



Red Hat Ceph Storage 3

Ceph Management API

Reference and Integration Guide

Red Hat Ceph Storage 3 Ceph Management API

Reference and Integration Guide

Legal Notice

Copyright © 2021 Red Hat, Inc.

The text of and illustrations in this document are licensed by Red Hat under a Creative Commons Attribution–Share Alike 3.0 Unported license ("CC-BY-SA"). An explanation of CC-BY-SA is available at

<http://creativecommons.org/licenses/by-sa/3.0/>

. In accordance with CC-BY-SA, if you distribute this document or an adaptation of it, you must provide the URL for the original version.

Red Hat, as the licensor of this document, waives the right to enforce, and agrees not to assert, Section 4d of CC-BY-SA to the fullest extent permitted by applicable law.

Red Hat, Red Hat Enterprise Linux, the Shadowman logo, the Red Hat logo, JBoss, OpenShift, Fedora, the Infinity logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

Linux[®] is the registered trademark of Linus Torvalds in the United States and other countries.

Java[®] is a registered trademark of Oracle and/or its affiliates.

XFS[®] is a trademark of Silicon Graphics International Corp. or its subsidiaries in the United States and/or other countries.

MySQL[®] is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js[®] is an official trademark of Joyent. Red Hat is not formally related to or endorsed by the official Joyent Node.js open source or commercial project.

The OpenStack[®] Word Mark and OpenStack logo are either registered trademarks/service marks or trademarks/service marks of the OpenStack Foundation, in the United States and other countries and are used with the OpenStack Foundation's permission. We are not affiliated with, endorsed or sponsored by the OpenStack Foundation, or the OpenStack community.

All other trademarks are the property of their respective owners.

Abstract

This guide describes how to use the RESTful plug-in to view information about a cluster, change its configuration, and administer it.

Table of Contents

CHAPTER 1. RESTFUL PLUG-IN	4
1.1. WHAT IS THE RESTFUL PLUG-IN?	4
1.2. ENABLING AND SECURING THE RESTFUL PLUG-IN	4
CHAPTER 2. QUESTIONS AND ANSWERS	6
2.1. GETTING INFORMATION	6
2.1.1. How Can I View All Cluster Configuration Options?	6
The curl Command	6
Python	7
Web Browser	7
Additional Resources	7
2.1.2. How Can I View a Particular Cluster Configuration Option?	7
The curl Command	7
Python	8
Web Browser	8
Additional Resources	9
2.1.3. How Can I View All Configuration Options for OSDs?	9
The curl Command	9
Python	9
Web Browser	9
Additional Resources	10
2.1.4. How Can I View CRUSH Rules?	10
The curl Command	10
Python	10
Web Browser	11
Additional Resources	11
2.1.5. How Can I View Information about Monitors?	11
The curl Command	11
Python	11
Web Browser	12
2.1.6. How Can I View Information About a Particular Monitor?	12
The curl Command	12
Python	13
Web Browser	13
2.1.7. How Can I View Information about OSDs?	13
The curl Command	14
Python	14
Web Browser	14
2.1.8. How Can I View Information about a Particular OSD?	15
The curl Command	15
Python	15
Web Browser	16
2.1.9. How Can I Determine What Processes Can Be Scheduled on an OSD?	16
The curl Command	16
Python	16
Web Browser	17
2.1.10. How Can I View Information About Pools?	17
The curl Command	17
Python	18
Web Browser	18
2.1.11. How Can I View Information About a Particular Pool?	18

The curl Command	18
Python	19
Web Browser	19
2.1.12. How Can I View Information About Requests?	19
The curl Command	20
Python	20
Web Browser	20
2.1.13. How Can I View Information About a Particular Request?	21
The curl Command	21
Python	21
Web Browser	21
2.1.14. How Can I View Information About Hosts?	22
The curl Command	22
Python	22
Web Browser	23
2.1.15. How Can I View Information About a Particular Host?	23
The curl Command	23
Python	23
Web Browser	24
2.2. CHANGING CONFIGURATION	24
2.2.1. How Can I Change OSD Configuration Options?	24
The curl Command	24
Python	25
2.2.2. How Can I Change the OSD State?	25
The curl Command	26
Python	26
2.2.3. How Can I Reweight an OSD?	27
The curl Command	27
Python	27
2.2.4. How Can I Change Information for a Pool?	28
The curl Command	28
Python	28
2.3. ADMINISTERING THE CLUSTER	29
2.3.1. How Can I Run a Scheduled Process on an OSD?	29
The curl Command	29
Python	30
2.3.2. How Can I Create a New Pool?	30
The curl Command	30
Python	31
2.3.3. How Can I Remove Pools?	31
The curl Command	31
Python	32
2.3.4. How Can I Remove All Finished Requests?	32
The curl Command	32
Python	33
2.3.5. How Can I Remove a Particular Request?	33
The curl Command	33
Python	34
2.3.6. How Can I Execute Administrative Commands?	34
Python	34
2.4. ADDITIONAL RESOURCES	35

CHAPTER 1. RESTFUL PLUG-IN

1.1. WHAT IS THE RESTFUL PLUG-IN?

The RESTful plug-in for the Ceph Manager (**ceph-mgr**) provides an API for interacting with a Red Hat Ceph Storage cluster.

You can use the API to:

- Show the information about Monitors and OSDs
- Create or edit pools
- View and start scheduled processes on OSDs
- Show configuration options for the cluster, Monitor, and OSDs

1.2. ENABLING AND SECURING THE RESTFUL PLUG-IN

The RESTful plug-in offers the REST API access to the status of the cluster over an SSL-secured connection. This section describes how to enable the plug-in and secure it.

Prerequisites

- Ensure that you have at least one **ceph-mgr** daemon active. See the *Installing a Red Hat Ceph Storage* section in the [Installation Guide for Red Hat Enterprise Linux](#) or [Ubuntu](#).
- If you use a firewall, ensure that the **8003** port is enabled on the node with the active **ceph-mgr** daemon.

Procedure

Use the following commands on a node with the administration keyring.

