



Using the Grid Management API

StorageGRID

NetApp
October 03, 2025

Table of Contents

- Using the Grid Management API 1
 - Top-level resources 1
 - Grid Management API operations 1
 - Issuing API requests 3
 - Grid Management API versioning 5
 - Determining which API versions are supported in the current release 5
 - Specifying an API version for a request 6
 - Protecting against Cross-Site Request Forgery (CSRF) 6
 - Using the API if single sign-on is enabled 7
 - Signing in to the API if single sign-on is enabled 7
 - Signing out of the API if single sign-on is enabled 12

Using the Grid Management API

You can perform system management tasks using the Grid Management REST API instead of the Grid Manager user interface. For example, you might want to use the API to automate operations or to create multiple entities, such as users, more quickly.

The Grid Management API uses the Swagger open source API platform. Swagger provides an intuitive user interface that allows developers and non-developers to perform real-time operations in StorageGRID with the API.

Top-level resources

The Grid Management API provides the following top-level resources:

- `/grid`: Access is restricted to Grid Manager users and is based on the configured group permissions.
- `/org`: Access is restricted to users who belong to a local or federated LDAP group for a tenant account. For details, see the information about using tenant accounts.
- `/private`: Access is restricted to Grid Manager users and is based on the configured group permissions. These APIs are intended for internal use only and are not publicly documented. These APIs are also subject to change without notice.

Related information

[Use a tenant account](#)

[Prometheus: Query basics](#)

Grid Management API operations

The Grid Management API organizes the available API operations into the following sections.

- **accounts** — Operations to manage storage tenant accounts, including creating new accounts and retrieving storage usage for a given account.
- **alarms** — Operations to list current alarms (legacy system), and return information about the health of the grid, including the current alerts and a summary of node connection states.
- **alert-history** — Operations on resolved alerts.
- **alert-receivers** — Operations on alert notification receivers (email).
- **alert-rules** — Operations on alert rules.
- **alert-silences** — Operations on alert silences.
- **alerts** — Operations on alerts.
- **audit** — Operations to list and update the audit configuration.
- **auth** — Operations to perform user session authentication.

The Grid Management API supports the Bearer Token Authentication Scheme. To sign in, you provide a username and password in the JSON body of the authentication request (that is, `POST /api/v3/authorize`). If the user is successfully authenticated, a security token is returned. This token

must be provided in the header of subsequent API requests ("Authorization: Bearer *token*").



If single sign-on is enabled for the StorageGRID system, you must perform different steps to authenticate. See “Authenticating in to the API if single sign-on is enabled.”

See “Protecting against Cross-Site Request Forgery” for information on improving authentication security.

- **client-certificates** — Operations to configure client certificates so that StorageGRID can be accessed securely using external monitoring tools.
- **config** — Operations related to the product release and versions of the Grid Management API. You can list the product release version and the major versions of the Grid Management API supported by that release, and you can disable deprecated versions of the API.
- **deactivated-features** — Operations to view features that might have been deactivated.
- **dns-servers** — Operations to list and change configured external DNS servers.
- **endpoint-domain-names** — Operations to list and change endpoint domain names.
- **erasure-coding** — Operations on Erasure Coding profiles.
- **expansion** — Operations on expansion (procedure-level).
- **expansion-nodes** — Operations on expansion (node-level).
- **expansion-sites** — Operations on expansion (site-level).
- **grid-networks** — Operations to list and change the Grid Network List.
- **grid-passwords** — Operations for grid password management.
- **groups** — Operations to manage local Grid Administrator Groups and to retrieve federated Grid Administrator Groups from an external LDAP server.
- **identity-source** — Operations to configure an external identity source and to manually synchronize federated group and user information.
- **ilm** — Operations on information lifecycle management (ILM).
- **license** — Operations to retrieve and update the StorageGRID license.
- **logs** — Operations for collecting and downloading log files.
- **metrics** — Operations on StorageGRID metrics including instant metric queries at a single point in time and range metric queries over a range of time. The Grid Management API uses the Prometheus systems monitoring tool as the backend data source. For information about constructing Prometheus queries, see the Prometheus web site.



