Automate using REST APIs

ONTAP tools for VMware vSphere 10

NetApp
September 30, 2025

This PDF was generated from https://docs.netapp.com/us-en/ontap-tools-vmware-vsphere-102/automation/overview-rest-apis.html on September 30, 2025. Always check docs.netapp.com for the latest.

Table of Contents

Automate using REST APIs . . . 1
- Overview of REST APIs . . . 1
  - How to access ONTAP tools for VMware vSphere REST API . . . 1
  - Network considerations . . . 1
  - ONTAP tools for VMware vSphere API online documentation page . . . 1
  - Custom software and tools . . . 1
  - Access ONTAP tools for VMware vSphere API reference documentation through the Swagger UI . . . 2
- Get started with the REST API . . . 2
  - Hello World . . . 2
  - How to access ONTAP tools for VMware vSphere REST API . . . 3
  - Network considerations . . . 3
  - Input variables controlling an API request . . . 3
  - HTTP methods . . . 3
  - Request headers . . . 4
  - Request body . . . 4
  - Filtering objects . . . 4
  - Requesting specific object fields . . . 5
  - Sorting objects in the output set . . . 5
  - Pagination when retrieving objects in a collection . . . 5
  - Size properties . . . 6

Automate using REST APIs

Overview of REST APIs

REST APIs can be used to perform several ONTAP tools for VMware vSphere management operations. REST APIs are exposed through the Swagger web page.

You can access the Swagger web page available at *https://loadbalancerIP:8443/* to display the REST API documentation, as well as to manually issue an API call.

All the APIs have request body and examples mentioned in the swagger page. The workflows and examples provided in this section are for reference purposes only.

How to access ONTAP tools for VMware vSphere REST API

You can access the ONTAP REST API in several different ways.

Network considerations

You can connect to the REST API through the following interfaces:

- Cluster management LIF
- Node management LIF
- SVM management LIF

The LIF you choose to use should be configured to support the HTTPS management protocol. Also, the firewall configuration in your network should allow the HTTPS traffic.

You should always use a cluster management LIF. This will load balance the API requests across all the nodes and avoid nodes that are offline or experiencing connectivity issues. If you have multiple cluster management LIFs configured, they are all equivalent regarding access to the REST API.

ONTAP tools for VMware vSphere API online documentation page

You can access the Swagger from the hyperlink in the support page of the NetAPP ONTAP tools for VMware vSphere plug-in.

The format of the URL used to access the documentation page for the most recent version of the API is:

`https://<loadbalancer_ip_address>/docs/api`

Custom software and tools

You can access ONTAP tools for VMware vSphere API using any of several different programming languages and tools. Popular choices include Python, Java, Curl, and PowerShell. A program, script, or tool that uses the API acts as a REST web services client. Using a programming language enables a deeper understanding of the API and provides an opportunity to automate ONTAP tools for VMware vSphere administration.

The format of the base URL used to directly access the most recent version of the API is:

`https://<loadbalancer_ip_address>/api`

To access a specific API version where multiple versions are supported, the format of the URL is:

`https://<loadbalancer_ip_address>/api/v1`

Access ONTAP tools for VMware vSphere API reference documentation through the Swagger UI

You can access the ONTAP REST API documentation through the Swagger UI at your local ONTAP system.

Before you begin

You should have the following:

- IP address or host name of the ONTAP cluster management LIF
- Username and password for an account with authority to access the ONTAP REST API

Steps

1. Type the URL in your browser and press **Enter**: *https://<ip_address>/docs/api*
2. Sign in using the ONTAP account

   The ONTAP API documentation page is displayed with the API calls organized in major resource categories at the bottom.

3. As an example of an individual API call, scroll down to the **cluster** category and click **GET /cluster**.

Get started with the REST API

You can quickly get started using ONTAP tools for VMware vSphere REST API. Accessing the API provides some perspective before you begin using it with the more complex workflow processes on a live setup.

Hello World

You can run a simple command on your system to get started using ONTAP tools for VMware vSphere REST API and confirm its availability.

