



About the Element software API

Element Software

NetApp
June 11, 2021

This PDF was generated from https://docs.netapp.com/us-en/element-software/api/reference_element_api_request_object_members.html on June 11, 2021. Always check docs.netapp.com for the latest.

Table of Contents

- About the Element software API 1
- Find more information 1
- Request object members 1
- Response object members 2
- Request endpoints 2
- API authentication 3
- Asynchronous methods 4
- Attributes 4

About the Element software API

The Element API is based on the JSON-RPC protocol over HTTPS. JSON-RPC is a simple text-based RPC protocol based on the lightweight JSON data-interchange format. Client libraries are available for all major programming languages.

You can make API requests via HTTPS POST requests to the API endpoint. The body of the POST request is a JSON-RPC request object. The API does not currently support batch requests (multiple request objects in a single POST). When submitting API requests, you must use "application/json-rpc" as the content-type of the request, and ensure that the body is not form-encoded.



The Element web UI makes use of the API methods described in this document. You can monitor API operations in the UI by enabling the API Log; this enables you to see the methods that are being issued to the system. You can enable both requests and responses to see how the system replies to the methods that are issued.

Unless stated otherwise, all date strings in the API responses are in UTC+0 format.



When the storage cluster is heavily loaded or you submit many consecutive API requests with no intervening delay, a method might fail and return the error "xDBVersionMismatch". If this happens, retry the method call.

- [Request object members](#)
- [Response object members](#)
- [Request endpoints](#)
- [API authentication](#)
- [Asynchronous methods](#)
- [Attributes](#)

Find more information

- [NetApp SolidFire Resources Page](#)
- [Documentation for earlier versions of NetApp SolidFire and Element products](#)

Request object members

Each Element software API request has the following basic parts:

Name	Description	Type	Default value	Required
method	Name of the method to be invoked.	string	None	Yes

Name	Description	Type	Default value	Required
parameters	Object containing the parameters to the method being invoked. Named parameters are required. Positional parameters (passed as an array) are not allowed.	JSON object	{}	No
id	Identifier used to match the request to response, returned in the result.	string or integer	{}	No

Response object members

Each Element software API response body has the following basic parts:

Name	Description	Type
result	The object returned by the method. The system returns an object with named members corresponding to the documented return value for the method. This member is not present if an error has occurred.	JSON object
error	The object returned when an error occurs. This member is present only if an error has occurred.	Object
id	An identifier used to match the request to response, as provided in the request.	string or integer
unusedParameters	A warning message that at least one incorrect parameter has been passed to the API method and has not been used.	Object

Request endpoints

There are three types of request endpoints used in the API (storage cluster, storage cluster creation, and per-node). You should always use the latest endpoint supported by your version of Element software.

The three request endpoints in the API are designated in the following ways:

Cluster API methods

The HTTPS endpoint for storage-cluster-wide API requests is `https://<mvip>/json-rpc/<api-version>`, where:

- `<mvip>` is the management virtual IP address for the storage cluster.
- `<api-version>` is the version of the API you are using.

Cluster creation and bootstrap API methods

The HTTPS endpoint for creating a storage cluster and accessing bootstrap API requests is `https://<nodeIP>/json-rpc/<api-version>`, where:

- `<nodeIP>` is the IP address of the node you are adding to the cluster.
- `<api-version>` is the version of the API you are using.

Per-node API methods

The HTTPS endpoint for individual storage node API requests is `https://<nodeIP>:442/json-rpc/<api-version>`, where:

- `<nodeIP>` is the management IP address of the storage node; 442 is the port the HTTPS server is running on.
- `<api-version>` is the version of the API you are using.

Find more information

- [NetApp SolidFire Resources Page](#)
- [Documentation for earlier versions of NetApp SolidFire and Element products](#)

API authentication

You can authenticate with the system when using the API by including an HTTP Basic authentication header with all API requests. If you omit authentication information, the system rejects the unauthenticated request with an HTTP 401 response. The system supports HTTP Basic authentication over TLS.

Use the cluster admin account for API authentication.

Find more information

- [NetApp SolidFire Resources Page](#)
- [Documentation for earlier versions of NetApp SolidFire and Element products](#)

Asynchronous methods

Some API methods are asynchronous, which means that the operation they perform might not be complete when the method returns. Asynchronous methods return a handle that you can query to see the status of the operation; status information for some operations might include a percentage of completion.

When you query an asynchronous operation, its result can be one of the following types:

- **DriveAdd**: The system is adding a drive to the cluster.
- **BulkVolume**: The system is performing a copy operation between volumes, such as a backup or restore.
- **Clone**: The system is cloning a volume.
- **DriveRemoval**: The system is copying data from a drive in preparation to remove it from the cluster.
- **RtffiPendingNode**: The system is installing compatible software on a node before adding it to the cluster.

Note the following points when using asynchronous methods or obtaining the status of a running asynchronous operation:

- Asynchronous methods are indicated in the individual method documentation.
- Asynchronous methods return an “asyncHandle”, which is a handle that is known by the issuing API method. You can use the handle to poll for the status or result of the asynchronous operation.
- You can obtain the result of individual asynchronous methods with the `GetAsyncResult` method. When you use `GetAsyncResult` to query a completed operation, the system returns the result and automatically purges the result from the system. When you use `GetAsyncResult` to query an incomplete operation, the system returns the result but does not purge it.
- You can obtain the status and results of all running or completed asynchronous methods using the `ListAsyncResults` method. In this case, the system does not purge results for completed operations.

Find more information

- [NetApp SolidFire Resources Page](#)
- [Documentation for earlier versions of NetApp SolidFire and Element products](#)

Attributes

Many of the API requests and responses use objects as well as simple types. Objects are a collection of key-value pairs, where the value is a simple type or possibly another object. Attributes are custom name-value pairs that can be set by the user in JSON objects. Some methods enable you to add attributes when creating or modifying objects.

There is a 1000-byte limit on encoded attribute objects.

Object member

This object contains the following member:

Name	Description	Type
attributes	List of name-value pairs in JSON object format.	JSON object

Request example

The following request example uses the AddClusterAdmin method:

```
{
  "method": "AddClusterAdmin",
  "params": {
    "username": "joeadmin",
    "password": "68!5Aru268)$",
    "access": [
      "volume",
      "reporting"
    ],
    "attributes": {
      "name1": "value1",
      "name2": "value2",
      "name3": "value3"
    }
  }
}
```

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.