1. Enable the RESTful plug-in.

```
[root@admin ~]# ceph mgr module enable restful
```

2. Configure an SSL certificate.

- a. If your organization's certificate authority provides a certificate, set the certificate:

```
ceph config-key set mgr/restful/hostname/cert -i certificate
ceph config-key set mgr/restful/hostname/key -i key
```

Replace *hostname* with the host name of the host where the active **ceph-mgr** instance is running, *certificate* with the path to the certificate file, and *key* with the path to the key file, for example:

```
[root@admin ~]# ceph config-key set mgr/restful/node1/cert -i restful.crt
[root@admin ~]# ceph config-key set mgr/restful/node1/key -i restful.key
```

If you want to use the certificate on all **ceph-mgr** instances, omit the *hostname* part, for example:


```
[root@admin ~]# ceph config-key set mgr/restful/crt -i restful.crt
[root@admin ~]# ceph config-key set mgr/restful/key -i restful.key
```

- b. Alternatively, generate a self-signed certificate. However, using a self-signed certificate does not provide full security benefits of the HTTPS protocol.

```
[root@admin ~]# ceph restful create-self-signed-cert
```

3. Create an HTTP user and generate a password for HTTP basic authentication.

```
ceph restful create-key username
```

Replace *username* with name of the user. For example, to create a user named **admin**:

```
[root@admin ~]# ceph restful create-key admin
3ce361b7-97fb-4820-8edc-1090841f078e
```

4. Connect to the RESTful plug-in web page. Open a web browser and enter the following URL:

```
https://_ceph-mgr_:8003
```

Replace *ceph-mgr* with the IP address or host name of the node with the active **ceph-mgr** daemon:

```
https://node1:8003
```

If you used a self-signed certificate, confirm a security exception.

5. Optional. If you want to use a static IP address for the RESTful plug-in, configure a load balancer.

Additional Resources

- The **ceph restful --help** command
- The **https://ceph-mgr:8003/doc** page, where *ceph-mgr* is the IP address or host name of the node with the running **ceph-mgr** instance
- The [Using OpenSSL](#) chapter in the *Security Guide* for Red Hat Enterprise Linux 7

CHAPTER 2. QUESTIONS AND ANSWERS

2.1. GETTING INFORMATION

This section describes how to use the RESTful plug-in to view information about the cluster, Monitors, OSDs, pools, hosts, and requests:

- [Section 2.1.1, “How Can I View All Cluster Configuration Options?”](#)
- [Section 2.1.2, “How Can I View a Particular Cluster Configuration Option?”](#)
- [Section 2.1.3, “How Can I View All Configuration Options for OSDs?”](#)
- [Section 2.1.4, “How Can I View CRUSH Rules?”](#)
- [Section 2.1.5, “How Can I View Information about Monitors?”](#)
- [Section 2.1.6, “How Can I View Information About a Particular Monitor?”](#)
- [Section 2.1.7, “How Can I View Information about OSDs?”](#)
- [Section 2.1.8, “How Can I View Information about a Particular OSD?”](#)
- [Section 2.1.9, “How Can I Determine What Processes Can Be Scheduled on an OSD?”](#)
- [Section 2.1.10, “How Can I View Information About Pools?”](#)
- [Section 2.1.11, “How Can I View Information About a Particular Pool?”](#)
- [Section 2.1.12, “How Can I View Information About Requests?”](#)
- [Section 2.1.13, “How Can I View Information About a Particular Request?”](#)
- [Section 2.1.14, “How Can I View Information About Hosts?”](#)
- [Section 2.1.15, “How Can I View Information About a Particular Host?”](#)

2.1.1. How Can I View All Cluster Configuration Options?

This section describes how to use the RESTful plug-in to view cluster configuration options and their values.

The **curl** Command

On the command line, use:

```
curl --silent --user <user> 'https://<ceph-mgr>:8003/config/cluster'
```

Replace:

- **<user>** with the user name
- **<ceph-mgr>** with the IP address or short host name of the node with the active **ceph-mgr** instance

Enter the user’s password when prompted.

If you used a self-signed certificate, use the **--insecure** option:

```
curl --silent --insecure --user <user> 'https://<ceph-mgr>:8003/config/cluster'
```

Python

In the Python interpreter, enter:

```
$ python
>> import requests
>> result = requests.get('https://<ceph-mgr>:8003/config/cluster', auth=("<user>", "<password>"))
>> print result.json()
```

Replace:

- **<ceph-mgr>** with the IP address or short host name of the node with the active **ceph-mgr** instance
- **<user>** with the user name
- **<password>** with the user's password

If you used a self-signed certificate, use the **verify=False** option:

```
$ python
>> import requests
>> result = requests.get('https://<ceph-mgr>:8003/config/cluster', auth=("<user>", "<password>"),
verify=False)
>> print result.json()
```

Web Browser

In the web browser, enter:

```
https://<ceph-mgr>:8003/config/cluster
```

Replace:

- **<ceph-mgr>** with the IP address or short host name of the node with the active **ceph-mgr** instance

Enter the user name and password when prompted.

Additional Resources

- The [Configuration Guide](#) for Red Hat Ceph Storage 3

2.1.2. How Can I View a Particular Cluster Configuration Option?

This section describes how to view a particular cluster option and its value.

The **curl** Command

On the command line, use:

```
curl --silent --user <user> 'https://<ceph-mgr>:8003/config/cluster/<argument>'
```

Replace:

- **<user>** with the user name
- **<ceph-mgr>** with the IP address or short host name of the node with the active **ceph-mgr** instance
- **<argument>** with the configuration option you want to view

Enter the user's password when prompted.

If you used a self-signed certificate, use the **--insecure** option:

```
curl --silent --insecure --user <user> 'https://<ceph-mgr>:8003/config/cluster/<argument>'
```

Python

In the Python interpreter, enter:

```
$ python
>> import requests
>> result = requests.get('https://<ceph-mgr>:8003/config/cluster/<argument>', auth=("<user>", "<password>"))
>> print result.json()
```

Replace:

- **<ceph-mgr>** with the IP address or short host name of the node with the active **ceph-mgr** instance
- **<argument>** with the configuration option you want to view
- **<user>** with the user name
- **<password>** with the user's password

If you used a self-signed certificate, use the **verify=False** option:

```
$ python
>> import requests
>> result = requests.get('https://<ceph-mgr>:8003/config/cluster/<argument>', auth=("<user>", "<password>"), verify=False)
>> print result.json()
```

Web Browser

In the web browser, enter:

```
https://<ceph-mgr>:8003/config/cluster/<argument>
```

Replace:

- **<ceph-mgr>** with the IP address or short host name of the node with the active **ceph-mgr** instance
- **<argument>** with the configuration option you want to view

Enter the user name and password when prompted.

