



Global Leader in Software-Defined Storage.

NexentaStor 5.1.1 REST API QuickStart Guide

Date: January, 2018

Part Number: 3000-nxs-REST-API-5.1.1-000092-A

Copyright © 2018 Nexenta Systems™, ALL RIGHTS RESERVED

Notice: No part of this publication may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying and recording, or stored in a database or retrieval system for any purpose, without the express written permission of Nexenta Systems (hereinafter referred to as “Nexenta”).

Nexenta reserves the right to make changes to this document at any time without notice and assumes no responsibility for its use. Nexenta products and services only can be ordered under the terms and conditions of Nexenta Systems’ applicable agreements. All of the features described in this document may not be available currently. Refer to the latest product announcement or contact your local Nexenta Systems sales office for information on feature and product availability. This document includes the latest information available at the time of publication.

Nexenta, NexentaStor, NexentaEdge, NexentaFusion, and NexentaConnect are registered trademarks of Nexenta Systems in the United States and other countries. All other trademarks, service marks, and company names in this document are properties of their respective owners.

Contents

Preface	iv
1 About NexentaStor REST API	1
Using NexentaStor REST API	1
Synchronous and Asynchronous Calls	2
Using Specific Version of API in URLs	3
Determining Latest API Version Supported by An Appliance	4
Examples of Sending Requests	4
Swagger Support for 1.1.1 API Version	5
API to Collect Real Time Performance Metrics	5
List of NexentaStor Statistics	5
Prerequisites for Pushing Statistics	6
Pushing Statistics to Your Carbon Server using REST API	6

Preface

This documentation presents information specific to Nexenta products. The information is for reference purposes and is subject to change.

Intended Audience

This documentation is intended for Network Storage Administrators who are looking for details on enabling data at rest encryption and assumes that you have experience with data storage concepts, such as NAS, SAN, NFS, and ZFS.

This document also assumes that you have prior experience with NexentaStor and its command line interface used for creating, provisioning, and managing virtual and physical storage devices.

Documentation History

The following table lists the released revisions of this documentation.

Product Versions Applicable to this Documentation:

Revision	Date	Description
3000-nxs-REST-API-5.1.1-000092-A	January, 2018	GA

Contacting Support

Send your support questions and requests to support@nexenta.com.

Comments

Your comments and suggestions to improve this documentation are greatly appreciated. Send any feedback to doc.comments@nexenta.com and include the documentation title, number, and revision. Refer to specific pages, sections, and paragraphs whenever possible.

This page intentionally left blank

About NexentaStor REST API

Using NexentaStor REST API

The REST APIs provide programmatic access to managing a NexentaStor 5 appliance through HTTPS requests. It is a robust foundation for the new Nexenta Management Framework, a high-performance, multi-threaded, fault-tolerant management plane that provides a streamlined and simplified, storage-centric management experience.

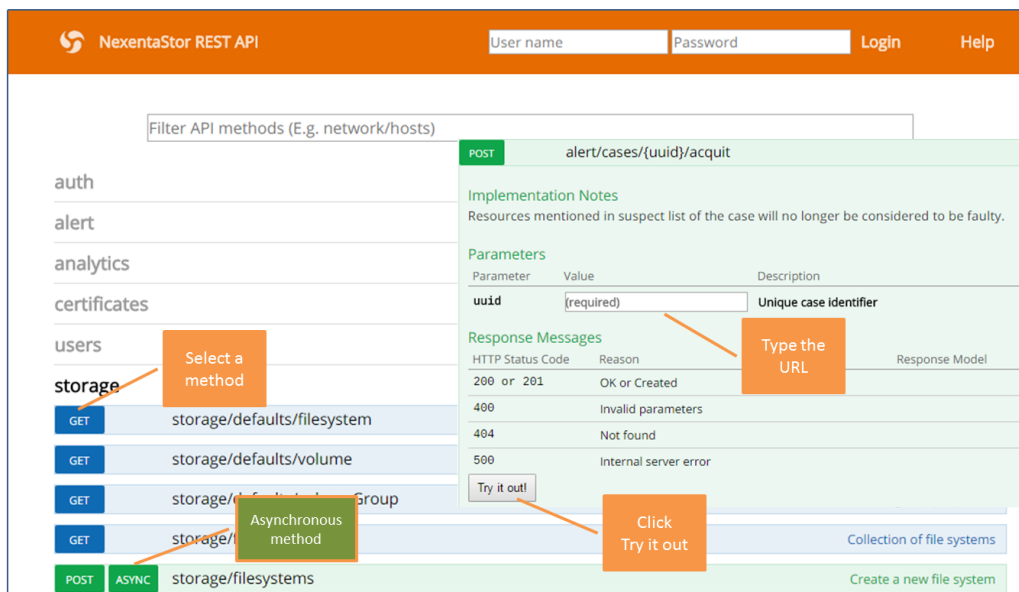
Note: The API online documentation is a development tool and access is disabled by default. Enabling access in a production environment is not recommended.