Metrics that include *private* in their names are intended for internal use only. These metrics are subject to change between StorageGRID releases without notice.

- **node-health** — Operations on node health status.
- **ntp-servers** — Operations to list or update external Network Time Protocol (NTP) servers.
- **objects** — Operations on objects and object metadata.
- **recovery** — Operations for the recovery procedure.
- **recovery-package** — Operations to download the Recovery Package.
- **regions** — Operations to view and create regions.
- **s3-object-lock** — Operations on global S3 Object Lock settings.

- **server-certificate** — Operations to view and update Grid Manager server certificates.
- **snmp** — Operations on the current SNMP configuration.
- **traffic-classes** — Operations for traffic classification policies.
- **untrusted-client-network** — Operations on the untrusted Client Network configuration.
- **users** — Operations to view and manage Grid Manager users.

Issuing API requests

The Swagger user interface provides complete details and documentation for each API operation.

What you'll need

- You must be signed in to the Grid Manager using a supported browser.
- You must have specific access permissions.



Any API operations you perform using the API Docs webpage are live operations. Be careful not to create, update, or delete configuration data or other data by mistake.

Steps

1. Select **Help > API Documentation** from the Grid Manager header.
2. Select the desired operation.

When you expand an API operation, you can see the available HTTP actions, such as GET, PUT, UPDATE, and DELETE.

3. Select an HTTP action to see the request details, including the endpoint URL, a list of any required or optional parameters, an example of the request body (when required), and the possible responses.

GET

/grid/groups Lists Grid Administrator Groups

🔒

Parameters

Try it out

Name	Description
type string <small>(query)</small>	filter by group type Available values : local, federated <div>— ▼</div>
limit integer <small>(query)</small>	maximum number of results Default value : 25 <div>25</div>
marker string <small>(query)</small>	marker-style pagination offset (value is Group's URN) <div>marker - marker-style pagination offset (value</div>
includeMarker boolean <small>(query)</small>	if set, the marker element is also returned <div>— ▼</div>
order string <small>(query)</small>	pagination order (desc requires marker) Available values : asc, desc <div>— ▼</div>

Responses

Response content type application/json ▼

Code	Description
200	successfully retrieved Example Value Model <pre>{ "responseTime": "2021-03-29T14:22:19.673Z", "status": "success", "apiVersion": "3.3", "deprecated": false, "data": [{ "displayName": "Developers", </pre>

- Determine if the request requires additional parameters, such as a group or user ID. Then, obtain these values. You might need to issue a different API request first to get the information you need.
- Determine if you need to modify the example request body. If so, you can click **Model** to learn the requirements for each field.
- Click **Try it out**.
- Provide any required parameters, or modify the request body as required.
- Click **Execute**.
- Review the response code to determine if the request was successful.

Grid Management API versioning

The Grid Management API uses versioning to support non-disruptive upgrades.

For example, this Request URL specifies version 3 of the API.

```
https://hostname_or_ip_address/api/v3/authorize
```

The major version of the Tenant Management API is bumped when changes are made that are **not compatible** with older versions. The minor version of the Tenant Management API is bumped when changes are made that **are compatible** with older versions. Compatible changes include the addition of new endpoints or new properties. The following example illustrates how the API version is bumped based on the type of changes made.

Type of change to API	Old version	New version
Compatible with older versions	2.1	2.2
Not compatible with older versions	2.1	3.0

When you install StorageGRID software for the first time, only the most recent version of the Grid Management API is enabled. However, when you upgrade to a new feature release of StorageGRID, you continue to have access to the older API version for at least one StorageGRID feature release.



You can use the Grid Management API to configure the supported versions. See the “config” section of the Swagger API documentation for more information. You should deactivate support for the older version after updating all Grid Management API clients to use the newer version.