Additional Resources

- The [Configuration Guide](#) for Red Hat Ceph Storage 3

2.1.3. How Can I View All Configuration Options for OSDs?

This section describes how to view all configuration options and their values for OSDs.

The `curl` Command

On the command line, use:

```
curl --silent --user <user> 'https://<ceph-mgr>:8003/config/osd'
```

Replace:

- **<user>** with the user name
- **<ceph-mgr>** with the IP address or short host name of the node with the active **ceph-mgr** instance

Enter the user's password when prompted.

If you used a self-signed certificate, use the **--insecure** option:

```
curl --silent --insecure --user <user> 'https://<ceph-mgr>:8003/config/osd'
```

Python

In the Python interpreter, enter:

```
$ python
>> import requests
>> result = requests.get('https://<ceph-mgr>:8003/config/osd', auth=("<user>", "<password>"))
>> print result.json()
```

Replace:

- **<ceph-mgr>** with the IP address or short host name of the node with the active **ceph-mgr** instance
- **<user>** with the user name
- **<password>** with the user's password

If you used a self-signed certificate, use the **verify=False** option:

```
$ python
>> import requests
>> result = requests.get('https://<ceph-mgr>:8003/config/osd', auth=("<user>", "<password>"),
verify=False)
>> print result.json()
```

Web Browser

In the web browser, enter:

```
https://<ceph-mgr>:8003/config/osd
```

Replace:

- **<ceph-mgr>** with the IP address or short host name of the node with the active **ceph-mgr** instance

Enter the user name and password when prompted.

Additional Resources

- The [Configuration Guide](#) for Red Hat Ceph Storage 3

2.1.4. How Can I View CRUSH Rules?

This section describes how to view CRUSH rules.

The curl Command

On the command line, use:

```
curl --silent --user <user> 'https://<ceph-mgr>:8003/crush/rule'
```

Replace:

- **<user>** with the user name
- **<ceph-mgr>** with the IP address or short host name of the node with the active **ceph-mgr** instance

Enter the user's password when prompted.

If you used a self-signed certificate, use the **--insecure** option:

```
curl --silent --insecure --user <user> 'https://<ceph-mgr>:8003/crush/rule'
```

Python

In the Python interpreter, enter:

```
$ python
>> import requests
>> result = requests.get('https://<ceph-mgr>:8003/crush/rule', auth=("<user>", "<password>"))
>> print result.json()
```

Replace:

- **<ceph-mgr>** with the IP address or short host name of the node with the active **ceph-mgr** instance
- **<user>** with the user name
- **<password>** with the user's password

If you used a self-signed certificate, use the **verify=False** option:

```
$ python
```

```
>> import requests
>> result = requests.get('https://<ceph-mgr>:8003/crush/rule', auth=("<user>", "<password>"),
verify=False)
>> print result.json()
```

Web Browser

In the web browser, enter:

```
https://<ceph-mgr>:8003/crush/rule
```

Replace:

- **<ceph-mgr>** with the IP address or short host name of the node with the active **ceph-mgr** instance

Enter the user name and password when prompted.

Additional Resources

- The [CRUSH Rules](#) section in the *Administration Guide* for Red Hat Ceph Storage 3

2.1.5. How Can I View Information about Monitors?

This section describes how to view information about a particular Monitor, such as:

- IP address
- Name
- Quorum status

The curl Command

On the command line, use:

```
curl --silent --user <user> 'https://<ceph-mgr>:8003/mon'
```

Replace:

- **<user>** with the user name
- **<ceph-mgr>** with the IP address or short host name of the node with the active **ceph-mgr** instance

Enter the user's password when prompted.

If you used a self-signed certificate, use the **--insecure** option:

```
curl --silent --insecure --user <user> 'https://<ceph-mgr>:8003/mon'
```

Python

In the Python interpreter, enter:

```
$ python
>> import requests
>> result = requests.get('https://<ceph-mgr>:8003/mon', auth=("<user>", "<password>"))
```

```
>> print result.json()
```

Replace:

- **<ceph-mgr>** with the IP address or short host name of the node with the active **ceph-mgr** instance
- **<user>** with the user name
- **<password>** with the user's password

If you used a self-signed certificate, use the **verify=False** option:

```
$ python
>> import requests
>> result = requests.get('https://<ceph-mgr>:8003/mon', auth=("<user>", "<password>"), verify=False)
>> print result.json()
```

Web Browser

In the web browser, enter:

```
https://<ceph-mgr>:8003/mon
```

Replace:

- **<ceph-mgr>** with the IP address or short host name of the node with the active **ceph-mgr** instance

Enter the user name and password when prompted.

2.1.6. How Can I View Information About a Particular Monitor?

This section describes how to view information about a particular Monitor, such as:

- IP address
- Name
- Quorum status

The curl Command

On the command line, use:

```
curl --silent --user <user> 'https://<ceph-mgr>:8003/mon/<name>'
```

Replace:

- **<user>** with the user name
- **<ceph-mgr>** with the IP address or short host name of the node with the active **ceph-mgr** instance
- **<name>** with the short host name of the Monitor

Enter the user's password when prompted.

If you used a self-signed certificate, use the **--insecure** option:

```
curl --silent --insecure --user <user> 'https://<ceph-mgr>:8003/mon/<name>'
```

Python

In the Python interpreter, enter:

```
$ python
>> import requests
>> result = requests.get('https://<ceph-mgr>:8003/mon/<name>', auth=("<user>", "<password>"))
>> print result.json()
```

Replace:

- **<ceph-mgr>** with the IP address or short host name of the node with the active **ceph-mgr** instance
- **<name>** with the short host name of the Monitor
- **<user>** with the user name
- **<password>** with the user's password

If you used a self-signed certificate, use the **verify=False** option:

```
$ python
>> import requests
>> result = requests.get('https://<ceph-mgr>:8003/mon/<name>', auth=("<user>", "<password>"),
verify=False)
>> print result.json()
```

Web Browser

In the web browser, enter:

```
https://<ceph-mgr>:8003/mon/<name>
```

Replace:

- **<ceph-mgr>** with the IP address or short host name of the node with the active **ceph-mgr** instance
- **<name>** with the short host name of the Monitor

Enter the user name and password when prompted.