If you must enable access to the detailed REST API documentation, follow these steps:

1. Run this CLI command:


```
CLI@nexenta> config set rest.useSwagger=true (case-sensitive parameter)
```
2. Point your browser to `https://<staticIPaddress>:8443/docs` using the static IP address and password you set up during the NexentaStor installation.
3. Login to try out the APIs.
4. Explore specific APIs by selecting a method, entering part of the URL string, and clicking Try It Out.

Figure 1-1: NexentaStor REST API Interface.



Synchronous and Asynchronous Calls

NexentaStor REST API has two options of running the API calls. Some of the requests are to be done synchronously and some asynchronously. Synchronous modes are meant for requests that will get a quick response so this is basically applied for small requests. Whereas async mode is generally applied for requests that will take longer time to process. When you run an asynchronous call, NexentaStor receives the call, and returns a job ID to help you track the status of the job. This mode also returns a 202 code that implies the call is in progress. Async calls are available for POST, PUT, and DELETE methods.

Figure 1-2: Example of an Async Request.

The screenshot shows an API call interface with three main sections:

- Request URL:** `https://10.3.53.109:8443/storage/filesystems`
- Response Body:** A JSON object with a "links" array. One element in the array has "rel": "monitor" and "href": "/jobStatus/6aeda7c0-578b-11e7-8ffd-6b9ff89824da". The href value is highlighted with a red box, and a cloud-shaped callout labeled "job ID" points to it.
- Response Code:** A button labeled "202" with an arrow pointing to it from the "Response Body" section.

Click on the 202 button to get a response. The call responds when the request is complete. Or you can check the job status by copying and pasting the job ID in the jobId parameter of the "jobstatus" method and by reading the "done" parameter in the "Response Body"(Figure 4).

Figure 1-3: Example to Check an Async Job Status.

jobStatus		Show/Hide List Operations Expand Operations		
GET	jobStatus	Collection of asynchronous request statuses		
Parameter	Value	Description	Parameter Type	Data Type
offset	<input type="text"/>	The first record number of collection, that would be included in result	query	integer
limit	<input type="text"/>	The number of records of collection returned	query	integer
fields	<input type="text"/>	Comma delimited list of fields that would be included in result	query	string
jobId	4e4ce670-578c-11e7-8ffd-6b9ff89824da	ID of job status	query	string

Figure 1-4: Example of an Async Job Status.

```

Response Body
{
  "data": [
    {
      "jobId": "4e4ce670-578c-11e7-8ffd-6b9ff89824da",
      "progress": 0,
      "originalMethod": "POST",
      "originalUrl": "/storage/filesystems",
      "startTime": "2017-06-22T20:49:37.879Z",
      "done": true,
      "finishTime": "2017-06-22T20:49:38.239Z",
      "href": "/jobStatus/4e4ce670-578c-11e7-8ffd-6b9ff89824da"
    }
  ],
  "links": [
    {
      "rel": "self",
      "href": "/jobStatus?jobId=4e4ce670-578c-11e7-8ffd-6b9ff89824da"
    }
  ]
}

```

Using Specific Version of API in URLs

NexentaStor 5.1.1 includes version 1.1.1 of the REST API. The new version 1.1.1 is documented through the swagger web pages. For backwards compatibility, NexentaStor 5.1.1 continues to support version 1.0.4 of the API that shipped with NexentaStor 5.0.3 and also supports version 1.1.0 that shipped with 5.1. The following tables lists the versions of the REST API that are supported by specific versions of NexentaStor and the type of changes were made:

Table 1-1: API Version Corresponding to the NexentaStor version.

