



Automate using the REST API

ONTAP tools for VMware vSphere 10

NetApp

November 17, 2025

This PDF was generated from <https://docs.netapp.com/us-en/ontap-tools-vmware-vsphere-103/automation/learn-rest-api.html> on November 17, 2025. Always check docs.netapp.com for the latest.

Table of Contents

- Automate using the REST API 1
 - Learn about the ONTAP tools for VMware vSphere 10 REST API 1
 - REST web services foundation 1
 - ONTAP tools Manager environment 1
 - Implementation details for the ONTAP tools for VMware vSphere 10 REST API 2
 - How to access the REST API. 2
 - HTTP details 3
 - Authentication 4
 - Synchronous and asynchronous requests. 4
 - Your first ONTAP tools for VMware vSphere 10 REST API call 4
 - Before you begin 4
 - Step 1: Acquire an access token 5
 - Step 2: Issue the REST API call. 5
 - API reference for the ONTAP tools for VMware vSphere 10 REST API 6

Automate using the REST API

Learn about the ONTAP tools for VMware vSphere 10 REST API

ONTAP tools for VMware vSphere 10 is a set of tools for virtual machine lifecycle management. It includes a robust REST API you can use as part of your automation processes.

REST web services foundation

Representational State Transfer (REST) is a style for creating distributed web applications including the design of web services APIs. It establishes a set of technologies for exposing server-based resources and managing their states.

Resources and state representation

Resources are the foundational components of a REST web services application. There are two important initial tasks when designing a REST API:

- Identify the system or server-based resources
- Define the resource states and associated state transition operations

Client applications can display and change the resource states through well-defined message flows.

HTTP messages

Hypertext Transfer Protocol (HTTP) is the protocol used by the web services client and server to exchange messages about the resources. It follows the CRUD model based on the generic operations create, read, update, and delete. The HTTP protocol includes request and response headers as well as response status codes.

JSON data formatting

While there are several message formats available, the most popular option is JavaScript Object Notation (JSON). JSON is an industry standard for representing simple data structures in plain text and is used to transfer state information describing the resources and desired actions.

Security

Security is an important aspect of a REST API. In addition to the Transport Layer Security (TLS) protocol used to protect the HTTP traffic over the network, the ONTAP tools for VMware vSphere 10 REST API also uses access tokens for authentication. You need to acquire an access token and use it on subsequent API calls.

Support for asynchronous requests

The ONTAP tools for VMware vSphere 10 REST API performs most requests synchronously, returning a status code when the operation is complete. It also supports asynchronous processing for tasks that require a longer time to complete.

ONTAP tools Manager environment

There are several aspects of the ONTAP tools Manager environment you should consider.

Virtual machine

ONTAP tools for VMware vSphere 10 is deployed using the vSphere remote plugin architecture. The software, including support for the REST API, runs in a separate virtual machine.

ONTAP tools IP address

ONTAP tools for VMware vSphere 10 exposes a single IP address which provides a gateway to the capabilities of the virtual machine. You need to provide the address during initial configuration and it's assigned to an internal load balancer component. The address is used by the ONTAP tools Manager user interface as well as to access the Swagger documentation page and REST API directly.

Two REST APIs

In addition to the ONTAP tools for VMware vSphere 10 REST API, the ONTAP cluster has its own REST API. ONTAP tools Manager uses the ONTAP REST API as a client to perform storage related tasks. It's important to keep in mind these two APIs are separate and distinct. For more information, refer to [ONTAP automation](#).

Implementation details for the ONTAP tools for VMware vSphere 10 REST API

While REST establishes a common set of technologies and best practices, the exact implementation of each API can vary based on the design choices. You should be familiar with how the ONTAP tools for VMware vSphere 10 REST API is designed before using it.

The REST API includes several resource categories such as vCenters and Aggregates. Review the [API reference](#) for more information.

How to access the REST API

You can access the ONTAP tools for VMware vSphere 10 REST API through the ONTAP tools load balancer IP address along with the port. There are several parts to the complete URL, including:

- ONTAP tools IP address and port
- API version
- Resource category
- Specific resource

You need to configure the IP address during initial configuration and the port is always 8443. Also, for a specific ONTAP tools for VMware vSphere 10 instance the first part of the URL is constant. Only the resource category and specific resource vary across the endpoints.



The IP address and port values in the examples below are for illustration purposes only. You need to change these values for your environment.

Example to access authentication services

```
https://10.61.25.34:8443/virtualization/api/v1/auth/login
```

This URL can be used to request an access token using the POST method.

Example to list the vCenter servers

```
https://10.61.25.34:8443/virtualization/api/v1/vcenters
```

This URL can be used to request a list of the defined vCenter server instances using the GET method.

HTTP details

The ONTAP tools for VMware vSphere 10 REST API uses HTTP and related parameters to act on the resource instances and collections. Details of the HTTP implementation are presented below.

HTTP methods

The HTTP methods or verbs supported by the REST API are presented in the table below.