2.1.7. How Can I View Information about OSDs?

This section describes how to view information about OSDs, such as:

- IP address
- Its pools
- Affinity

- Weight

The `curl` Command

On the command line, use:

```
curl --silent --user <user> 'https://<ceph-mgr>:8003/osd'
```

Replace:

- **<user>** with the user name
- **<ceph-mgr>** with the IP address or short host name of the node with the active **ceph-mgr** instance

Enter the user's password when prompted.

If you used a self-signed certificate, use the **--insecure** option:

```
curl --silent --insecure --user <user> 'https://<ceph-mgr>:8003/osd'
```

Python

In the Python interpreter, enter:

```
$ python
>> import requests
>> result = requests.get("https://<ceph-mgr>:8003/osd/", auth=("<user>", "<password>"))
>> print result.json()
```

Replace:

- **<ceph-mgr>** with the IP address or short host name of the node with the active **ceph-mgr** instance
- **<user>** with the user name
- **<password>** with the user's password

If you used a self-signed certificate, use the **verify=False** option:

```
$ python
>> import requests
>> result = requests.get("https://<ceph-mgr>:8003/osd/", auth=("<user>", "<password>"), verify=False)
>> print result.json()
```

Web Browser

In the web browser, enter:

```
https://<ceph-mgr>:8003/osd
```

Replace:

- **<ceph-mgr>** with the IP address or short host name of the node with the active **ceph-mgr** instance

Enter the user name and password when prompted.

2.1.8. How Can I View Information about a Particular OSD?

This section describes how to view information about a particular OSD, such as:

- IP address
- Its pools
- Affinity
- Weight

The `curl` Command

On the command line, use:

```
curl --silent --user <user> 'https://<ceph-mgr>:8003/osd/<id>'
```

Replace:

- **<user>** with the user name
- **<ceph-mgr>** with the IP address or short host name of the node with the active **ceph-mgr** instance
- **<id>** with the ID of the OSD listed in the **osd** field

Enter the user's password when prompted.

If you used a self-signed certificate, use the **--insecure** option:

```
curl --silent --insecure --user <user> 'https://<ceph-mgr>:8003/osd/<id>'
```

Python

In the Python interpreter, enter:

```
$ python
>> import requests
>> result = requests.get('https://<ceph-mgr>:8003/osd/<id>', auth=("<user>", "<password>"))
>> print result.json()
```

Replace:

- **<ceph-mgr>** with the IP address or short host name of the node with the active **ceph-mgr** instance
- **<id>** with the ID of the OSD listed in the **osd** field
- **<user>** with the user name
- **<password>** with the user's password

If you used a self-signed certificate, use the **verify=False** option:

```
$ python
>> import requests
>> result = requests.get('https://<ceph-mgr>:8003/osd/<id>', auth=("<user>", "<password>"),
```

```
verify=False)
>> print result.json()
```

Web Browser

In the web browser, enter:

```
https://<ceph-mgr>:8003/osd/<id>
```

Replace:

- **<ceph-mgr>** with the IP address or short host name of the node with the active **ceph-mgr** instance
- **<id>** with the ID of the OSD listed in the **osd** field

Enter the user name and password when prompted.

2.1.9. How Can I Determine What Processes Can Be Scheduled on an OSD?

This section describes how to use the RESTful plug-in to view what processes, such as scrubbing or deep scrubbing, can be scheduled on an OSD.

The curl Command

On the command line, use:

```
curl --silent --user <user> 'https://<ceph-mgr>:8003/osd/<id>/command'
```

Replace:

- **<user>** with the user name
- **<ceph-mgr>** with the IP address or short host name of the node with the active **ceph-mgr** instance
- **<id>** with the ID of the OSD listed in the **osd** field

Enter the user's password when prompted.

If you used a self-signed certificate, use the **--insecure** option:

```
curl --silent --insecure --user <user> 'https://<ceph-mgr>:8003/osd/<id>/command'
```

Python

In the Python interpreter, enter:

```
$ python
>> import requests
>> result = requests.get('https://<ceph-mgr>:8003/osd/<id>/command', auth=("<user>", "<password>"))
>> print result.json()
```

Replace:

- **<ceph-mgr>** with the IP address or short host name of the node with the active **ceph-mgr** instance

- **<id>** with the ID of the OSD listed in the **osd** field
- **<user>** with the user name
- **<password>** with the user's password

If you used a self-signed certificate, use the **verify=False** option:

```
$ python
>> import requests
>> result = requests.get('https://<ceph-mgr>:8003/osd/<id>/command', auth=("<user>", "
<password>"), verify=False)
>> print result.json()
```

Web Browser

In the web browser, enter:

```
https://<ceph-mgr>:8003/osd/<id>/command
```

Replace:

- **<ceph-mgr>** with the IP address or short host name of the node with the active **ceph-mgr** instance
- **<id>** with the ID of the OSD listed in the **osd** field

Enter the user name and password when prompted.

2.1.10. How Can I View Information About Pools?

This section describes how to view information about pools, such as:

- Flags
- Size
- Number of placement groups

The curl Command

On the command line, use:

```
curl --silent --user <user> 'https://<ceph-mgr>:8003/pool'
```

Replace:

- **<user>** with the user name
- **<ceph-mgr>** with the IP address or short host name of the node with the active **ceph-mgr** instance

Enter the user's password when prompted.

If you used a self-signed certificate, use the **--insecure** option:

```
curl --silent --insecure --user <user> 'https://<ceph-mgr>:8003/pool'
```

Python

In the Python interpreter, enter:

```
$ python
>> import requests
>> result = requests.get("https://<ceph-mgr>:8003/pool", auth=("<user>", "<password>"))
>> print result.json()
```

Replace:

- **<ceph-mgr>** with the IP address or short host name of the node with the active **ceph-mgr** instance
- **<user>** with the user name
- **<password>** with the user's password

If you used a self-signed certificate, use the **verify=False** option:

```
$ python
>> import requests
>> result = requests.get("https://<ceph-mgr>:8003/pool", auth=("<user>", "<password>"), verify=False)
>> print result.json()
```

Web Browser

In the web browser, enter:

```
https://<ceph-mgr>:8003/pool
```

Replace:

- **<ceph-mgr>** with the IP address or short host name of the node with the active **ceph-mgr** instance

Enter the user name and password when prompted.