Before you begin

- Ensure that the Curl utility is available on your system.
- IP address or host name of ONTAP tools for VMware vSphere server
- Username and password for an account with authority to access ONTAP tools for VMware vSphere REST API.

> If your credentials include special characters, you need to format them in a way that is acceptable to Curl based on the shell you are using. For example, you can insert a backslash before each special character or wrap the entire `username:password` string in single quotes.

Step

At the command line interface, run the following to retrieve the plug-in information:

```
curl -X GET -u username:password -k
"https://<ip_address>/api/hosts?fields=IncludePluginInfo"
```

Example:

```
curl -X GET -u admin:password -k
"'https://10.225.87.97/api/hosts?fields=IncludePluginInfo"
```

How to access ONTAP tools for VMware vSphere REST API

You can access the ONTAP REST API in several different ways.

Network considerations

You can connect to the REST API through the following interfaces:

- Cluster management LIF
- Node management LIF
- SVM management LIF

The LIF you choose to use should be configured to support the HTTPS management protocol. Also, the firewall configuration in your network should allow the HTTPS traffic.

> You should always use a cluster management LIF. This will load balance the API requests across all the nodes and avoid nodes that are offline or experiencing connectivity issues. If you have multiple cluster management LIFs configured, they are all equivalent regarding access to the REST API.

Input variables controlling an API request

You can control how an API call is processed through parameters and variables set in the HTTP request.

HTTP methods

The HTTP methods supported by ONTAP tools for VMware vSphere REST API are shown in the following table.

> Not all the HTTP methods are available at each of the REST endpoints.

HTTP method	Description
GET	Retrieves object properties on a resource instance or collection.
POST	Creates a new resource instance based on the supplied input.
DELETE	Deletes an existing resource instance.
PUT	Modifies an existing resource instance.

Request headers

You should include several headers in the HTTP request.

Content-type

If the request body includes JSON, this header should be set to *application/json*.

Accept

This header should be set to *application/json*.

Authorization

Basic authentication should be set with the username and password encoded as a base64 string.

Request body

The content of the request body varies depending on the specific call. The HTTP request body consists of one of the following:

- JSON object with input variables
- Empty

Filtering objects

When issuing an API call that uses GET, you can limit or filter the returned objects based on any attribute. For example, you can specify an exact value to match:

```
<field>=<query value>
```

In addition to an exact match, other operators are available to return a set of objects over a range of values. ONTAP tools for VMware vSphere REST API supports the filtering operators shown in the table below.

Operator	Description
=	Equal to
<	Less than
>	Greater than
⇐	Less than or equal to
>=	Greater than or equal to
UPDATE	Or
!	Not equal to
*	Greedy wildcard

You can also return a collection of objects based on whether a specific field is set or not set by using the **null** keyword or its negation **!null** as part of the query.

Any fields that are not set are generally excluded from matching queries.

Requesting specific object fields

By default, issuing an API call using GET returns only the attributes that uniquely identify the object or objects. This minimum set of fields acts as a key for each object and varies based on the object type. You can select additional object properties using the `fields` query parameter in the following ways:

Common or standard fields

Specify **fields=*** to retrieve the most commonly used object fields. These fields are typically maintained in local server memory or require little processing to access. These are the same properties returned for an object after using GET with a URL path key (UUID).

All fields

Specify **fields=**** to retrieve all the object fields, including those requiring additional server processing to access.

Custom field selection

Use **fields=<field_name>** to specify the exact field you want. When requesting multiple fields, the values should be separated using commas without spaces.

> As a best practice, you should always identify the specific fields you want. You should only retrieve the set of common fields or all fields when needed. Which fields are classified as common, and returned using *fields=**, is determined by NetApp based on internal performance analysis. The classification of a field might change in future releases.

Sorting objects in the output set

The records in a resource collection are returned in the default order defined by the object. You can change the order using the `order_by` query parameter with the field name and sort direction as follows:

```
order_by=<field name> asc|desc
```

For example, you can sort the type field in descending order followed by id in ascending order:

```
order_by=type desc, id asc
```

- If you specify a sort field but do not provide a direction, the values are sorted in ascending order.
- When including multiple parameters, you should separate the fields with a comma.