NexentaStor Version	API Version	Changes in API version
5.0.1	v1.0.1	Initial release
5.0.2	v1.0.2	Addition of new API calls
5.0.3	v1.0.4	Addition of new API calls
5.1	v1.1.0	Addition of new API calls
5.1.1	v1.1.1	Addition of new API calls, removal of obsolete calls, changes to output of existing API calls

Note: For backwards compatibility, API calls without an explicit version to a NexentaStor 5.1.1 are treated per version 1.0.4 of the API. To get access to the latest API methods supported in NexentaStor 5.1.1, the application must explicitly specify API version 1.1.1.

Determining Latest API Version Supported by An Appliance

Before using the API, verify from NexentaStor appliance that it supports one of the API versions the application uses.

❖ *To view the supported API version using curl:*

1. Send a request to an arbitrary REST API endpoint. (Use the endpoint for getting the information about the current user)

Example:

```
curl -k -v https://10.3.53.109:8443/auth/user
```

This example will respond with two headers:

- X-Latest-API-Version
 - X-API-Version
2. The header "X-Latest-API-Version" will appear only if you did not specify the API version (as in the above Example) and if you are not using the latest API version. This header represents the latest supported API version.
 3. The header "X-API-Version" indicates the API version used by default.

```
GET /auth/user HTTP/1.1
Host: 10.3.70.19:8443
User-Agent: curl/7.54.0
Accept: */*

HTTP/1.1 200 OK
X-XSS-Protection: 0
X-API-Version: 1.0.4
X-Latest-API-Version: 1.1.1
```

Examples of Sending Requests

- When sending an API request call using an URL if you need to use a specific version of API, for example 1.1.1, just add the version of the API in the URL.

Example:

```
https://<staticIPAddress>:8443/v1.1.1/settings/properties
```

Or

```
curl -H -k 'accept-version: 1.1.1' -v https://<staticIPAddress>:8443/settings/properties
```

- If you do not specify an API version number in the URL, as in this example, NexentaStor 5.1.1 REST server will default to API version 1.0.4

```
curl -k -v https://<staticIPAddress>:8443/settings/properties
```

Following are the different ways you can use the REST API in NexentaStor 5.1.1

- If you specify API version 1.1.1 in the URL and use an old REST call that no longer exists in 1.1.1, the system will respond with a message that the endpoint is not available in the current version.
- If you do not specify an API version in the URL and send a REST call that was removed or changed in the current API version 1.1.1, then you will get the older version results.
- When managing async jobs, use the same version of API for creating and viewing the results of a job. For example, if you create an ASYNC job (like create a pool) using a certain API version then this specific job should be viewed (managed) using the same version. Avoid mixing API versions for async jobs.

Note: To ensure an application continues to work with future versions of NexentaStor, it should always explicitly call out the API version it expects the NexentaStor system to use.

Swagger Support for 1.1.1 API Version

Once you install NexentaStor 5.1.1 and enable swagger using this CLI command, `CLI@nexenta> config set rest.useSwagger=true` (case-sensitive parameter) and point your browser to `https://<staticIpAddress>:8443/docs`, you will be pointed to the updated API version 1.1.1

Note: Swagger does not support the older versions of API. Only the latest API version 1.1.1 can be viewed from Swagger.

API to Collect Real Time Performance Metrics

List of NexentaStor Statistics

Currently NexentaStor generates the following statistics. All statistics that are currently exposed have predefined intervals of 15 secs, 30 secs, 1 minute, 30 mins, etc which means the new data is refreshed only at those predefined rates.