2.1.11. How Can I View Information About a Particular Pool?

This section describes how to view information about a particular pool, such as:

- Flags
- Size
- Number of placement groups

The curl Command

On the command line, use:

```
curl --silent --user <user> 'https://<ceph-mgr>:8003/pool/<id>'
```

Replace:

- **<user>** with the user name

- **<ceph-mgr>** with the IP address or short host name of the node with the active **ceph-mgr** instance
- **<id>** with the ID the pool listed in the **pool** field

Enter the user's password when prompted.

If you used a self-signed certificate, use the **--insecure** option:

```
curl --silent --insecure --user <user> 'https://<ceph-mgr>:8003/config/cluster'
```

Python

In the Python interpreter, enter:

```
$ python
>> import requests
>> result = requests.get('https://<ceph-mgr>:8003/pool/<id>', auth=("<user>", "<password>"))
>> print result.json()
```

Replace:

- **<ceph-mgr>** with the IP address or short host name of the node with the active **ceph-mgr** instance
- **<id>** with the ID the pool listed in the **pool** field
- **<user>** with the user name
- **<password>** with the user's password

If you used a self-signed certificate, use the **verify=False** option:

```
$ python
>> import requests
>> result = requests.get('https://<ceph-mgr>:8003/pool/<id>', auth=("<user>", "<password>"),
verify=False)
>> print result.json()
```

Web Browser

In the web browser, enter:

```
https://<ceph-mgr>:8003/pool/<id>
```

Replace:

- **<ceph-mgr>** with the IP address or short host name of the node with the active **ceph-mgr** instance
- **<id>** with the ID the pool listed in the **pool** field

Enter the user name and password when prompted.

2.1.12. How Can I View Information About Requests?

This section describes how view information about requests.

The curl Command

On the command line, use:

```
curl --silent --user <user> 'https://<ceph-mgr>:8003/request'
```

Replace:

- **<user>** with the user name
- **<ceph-mgr>** with the IP address or short host name of the node with the active **ceph-mgr** instance

Enter the user's password when prompted.

If you used a self-signed certificate, use the **--insecure** option:

```
curl --silent --insecure --user <user> 'https://<ceph-mgr>:8003/request'
```

Python

In the Python interpreter, enter:

```
$ python
>> import requests
>> result = requests.get('https://<ceph-mgr>:8003/request', auth=("<user>", "<password>"))
>> print result.json()
```

Replace:

- **<ceph-mgr>** with the IP address or short host name of the node with the active **ceph-mgr** instance
- **<user>** with the user name
- **<password>** with the user's password

If you used a self-signed certificate, use the **verify=False** option:

```
$ python
>> import requests
>> result = requests.get('https://<ceph-mgr>:8003/request', auth=("<user>", "<password>"),
verify=False)
>> print result.json()
```

Web Browser

In the web browser, enter:

```
https://<ceph-mgr>:8003/request
```

Replace:

- **<ceph-mgr>** with the IP address or short host name of the node with the active **ceph-mgr** instance

Enter the user name and password when prompted.

2.1.13. How Can I View Information About a Particular Request?

This section describes how to view information about a particular request.

The `curl` Command

On the command line, use:

```
curl --silent --user <user> 'https://<ceph-mgr>:8003/request/<id>'
```

Replace:

- **<user>** with the user name
- **<ceph-mgr>** with the IP address or short host name of the node with the active **ceph-mgr** instance
- **<id>** with the ID of the request listed in the **id** field

Enter the user's password when prompted.

If you used a self-signed certificate, use the **--insecure** option:

```
curl --silent --insecure --user <user> 'https://<ceph-mgr>:8003/request/<id>'
```

Python

In the Python interpreter, enter:

```
$ python
>> import requests
>> result = requests.get('https://<ceph-mgr>:8003/request/<id>', auth=("<user>", "<password>"))
>> print result.json()
```

Replace:

- **<ceph-mgr>** with the IP address or short host name of the node with the active **ceph-mgr** instance
- **<id>** with the ID of the request listed in the **id** field
- **<user>** with the user name
- **<password>** with the user's password

If you used a self-signed certificate, use the **verify=False** option:

```
$ python
>> import requests
>> result = requests.get('https://<ceph-mgr>:8003/request/<id>', auth=("<user>", "<password>"),
verify=False)
>> print result.json()
```

Web Browser

In the web browser, enter:

```
https://<ceph-mgr>:8003/request/<id>
```

Replace:

- **<ceph-mgr>** with the IP address or short host name of the node with the active **ceph-mgr** instance
- **<id>** with the ID of the request listed in the **id** field

Enter the user name and password when prompted.

2.1.14. How Can I View Information About Hosts?

This section describes how to view information about hosts, such as:

- Host names
- Ceph daemons and their IDs
- Ceph version

The **curl** Command

On the command line, use:

```
curl --silent --user <user> 'https://<ceph-mgr>:8003/server'
```

Replace:

- **<user>** with the user name
- **<ceph-mgr>** with the IP address or short host name of the node with the active **ceph-mgr** instance

Enter the user's password when prompted.

If you used a self-signed certificate, use the **--insecure** option:

```
curl --silent --insecure --user <user> 'https://<ceph-mgr>:8003/server'
```

Python

In the Python interpreter, enter:

```
$ python
>> import requests
>> result = requests.get('https://<ceph-mgr>:8003/server', auth=("<user>", "<password>"))
>> print result.json()
```

Replace:

- **<ceph-mgr>** with the IP address or short host name of the node with the active **ceph-mgr** instance
- **<user>** with the user name
- **<password>** with the user's password

If you used a self-signed certificate, use the **verify=False** option:

-

```
$ python
>> import requests
>> result = requests.get('https://<ceph-mgr>:8003/server', auth=("<user>", "<password>"),
verify=False)
>> print result.json()
```

Web Browser

In the web browser, enter:

```
https://<ceph-mgr>:8003/server
```

Replace:

- **<ceph-mgr>** with the IP address or short host name of the node with the active **ceph-mgr** instance

Enter the user name and password when prompted.

2.1.15. How Can I View Information About a Particular Host?

This section describes how to view information about a particular host, such as:

- Host names
- Ceph daemons and their IDs
- Ceph version

The curl Command

On the command line, use:

```
curl --silent --user <user> 'https://<ceph-mgr>:8003/server/<hostname>'
```

Replace:

- **<user>** with the user name
- **<ceph-mgr>** with the IP address or short host name of the node with the active **ceph-mgr** instance
- **<hostname>** with the host name of the host listed in the **hostname** field

Enter the user's password when prompted.