Method	CRUD	Description
GET	Read	Retrieves object properties for a resource instance or collection. This is considered a list operation when used with a collection.
POST	Create	Creates a new resource instance based on the input parameters.
PUT	Update	Updates an entire resource instance with the supplied JSON request body. Key values that are not user-modifiable are preserved.
PATCH	Update	Requests a set of selected changes in the request be applied to the resource instance.
DELETE	Delete	Deletes an existing resource instance.

Request and response headers

The following table summarizes the most important HTTP headers used with the REST API.

Header	Type	Usage notes
Accept	Request	This is the type of content the client application can accept. Valid values include <code>*/*</code> or <code>application/json</code> .
x-auth	Request	Contains an access token identifying the user issuing the request through the client application.
Content-Type	Response	Returned by the server based on the <code>Accept</code> request header.

HTTP status codes

The HTTP status codes used by the REST API are described below.

Code	Meaning	Description
200	OK	Indicates success for calls that do not create a new resource instance.
201	Created	An object has been successfully created with a unique identifier for the resource instance.
202	Accepted	The request has been accepted and a background job created to perform the request.
204	No content	The request was successful although no content was returned.
400	Bad request	The request input is not recognized or is inappropriate.
401	Unauthorized	The user is not authorized and must authenticate.

Code	Meaning	Description
403	Forbidden	Access is denied due to an authorization error.
404	Not found	The resource referred to in the request does not exist.
409	Conflict	An attempt to create an object failed because the object already exists.
500	Internal error	A general internal error occurred at the server.

Authentication

Authentication of a client to the REST API is performed using an access token. The relevant characteristics of the token and authentication process include:

- The client must request a token using ONTAP tools Manager admin credentials (username and password).
- Tokens are formatted as a JSON Web Token (JWT).
- Each token expires after 60 minutes.
- API requests from a client must include the token in the `x-auth` request header.

Refer to [Your first REST API call](#) for an example of requesting and using an access token.

Synchronous and asynchronous requests

Most REST API calls complete quickly and therefore run synchronously. That is, they return a status code (such as 200) after a request has been completed. Requests that take longer to complete run asynchronously using a background job.

After issuing an API call that runs asynchronously, the server returns a 202 HTTP status code. This indicates the request has been accepted but not yet completed. You can query the background job to determine its status including success or failure.

Asynchronous processing is used for several types of long running operations, including datastore and vVol operations. Refer to the job manager category of the REST API at the Swagger page for more information.

Your first ONTAP tools for VMware vSphere 10 REST API call

You can issue an API call using curl to get started with the ONTAP tools for VMware vSphere 10 REST API.

Before you begin

You should review the required information and parameters needed in the curl examples.

Required information

You need the following:

- ONTAP tools for VMware vSphere 10 IP address or FQDN as well as the port
- Credentials for the ONTAP tools Manager admin (username and password)

Parameters and variables

The curl examples presented below include Bash style variables. You can set these variables in the Bash environment or manually update them before issuing the commands. If you set the variables, the shell will substitute the values into each command before it's executed. The variables are described in the table below.

Variable	Description
\$FQDN_IP_PORT	The fully qualified domain name or IP address of the ONTAP tools Manager along with the port number.
\$MYUSER	Username for the ONTAP tools Manager account.
\$MYPASSWORD	Password associated with the ONTAP tools Manager username.
\$ACCESS_TOKEN	The access token issued by the ONTAP tools Manager.

The following commands and output at the Linux CLI illustrate how a variable can be set and displayed:

```
FQDN_IP_PORT=172.14.31.224:8443
echo $FQDN_IP
172.14.31.224:8443
```

Step 1: Acquire an access token

You need to acquire an access token to use the REST API. An example of how to request an access token is presented below. You should substitute in the appropriate values for your environment.

```
curl --request POST \
--location "https://$FQDN_IP_PORT/virtualization/api/v1/auth/login" \
--header "Content-Type: application/json" \
--header "Accept: */*" \
-d '{"username": "$MYUSER", "password": "$MYPASSWORD}"
```

Copy and save the the access token provided in the response.

Step 2: Issue the REST API call

After you have an access token, you can use curl to issue a REST API call. Include the access token acquired in the first step.

Curl example

```
curl --request GET \
--location "https://$FQDN_IP_PORT/virtualization/api/v1/vcenters" \
--header "Accept: */*" \
--header "x-auth: $ACCESS_TOKEN"
```

The JSON response includes a list of the VMware vCenter instances configured to the ONTAP tools Manager.

API reference for the ONTAP tools for VMware vSphere 10 REST API

The ONTAP tools for VMware vSphere 10 REST API reference contains details about all the API calls. This reference is helpful when developing automation applications.

You can access the ONTAP tools for VMware vSphere 10 REST API documentation online through the Swagger user interface. You need the IP address or FQDN of the ONTAP tools for VMware vSphere 10 gateway service as well as the port.

Steps

1. Type the following URL into your browser substituting the appropriate IP address and port combination for the variable and press **Enter**.

```
https://$FQDN_IP_PORT/
```

Example

```
https://10.61.25.33:8443/
```

2. As an example of an individual API call, scroll down to the **vCenters** category and select **GET** next to the endpoint `/virtualization/api/v1/vcenters`

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.