Outdated requests are marked as deprecated in the following ways:

- The response header is "Deprecated: true"
- The JSON response body includes "deprecated": true
- A deprecated warning is added to nms.log. For example:

```
Received call to deprecated v1 API at POST "/api/v1/authorize"
```

Determining which API versions are supported in the current release

Use the following API request to return a list of the supported API major versions:

```
GET https://{IP-Address}/api/versions
{
  "responseTime": "2019-01-10T20:41:00.845Z",
  "status": "success",
  "apiVersion": "3.0",
  "data": [
    2,
    3
  ]
}
```

Specifying an API version for a request

You can specify the API version using a path parameter (`/api/v3`) or a header (`Api-Version: 3`). If you provide both values, the header value overrides the path value.

```
curl https://[IP-Address]/api/v3/grid/accounts

curl -H "Api-Version: 3" https://[IP-Address]/api/grid/accounts
```

Protecting against Cross-Site Request Forgery (CSRF)

You can help protect against Cross-Site Request Forgery (CSRF) attacks against StorageGRID by using CSRF tokens to enhance authentication that uses cookies. The Grid Manager and Tenant Manager automatically enable this security feature; other API clients can choose whether to enable it when they sign in.

An attacker that can trigger a request to a different site (such as with an HTTP form POST) can cause certain requests to be made using the signed-in user's cookies.

StorageGRID helps protect against CSRF attacks by using CSRF tokens. When enabled, the contents of a specific cookie must match the contents of either a specific header or a specific POST body parameter.

To enable the feature, set the `csrfToken` parameter to `true` during authentication. The default is `false`.

```
curl -X POST --header "Content-Type: application/json" --header "Accept: application/json" -d "{
  \"username\": \"MyUserName\",
  \"password\": \"MyPassword\",
  \"cookie\": true,
  \"csrfToken\": true
}" "https://example.com/api/v3/authorize"
```

When `true`, a `GridCsrfToken` cookie is set with a random value for sign-ins to the Grid Manager, and the

`AccountCsrfToken` cookie is set with a random value for sign-ins to the Tenant Manager.

If the cookie is present, all requests that can modify the state of the system (POST, PUT, PATCH, DELETE) must include one of the following:

- The `X-Csrf-Token` header, with the value of the header set to the value of the CSRF token cookie.
- For endpoints that accept a form-encoded body: A `csrfToken` form-encoded request body parameter.

See the online API documentation for additional examples and details.



Requests that have a CSRF token cookie set will also enforce the "`Content-Type: application/json`" header for any request that expects a JSON request body as an additional protection against CSRF attacks.

Using the API if single sign-on is enabled

If single sign-on (SSO) has been enabled for your StorageGRID system, you cannot use the standard Authenticate API requests to sign in to and sign out of the Grid Management API or the Tenant Management API.

Signing in to the API if single sign-on is enabled

If single sign-on (SSO) has been enabled, you must issue a series of API requests to obtain an authentication token from AD FS that is valid for the Grid Management API or the Tenant Management API.

What you'll need

- You know the SSO username and password for a federated user who belongs to a StorageGRID user group.
- If you want to access the Tenant Management API, you know the tenant account ID.

About this task

To obtain an authentication token, you can use one of the following examples:

- The `storagegrid-ssoauth.py` Python script, which is located in the StorageGRID installation files directory (`./rpms` for Red Hat Enterprise Linux or CentOS, `./debs` for Ubuntu or Debian, and `./vsphere` for VMware).
- An example workflow of curl requests.

The curl workflow might time out if you perform it too slowly. You might see the error: A valid SubjectConfirmation was not found on this Response.



The example curl workflow does not protect the password from being seen by other users.

If you have a URL-encoding issue, you might see the error: Unsupported SAML version.

Steps

1. Select one of the following methods to obtain an authentication token:
 - Use the `storagegrid-ssoauth.py` Python script. Go to step 2.

- Use curl requests. Go to step 3.
2. If you want to use the `storagegrid-ssoauth.py` script, pass the script to the Python interpreter and run the script.