If you used a self-signed certificate, use the **--insecure** option:

```
curl --silent --insecure --user <user> 'https://<ceph-mgr>:8003/server/<hostname>'
```

Python

In the Python interpreter, enter:

```
$ python
>> import requests
>> result = requests.get('https://<ceph-mgr>:8003/server/<hostname>', auth=("<user>", "
```

```
<password>"))  
>> print result.json()
```

Replace:

- **<ceph-mgr>** with the IP address or short host name of the node with the active **ceph-mgr** instance
- **<hostname>** with the host name of the host listed in the **hostname** field
- **<user>** with the user name
- **<password>** with the user's password

If you used a self-signed certificate, use the **verify=False** option:

```
$ python  
>> import requests  
>> result = requests.get('https://<ceph-mgr>:8003/server/<hostname>', auth=("<user>", "  
<password>"), verify=False)  
>> print result.json()
```

Web Browser

In the web browser, enter:

```
https://<ceph-mgr>:8003/server/<hostname>
```

Replace:

- **<ceph-mgr>** with the IP address or short host name of the node with the active **ceph-mgr** instance
- **<hostname>** with the host name of the host listed in the **hostname** field

Enter the user name and password when prompted.

2.2. CHANGING CONFIGURATION

This section describes how to use the RESTful plug-in to change OSD configuration options, state of an OSD, and information about pools:

- [Section 2.2.1, "How Can I Change OSD Configuration Options?"](#)
- [Section 2.2.2, "How Can I Change the OSD State?"](#)
- [Section 2.2.3, "How Can I Reweight an OSD?"](#)
- [Section 2.2.4, "How Can I Change Information for a Pool?"](#)

2.2.1. How Can I Change OSD Configuration Options?

This section describes how to use the RESTful plug-in to change OSD configuration options.

The curl Command

On the command line, use:

```
echo -En '{"<option>": <value>}' | curl --request PATCH --data @- --silent --user <user>
'https://<ceph-mgr>:8003/config/osd'
```

Replace:

- **<option>** with the option to modify; **pause, noup, nodown, noout, noin, nobackfill, norecover, noscrub, nodeep-scrub**
- **<value>** with **true** or **false**
- **<user>** with the user name
- **<ceph-mgr>** with the IP address or short host name of the node with the active **ceph-mgr** instance

Enter the user's password when prompted.

If you used a self-signed certificate, use the **--insecure** option:

```
echo -En '{"<option>": <value>}' | curl --request PATCH --data @- --silent --insecure --user <user>
'https://<ceph-mgr>:8003/config/osd'
```

Python

In the Python interpreter, enter:

```
$ python
>> import requests
>> result = requests.patch('https://<ceph-mgr>:8003/config/osd', json={"<option>": <value>}, auth=(
<user>, "<password>"))
>> print result.json()
```

Replace:

- **<ceph-mgr>** with the IP address or short host name of the node with the active **ceph-mgr** instance
- **<option>** with the option to modify; **pause, noup, nodown, noout, noin, nobackfill, norecover, noscrub, nodeep-scrub**
- **<value>** with **True** or **False**
- **<user>** with the user name
- **<password>** with the user's password

If you used a self-signed certificate, use the **verify=False** option:

```
$ python
>> import requests
>> result = requests.patch('https://<ceph-mgr>:8003/config/osd', json={"<option>": <value>}, auth=(
<user>, "<password>"), verify=False)
>> print result.json()
```

2.2.2. How Can I Change the OSD State?

This section describes how to use the RESTful plug-in to change the state of an OSD.

The curl Command

On the command line, use:

```
echo -En '{"<state>": <value>}' | curl --request PATCH --data @- --silent --user <user> 'https://<ceph-mgr>:8003/osd/<id>'
```

Replace:

- **<state>** with the state to change (**in** or **up**)
- **<value>** with **true** or **false**
- **<user>** with the user name
- **<ceph-mgr>** with the IP address or short host name of the node with the active **ceph-mgr** instance
- **<id>** with the ID of the OSD listed in the **osd** field

Enter the user's password when prompted.

If you used a self-signed certificate, use the **--insecure** option:

```
echo -En '{"<state>": <value>}' | curl --request PATCH --data @- --silent --insecure --user <user> 'https://<ceph-mgr>:8003/osd/<id>'
```

Python

In the Python interpreter, enter:

```
$ python
>> import requests
>> result = requests.patch('https://<ceph-mgr>:8003/osd/<id>', json={"<state>": <value>}, auth=("
<user>", "<password>"))
>> print result.json()
```

Replace:

- **<ceph-mgr>** with the IP address or short host name of the node with the active **ceph-mgr** instance
- **<state>** with the state to change (**in** or **up**)
- **<value>** with **True** or **False**
- **<id>** with the ID of the OSD listed in the **osd** field
- **<user>** with the user name
- **<password>** with the user's password

If you used a self-signed certificate, use the **verify=False** option:

```
$ python
>> import requests
```

```
>> result = requests.patch('https://<ceph-mgr>:8003/osd/<id>', json={"<state>": <value>}, auth=("  
<user>", "<password>"), verify=False)  
>> print result.json()
```

2.2.3. How Can I Reweight an OSD?

This section describes how to change the weight of an OSD.

The curl Command

On the command line, use:

```
echo -En '{"reweight": <value>}' | curl --request PATCH --data @- --silent --user <user>  
'https://<ceph-mgr>:8003/osd/<id>'
```

Replace:

- **<value>** with the new weight
- **<user>** with the user name
- **<ceph-mgr>** with the IP address or short host name of the node with the active **ceph-mgr** instance
- **<id>** with the ID of the OSD listed in the **osd** field

Enter the user's password when prompted.

If you used a self-signed certificate, use the **--insecure** option:

```
echo -En '{"reweight": <value>}' | curl --request PATCH --data @- --silent --insecure --user <user>  
'https://<ceph-mgr>:8003/osd/<id>'
```

Python

In the Python interpreter, enter:

```
$ python  
>> import requests  
>> result = requests.patch('https://<ceph-mgr>:8003/osd/<id>', json={"reweight": <value>}, auth=("  
<user>", "<password>"))  
>> print result.json()
```

Replace:

- **<ceph-mgr>** with the IP address or short host name of the node with the active **ceph-mgr** instance
- **<value>** with the new weight
- **<id>** with the ID of the OSD listed in the **osd** field
- **<user>** with the user name
- **<password>** with the user's password

If you used a self-signed certificate, use the **verify=False** option:

```
$ python
>> import requests
>> result = requests.patch('https://<ceph-mgr>:8003/osd/<id>', json={"reweight": <value>}, auth=("<user>", "<password>"), verify=False)
>> print result.json()
```

2.2.4. How Can I Change Information for a Pool?

This section describes how to use the RESTful plug-in to change information for a particular pool.