Pagination when retrieving objects in a collection

When issuing an API call using GET to access a collection of objects of the same type, ONTAP tools for VMware vSphere attempts to return as many objects as possible based on two constraints. You can control each of these constraints using additional query parameters on the request. The first constraint reached for a specific GET request terminates the request and therefore limits the number of records returned.

> If a request ends before iterating over all the objects, the response contains the link needed to retrieve the next batch of records.

Limiting the number of objects

By default, ONTAP tools for VMware vSphere returns a maximum of 10,000 objects for a GET request. You can change this limit using the *max_records* query parameter. For example:

```
max_records=20
```

The number of objects returned can be less than the maximum in effect, based on the related time constraint as well as the total number of objects in the system.

Limiting the time used to retrieve the objects

By default, ONTAP tools for VMware vSphere returns as many objects as possible within the time allowed for the GET request. The default timeout is 15 seconds. You can change this limit using the *return_timeout* query parameter. For example:

```
return_timeout=5
```

The number of objects returned can be less than the maximum in effect, based on the related constraint on the number of objects as well as the total number of objects in the system.

Narrowing the result set

If needed, you can combine these two parameters with additional query parameters to narrow the result set. For example, the following returns up to 10 EMS events generated after the specified time:

```
time⇒ 2018-04-04T15:41:29.140265Z&max_records=10
```

You can issue multiple requests to page through the objects. Each subsequent API call should use a new time value based on the latest event in the last result set.

Size properties

The input values used with some API calls as well as certain query parameters are numeric. Rather than provide an integer in bytes, you can optionally use a suffix as shown in the following table.

Suffix	Description
KB	KB Kilobytes (1024 bytes) or kibibytes
MB	MB Megabytes (KB x 1024 bytes) or mebibytes
GB	GB Gigabytes (MB x 1024 bytes) or gibibytes
TB	TB Terabytes (GB x 1024 byes) or tebibytes
PB	PB Petabytes (TB x 1024 byes) or pebibytes

Copyright information

Copyright © 2025 NetApp, Inc. All Rights Reserved. Printed in the U.S. No part of this document covered by copyright may be reproduced in any form or by any means—graphic, electronic, or mechanical, including photocopying, recording, taping, or storage in an electronic retrieval system—without prior written permission of the copyright owner.

Software derived from copyrighted NetApp material is subject to the following license and disclaimer:

THIS SOFTWARE IS PROVIDED BY NETAPP “AS IS” AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, WHICH ARE HEREBY DISCLAIMED. IN NO EVENT SHALL NETAPP BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

NetApp reserves the right to change any products described herein at any time, and without notice. NetApp assumes no responsibility or liability arising from the use of products described herein, except as expressly agreed to in writing by NetApp. The use or purchase of this product does not convey a license under any patent rights, trademark rights, or any other intellectual property rights of NetApp.

The product described in this manual may be protected by one or more U.S. patents, foreign patents, or pending applications.

LIMITED RIGHTS LEGEND: Use, duplication, or disclosure by the government is subject to restrictions as set forth in subparagraph (b)(3) of the Rights in Technical Data -Noncommercial Items at DFARS 252.227-7013 (FEB 2014) and FAR 52.227-19 (DEC 2007).

Data contained herein pertains to a commercial product and/or commercial service (as defined in FAR 2.101) and is proprietary to NetApp, Inc. All NetApp technical data and computer software provided under this Agreement is commercial in nature and developed solely at private expense. The U.S. Government has a non-exclusive, non-transferrable, nonsublicensable, worldwide, limited irrevocable license to use the Data only in connection with and in support of the U.S. Government contract under which the Data was delivered. Except as provided herein, the Data may not be used, disclosed, reproduced, modified, performed, or displayed without the prior written approval of NetApp, Inc. United States Government license rights for the Department of Defense are limited to those rights identified in DFARS clause 252.227-7015(b) (FEB 2014).

Trademark information

NETAPP, the NETAPP logo, and the marks listed at http://www.netapp.com/TM are trademarks of NetApp, Inc. Other company and product names may be trademarks of their respective owners.