When prompted, enter values for the following arguments:

- The SSO username
- The domain where StorageGRID is installed
- The address for StorageGRID
- If you want to access the Tenant Management API, enter tenant account ID.

```
python3 /tmp/storagegrid-ssoauth.py
saml_user: my-sso-username
saml_domain: my-domain
sg_address: storagegrid.example.com
tenant_account_id: 12345
Enter the user's SAML password:
*****

*****

StorageGRID Auth Token: 56eb07bf-21f6-40b7-afob-5c6cacfb25e7
```

The StorageGRID authorization token is provided in the output. You can now use the token for other requests, similar to how you would use the API if SSO was not being used.

3. If you want to use curl requests, use the following procedure.
- a. Declare the variables needed to sign in.

```
export SAMLUSER='my-sso-username'
export SAMLPASSWORD='my-password'
export SAMLDOMAIN='my-domain'
export TENANTACCOUNTID='12345'
export STORAGEGRID_ADDRESS='storagegrid.example.com'
export AD_FS_ADDRESS='adfs.example.com'
```



To access the Grid Management API, use 0 as `TENANTACCOUNTID`.

- b. To receive a signed authentication URL, issue a POST request to `/api/v3/authorize-saml`, and remove the additional JSON encoding from the response.

This example shows a POST request for a signed authentication URL for `TENANTACCOUNTID`. The results will be passed to `python -m json.tool` to remove the JSON encoding.

```
curl -X POST "https://$STORAGEGRID_ADDRESS/api/v3/authorize-saml" \
  -H "accept: application/json" -H "Content-Type: application/json" \
  --data "{\"accountId\": \"$TENANTACCOUNTID\"}" | python -m
json.tool
```

The response for this example includes a signed URL that is URL-encoded, but it does not include the additional JSON-encoding layer.

```
{
  "apiVersion": "3.0",
  "data":
  "https://adfs.example.com/adfs/ls/?SAMLRequest=fZHLbsIwEEV%2FJTuv7...
sSl%2BfQ33cvfwA%3D&RelayState=12345",
  "responseTime": "2018-11-06T16:30:23.355Z",
  "status": "success"
}
```

- c. Save the `SAMLRequest` from the response for use in subsequent commands.

```
export SAMLREQUEST='fZHLbsIwEEV%2FJTuv7...sSl%2BfQ33cvfwA%3D'
```

- d. Get a full URL that includes the client request ID from AD FS.

One option is to request the login form using the URL from the previous response.

```
curl
  "https://$AD_FS_ADDRESS/adfs/ls/?SAMLRequest=$SAMLREQUEST&RelayState=
  $TENANTACCOUNTID" | grep 'form method="post" id="loginForm"'
```

The response includes the client request ID:

```
<form method="post" id="loginForm" autocomplete="off"
novalidate="novalidate" onKeyPress="if (event && event.keyCode == 13)
Login.submitLoginRequest();" action="/adfs/ls/?
SAMLRequest=fZHRTOMwFIZfhh...UJikvo77sXPw%3D%3D&RelayState=12345&clie
nt-request-id=00000000-0000-0000-ee02-0080000000de" >
```

- e. Save the client request ID from the response.

```
export SAMLREQUESTID='00000000-0000-0000-ee02-0080000000de'
```

- f. Send your credentials to the form action from the previous response.

```
curl -X POST
"https://$AD_FS_ADDRESS/adfs/ls/?SAMLRequest=$SAMLREQUEST&RelayState=
$TENANTACCOUNTID&client-request-id=$SAMLREQUESTID" \
--data
"UserName=$SAMLUSER@$SAMLDOMAIN&Password=$SAMLPASSWORD&AuthMethod=For
msAuthentication" --include
```

AD FS returns a 302 redirect, with additional information in the headers.