The curl Command

On the command line, use:

```
echo -En '{"<option>": <value>}' | curl --request PATCH --data @- --silent --user <user>
'https://<ceph-mgr>:8003/pool/<id>'
```

Replace:

- **<option>** with the option to modify
- **<value>** with the new value of the option
- **<user>** with the user name
- **<ceph-mgr>** with the IP address or short host name of the node with the active **ceph-mgr** instance
- **<id>** with the ID the pool listed in the **pool** field

Enter the user's password when prompted.

If you used a self-signed certificate, use the **--insecure** option:

```
echo -En '{"<option>": <value>}' | curl --request PATCH --data @- --silent --insecure --user <user>
'https://<ceph-mgr>:8003/pool/<id>'
```

Python

In the Python interpreter, enter:

```
$ python
>> import requests
>> result = requests.patch('https://<ceph-mgr>:8003/pool/<id>', json={"<option>": <value>}, auth=("<user>", "<password>"))
>> print result.json()
```

Replace:

- **<ceph-mgr>** with the IP address or short host name of the node with the active **ceph-mgr** instance
- **<id>** with the ID the pool listed in the **pool** field
- **<option>** with the option to modify
- **<value>** with the new value of the option

- **<user>** with the user name
- **<password>** with the user's password

If you used a self-signed certificate, use the **verify=False** option:

```
$ python
>> import requests
>> result = requests.patch('https://<ceph-mgr>:8003/pool/<id>', json={"<option>": <value>}, auth=("<user>", "<password>"), verify=False)
>> print result.json()
```

2.3. ADMINISTERING THE CLUSTER

This section describes how to use the RESTful plug-in to initialize scrubbing or deep scrubbing on an OSD, create a pool or remove data from a pool, remove requests, or create a request:

- [Section 2.3.1, "How Can I Run a Scheduled Process on an OSD?"](#)
- [Section 2.3.2, "How Can I Create a New Pool?"](#)
- [Section 2.3.3, "How Can I Remove Pools?"](#)
- [Section 2.3.4, "How Can I Remove All Finished Requests?"](#)
- [Section 2.3.5, "How Can I Remove a Particular Request?"](#)
- [Section 2.3.6, "How Can I Execute Administrative Commands?"](#)

2.3.1. How Can I Run a Scheduled Process on an OSD?

This section describes how to use the RESTful API to run scheduled processes, such as scrubbing or deep scrubbing, on an OSD.

The curl Command

On the command line, use:

```
echo -En '{"command": "<command>"}' | curl --request POST --data @- --silent --user <user>
'https://<ceph-mgr>:8003/osd/<id>/command'
```

Replace:

- **command** with the process (**scrub**, **deep-scrub**, or **repair**) you want to start. Verify it the process is supported on the OSD. See [Section 2.1.9, "How Can I Determine What Processes Can Be Scheduled on an OSD?"](#) for details.
- **<user>** with the user name
- **<ceph-mgr>** with the IP address or short host name of the node with the active **ceph-mgr** instance
- **<id>** with the ID of the OSD listed in the **osd** field

Enter the user's password when prompted.

If you used a self-signed certificate, use the **--insecure** option:

```
echo -En '{"command": "<command>"}' | curl --request POST --data @- --silent --insecure --user
<user> 'https://<ceph-mgr>:8003/osd/<id>/command'
```

Python

In the Python interpreter, enter:

```
$ python
>> import requests
>> result = requests.post('https://<ceph-mgr>:8003/osd/<id>/command', json={"command": "
<command>"}, auth=("<user>", "<password>"))
>> print result.json()
```

Replace:

- **<ceph-mgr>** with the IP address or short host name of the node with the active **ceph-mgr** instance
- **<id>** with the ID of the OSD listed in the **osd** field
- **command** with the process (**scrub**, **deep-scrub**, or **repair**) you want to start. Verify it the process is supported on the OSD. See [Section 2.1.9, “How Can I Determine What Processes Can Be Scheduled on an OSD?”](#) for details.
- **<user>** with the user name
- **<password>** with the user’s password

If you used a self-signed certificate, use the **verify=False** option:

```
$ python
>> import requests
>> result = requests.post('https://<ceph-mgr>:8003/osd/<id>/command', json={"command": "
<command>"}, auth=("<user>", "<password>"), verify=False)
>> print result.json()
```

2.3.2. How Can I Create a New Pool?

This section describes how to use the RESTful plug-in to create a new pool.

The curl Command

On the command line, use:

```
echo -En '{"name": "<name>", "pg_num": <number>}' | curl --request POST --data @- --silent --user
<user> 'https://<ceph-mgr>:8003/pool'
```

Replace:

- **<name>** with the name of the new pool
- **<number>** with the number of the placement groups
- **<user>** with the user name

- **<ceph-mgr>** with the IP address or short host name of the node with the active **ceph-mgr** instance

Enter the user's password when prompted.

If you used a self-signed certificate, use the **--insecure** option:

```
echo -En '{"name": "<name>", "pg_num": <number>}' | curl --request POST --data @- --silent --insecure --user <user> 'https://<ceph-mgr>:8003/pool'
```

Python

In the Python interpreter, enter:

```
$ python
>> import requests
>> result = requests.post('https://<ceph-mgr>:8003/pool', json={"name": "<name>", "pg_num": <number>}, auth=("<user>", "<password>"))
>> print result.json()
```

Replace:

- **<ceph-mgr>** with the IP address or short host name of the node with the active **ceph-mgr** instance
- **<name>** with the name of the new pool
- **<number>** with the number of the placement groups
- **<user>** with the user name
- **<password>** with the user's password

If you used a self-signed certificate, use the **verify=False** option:

```
$ python
>> import requests
>> result = requests.post('https://<ceph-mgr>:8003/pool', json={"name": "<name>", "pg_num": <number>}, auth=("<user>", "<password>"), verify=False)
>> print result.json()
```

2.3.3. How Can I Remove Pools?

This section describes how to use the RESTful plug-in to remove a pool.

