaseAPI attribute), 14  
 TokenError, 15  
 transfer() (digitalocean.Image.Image method), 7

## U

unassign() (digitalocean.FloatingIP.FloatingIP method), 7  
 update\_tag() (digitalocean.Tag.Tag method), 13

## V

Volume (class in digitalocean.Volume), 13

## W

wait() (digitalocean.Action.Action method), 1