



# **Basic operational characteristics**

## **SnapCenter Software**

Soumik Das  
June 09, 2021

This PDF was generated from [https://docs.netapp.com/us-en/snapcenter/sc-automation/concept\\_basic\\_operational\\_characterstics.html](https://docs.netapp.com/us-en/snapcenter/sc-automation/concept_basic_operational_characterstics.html) on June 24, 2021. Always check docs.netapp.com for the latest.

# Table of Contents

- Basic operational characteristics ..... 1
  - Request and response API transaction ..... 1
  - Support for CRUD operations ..... 1
  - Object identifiers ..... 1
  - Object instances and collections ..... 1
  - Synchronous and asynchronous operations ..... 1
  - Security ..... 2

# Basic operational characteristics

While REST establishes a common set of technologies and best practices, the details of each API can vary based on the design choices.

## Request and response API transaction

Every REST API call is performed as an HTTP request to the SnapCenter Server system which generates an associated response to the client. This request and response pair is considered an API transaction.

Before using the API, you should be familiar with the input variables available to control a request and the contents of the response output.

## Support for CRUD operations

Each of the resources available through the SnapCenter REST API is accessed based on the CRUD model:

- Create
- Read
- Update
- Delete

For some of the resources, only a subset of the operations is supported.

## Object identifiers

Each resource instance or object is assigned a unique identifier when it is created. In most cases, the identifier is a 128-bit UUID. These identifiers are globally unique within a specific SnapCenter Server.

After issuing an API call that creates a new object instance, a URL with the associated ID is returned to the caller in the location header of the HTTP response. You can extract the identifier and use it on subsequent calls when referring to the resource instance.



The content and internal structure of the object identifiers can change at any time. You should only use the identifiers on the applicable API calls as needed when referring to the associated objects.

## Object instances and collections

Depending on the resource path and HTTP method, an API call can apply to a specific object instance or a collection of objects.

## Synchronous and asynchronous operations

SnapCenter performs an HTTP request received from a client either synchronously or asynchronously.

## **Synchronous processing**

SnapCenter performs the request immediately and responds with an HTTP status code of 200 or 201 if it is successful.

Every request using the method GET is always performed synchronously. In addition, requests that use POST are designed to run synchronously if they are expected to complete in less than two seconds.

## **Asynchronous processing**

If an asynchronous request is valid, SnapCenter creates a background task to process the request and a job object to anchor the task. The HTTP status code 202 is returned to the caller along with the job object. You should retrieve the state of the job to determine success or failure.

Requests that use the methods POST and DELETE are designed to run asynchronously if they are expected to take more than two seconds to complete.

## **Security**

The security provided with the REST API is based primarily on the existing security features available with SnapCenter. The following security is used by the API:

### **Transport Layer Security**

All traffic sent over the network between the SnapCenter Server and client is typically encrypted using TLS, based on the SnapCenter configuration settings.

### **HTTP authentication**

At an HTTP level, basic authentication is used for the API transactions. An HTTP header with the user name and password in a base64 string is added to each request.

## Copyright Information

Copyright © 2021 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.

RESTRICTED RIGHTS LEGEND: Use, duplication, or disclosure by the government is subject to restrictions as set forth in subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.277-7103 (October 1988) and FAR 52-227-19 (June 1987).

## 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.