This request is by default forbidden. To allow it, add the following parameter to the Ceph configuration guide.

```
mon_allow_pool_delete = true
```

The curl Command

On the command line, use:

```
curl --request DELETE --silent --user <user> 'https://<ceph-mgr>:8003/pool/<id>'
```

Replace:

- **<user>** with the user name
- **<ceph-mgr>** with the IP address or short host name of the node with the active **ceph-mgr** instance
- **<id>** with the ID the pool listed in the **pool** field

Enter the user's password when prompted.

If you used a self-signed certificate, use the **--insecure** option:

```
curl --request DELETE --silent --insecure --user <user> 'https://<ceph-mgr>:8003/pool/<id>'
```

Python

In the Python interpreter, enter:

```
$ python
>> import requests
>> result = requests.delete('https://<ceph-mgr>:8003/pool/<id>', auth=("<user>", "<password>"))
>> print result.json()
```

Replace:

- **<ceph-mgr>** with the IP address or short host name of the node with the active **ceph-mgr** instance
- **<id>** with the ID the pool listed in the **pool** field
- **<user>** with the user name
- **<password>** with the user's password

If you used a self-signed certificate, use the **verify=False** option:

```
$ python
>> import requests
>> result = requests.delete('https://<ceph-mgr>:8003/pool/<id>', auth=("<user>", "<password>"),
verify=False)
>> print result.json()
```

2.3.4. How Can I Remove All Finished Requests?

This section describes how to use the RESTful plug-in to remove all finished requests.

The curl Command

On the command line, use:

```
curl --request DELETE --silent --user <user> 'https://<ceph-mgr>:8003/request'
```

Replace:

- **<user>** with the user name

- **<ceph-mgr>** with the IP address or short host name of the node with the active **ceph-mgr** instance

Enter the user's password when prompted.

If you used a self-signed certificate, use the **--insecure** option:

```
curl --request DELETE --silent --insecure --user <user> 'https://<ceph-mgr>:8003/request'
```

Python

In the Python interpreter, enter:

```
$ python
>> import requests
>> result = requests.delete('https://<ceph-mgr>:8003/request', auth=("<user>", "<password>"))
>> print result.json()
```

Replace:

- **<ceph-mgr>** with the IP address or short host name of the node with the active **ceph-mgr** instance
- **<user>** with the user name
- **<password>** with the user's password

If you used a self-signed certificate, use the **verify=False** option:

```
$ python
>> import requests
>> result = requests.delete('https://<ceph-mgr>:8003/request', auth=("<user>", "<password>"),
verify=False)
>> print result.json()
```

2.3.5. How Can I Remove a Particular Request?

This section describes how to use the RESTful plug-in to remove a particular request from the database.

The **curl** Command

On the command line, use:

```
curl --request DELETE --silent --user <user> 'https://<ceph-mgr>:8003/request/<id>'
```

Replace:

- **<user>** with the user name
- **<ceph-mgr>** with the IP address or short host name of the node with the active **ceph-mgr** instance
- **<id>** with the ID of the request listed in the **id** field

Enter the user's password when prompted.

If you used a self-signed certificate, use the **--insecure** option:

```
curl --request DELETE --silent --insecure --user <user> 'https://<ceph-mgr>:8003/request/<id>'
```

Python

In the Python interpreter, enter:

```
$ python
>> import requests
>> result = requests.delete('https://<ceph-mgr>:8003/request/<id>', auth=("<user>", "<password>"))
>> print result.json()
```

Replace:

- **<ceph-mgr>** with the IP address or short host name of the node with the active **ceph-mgr** instance
- **<id>** with the ID of the request listed in the **id** field
- **<user>** with the user name
- **<password>** with the user's password

If you used a self-signed certificate, use the **verify=False** option:

```
$ python
>> import requests
>> result = requests.delete('https://<ceph-mgr>:8003/request/<id>', auth=("<user>", "<password>"),
verify=False)
>> print result.json()
```

2.3.6. How Can I Execute Administrative Commands?

Use the **/request** endpoint with the POST method to execute an administrative command. With this approach, you can use even commands that are not directly supported by the API, but are defined in the **src/mon/MonCommands.h** file in the Ceph source code. This is useful, for example, when building your own application that uses the API.

Python

1. Review **src/mon/MonCommands.h** and find the command you want to execute in the API, for example:

```
COMMAND("osd ls " \
        "name=epoch,type=CephInt,range=0,req=false", \
        "show all OSD ids", "osd", "r", "cli,rest")
```

The command is **osd ls**. The **name** bit specifies the name of an argument the command has, **type** specifies the type of values the argument takes, **range** specifies the range of accepted values of the argument, and **req** specifies whether the argument is required or not. In this example, the argument is **epoch**, the type of values the argument takes is integer, the range of accepted values is **0**, and the argument is optional.

2. In the Python interpreter, enter:

```
$ python
>> import requests
```

```
>> result = requests.post(
    'https://<ceph-mgr>:8003/request',
    json={'prefix': '<command>', 'argument':<value>},
    auth=("<user>", "<password>")
)
>> print result.json()
```

Replace:

- **<ceph-mgr>** with the IP address or short host name of the node with the active **ceph-mgr** instance
- **<command>** with the command listed in the **src/mon/MonCommands.h** file
- **<argument>** with any arguments of the command listed in the **src/mon/MonCommands.h** file, if the argument is optional, you can omit it
- **<value>** with the value of the argument
- **<user>** with the user name
- **<password>** with the user's password

For example, to use the **osd ls epoch 0** command. Replace **<command>** with **osd ls** and **<argument>** with **epoch** and **<value>** with **0**:

```
$ python
>> import requests
>> result = requests.post(
    'https://ceph-node1:8003/request',
    json={'prefix': 'osd ls', 'epoch': 0},
    auth=("ceph-user", "<password>")
)
>> print result.json()
```

If you used a self-signed certificate, use the **verify=False** option:

```
$ python
>> import requests
>> result = requests.post(
    'https://<ceph-mgr>:8003/request',
    json={'prefix': '<command>', 'optional_argument':<value>},
    auth=("<user>", "<password>"),
    verify=False
)
>> print result.json()
```

Additional Resources

- The [src/mon/MonCommands.h](#) file on GitHub

2.4. ADDITIONAL RESOURCES

- [Test API](#)