- Over all I/O statistics for NFS/SMB protocols (IOPS/bandwidth/R-W avg. latency);
- per NFS/SMB share I/O statistics (IOPS/bandwidth/R-W avg. latency);
- ARC stats (hit ratio);
- per CPU utilisation;
- per disk I/O utilisation (run service time, wait service time);
- per disk I/O latency (in form of histogram, per latency range);
- per FC LUNs I/O stats (IOPS/bandwidth/average latency);
- overall I/O statistics for FC protocol (IOPS/bandwidth/average latency);
- per FC target I/O stats (IOPS/bandwidth/average latency);
- per FC initiator I/O statistics (IOPS/bandwidth/R-W avg. latency);
- per dataset space usage (used/referenced/available);

- per HPR service I/O stats (send/receive speed);
- per iSCSI LUNs I/O stats (IOPS/bandwidth/average latency);
- overall I/O statistics for iSCSI protocol (IOPS/bandwidth/average latency);
- per iSCSI target I/O stats (IOPS/bandwidth/average latency);
- per iSCSI initiator I/O statistics (IOPS/bandwidth/R-W avg. latency);
- per NIC I/O statistics (bandwidth, read/write utilisation);
- per NFS protocol (v3/v4) overall I/O statistics (IOPS);
- per NFS/SMB client I/O statistics (IOPS/bandwidth/R-W avg. latency);
- per zpool generic stats (total space, allocated space, available space, deduplication/compression ratio, bytes used by snapshots);
- per zpool I/O stats (IOPS/bandwidth/R-W avg. latency).

To view and analyze these statistics in an intuitive graphical format, follow the steps explained in the next two sections.

Prerequisites for Pushing Statistics

To investigate aspects of the appliance components operation and performance, ensure that you have the following components set up in your environment.

- Carbon server(s)
 - To expose these statistics NexentaStor pushes the analytics to a Carbon server in your environment. See appropriate documentation for information on setting up a Carbon server if you don't have one yet.
- Carbon metrics plain-text protocol that is supported by Carbon
 - For the transfer of data statistics, you must support Carbon metrics plain-text protocol.
- For more information on the carbon format and graphite, refer to <http://graphite.readthedocs.io/en/latest/index.html>

Now that you have a Carbon server in your environment, read the contents below to push the statistics collection to your Carbon server.

Pushing Statistics to Your Carbon Server using REST API

To help get your statistics collection into Carbon server, you simply have to register the IP of your Carbon instance using NexentaStor REST API. Once you register the IP address of the Carbon instance(s), NexentaStor pushes the statistics in Carbon compatible format. This registration process facilitates generation of summaries of capacity and performance metrics in real time and historical time. You may view the real/historical-time visual data for a selected NexentaStor appliance.

You can register more than one Carbon analytics servers using the following REST API.

- To register the IP address of the Carbon analytics consumer, do the following:
 1. Enable access to the detailed REST API using this CLI command:

```
CLI@nexenta> config set rest.useSwagger=true (case-sensitive parameter)
```

- Point your browser to `https://<staticIPAddress>:8443/docs` using the static IP address and password you set up during the NexentaStor installation.
- In the NexentaStor REST API interface, select “analytics”.

The screenshot displays the REST API interface for the 'analytics' endpoint. The 'PUT' method is selected for the endpoint 'analytics/consumers/carbon'. The 'Payload' section shows a JSON body with an empty object {} and a data type 'Model Schema' for 'setCarbonInfo'. The 'Response Messages' table lists HTTP status codes 200 (OK), 400 (Invalid parameters), and 500 (Internal server error). A 'Try it out!' button is visible at the bottom.

- Select PUT Request for “Updating the carbon analytics configuration”.
- Add the list of IP Carbon analytics server(s) in the servers (array[string]).

```
"servers": ["IP address1:port", "IP address1:port", ...]
```
- For NexentaStor to push the statistics in Carbon compatible format to the Carbon analytics server, select “enabled”:true.

Example:

```
{
  "enabled": true,
  "servers": ["IP address:port"]
}
```

- Click “Try it out!” button to add the Carbon servers.
- To stop pushing the NexentaStor statistics to the Carbon analytics server, use the following payload.

```
{
  "enabled": false
}
```

Now that you have pushed the statistics collection to the Carbon server, your statistics collection can be rendered in graphical format using a webapp that is supported by Carbon server.