If multi-factor authentication (MFA) is enabled for your SSO system, the form post will also contain the second password or other credentials.

```
HTTP/1.1 302 Found
Content-Length: 0
Content-Type: text/html; charset=utf-8
Location:
https://adfs.example.com/adfs/ls/?SAMLRequest=fZHRTomwFIZfhh...UJikvo
77sXPw%3D%3D&RelayState=12345&client-request-id=00000000-0000-0000-
ee02-0080000000de
Set-Cookie: MSISAuth=AAEAADAvsHpXk6ApV...pmP0aEiNtJvWY=; path=/adfs;
HttpOnly; Secure
Date: Tue, 06 Nov 2018 16:55:05 GMT
```

- g. Save the MSISAuth cookie from the response.

```
export MSISAuth='AAEAADAvsHpXk6ApV...pmP0aEiNtJvWY='
```

- h. Send a GET request to the specified location with the cookies from the authentication POST.

```
curl
"https://$AD_FS_ADDRESS/adfs/ls/?SAMLRequest=$SAMLREQUEST&RelayState=
$TENANTACCOUNTID&client-request-id=$SAMLREQUESTID" \
--cookie "MSISAuth=$MSISAuth" --include
```

The response headers will contain AD FS session information for later logout usage, and the response body contains the SAMLResponse in a hidden form field.

```

HTTP/1.1 200 OK
Cache-Control: no-cache,no-store
Pragma: no-cache
Content-Length: 5665
Content-Type: text/html; charset=utf-8
Expires: -1
Server: Microsoft-HTTPAPI/2.0
P3P: ADFS doesn't have P3P policy, please contact your site's admin
for more details
Set-Cookie:
SamlSession=a3dpbnRlcnMtUHJpbWFyeS1BZG1pbi0xNzgmRmFsc2Umcng4NnJDZmFKV
XfXVWx3bk1lMnFuUSUzZCUzZCYmJiYmXzE3MjAyZTA5LTNmMDgtNDRkZC04Yzg5LTQ3ND
UxYzA3ZjkzYw==; path=/adfs; HttpOnly; Secure
Set-Cookie: MSISAuthenticated=MTEvNy8yMDE4IDQ6MzI6NTkgUE0=;
path=/adfs; HttpOnly; Secure
Set-Cookie: MSISLoopDetectionCookie=MjAxOC0xMS0wNzoxNjoxMjoxOVpcMQ==;
path=/adfs; HttpOnly; Secure
Date: Wed, 07 Nov 2018 16:32:59 GMT

<form method="POST" name="hiddenform"
action="https://storagegrid.example.com:443/api/saml-response">
  <input type="hidden" name="SAMLResponse"
value="PHNhbWxwOlJlc3Bvb3N...1scDpSZXNwb25zZT4=" /><input
type="hidden" name="RelayState" value="12345" />

```

- i. Save the SAMLResponse from the hidden field:

```
export SAMLResponse='PHNhbWxwOlJlc3Bvb3N...1scDpSZXNwb25zZT4='
```

- j. Using the saved SAMLResponse, make a StorageGRID/api/saml-response request to generate a StorageGRID authentication token.

For RelayState, use the tenant account ID or use 0 if you want to sign in to the Grid Management API.

```

curl -X POST "https://$STORAGEGRID_ADDRESS:443/api/saml-response" \
-H "accept: application/json" \
--data-urlencode "SAMLResponse=$SAMLResponse" \
--data-urlencode "RelayState=$TENANTACCOUNTID" \
| python -m json.tool

```

The response includes the authentication token.

```
{
  "apiVersion": "3.0",
  "data": "56eb07bf-21f6-40b7-af0b-5c6cacfb25e7",
  "responseTime": "2018-11-07T21:32:53.486Z",
  "status": "success"
}
```

k. Save the authentication token in the response as MYTOKEN.

```
export MYTOKEN="56eb07bf-21f6-40b7-af0b-5c6cacfb25e7"
```

You can now use MYTOKEN for other requests, similar to how you would use the API if SSO was not being used.

Signing out of the API if single sign-on is enabled

If single sign-on (SSO) has been enabled, you must issue a series of API requests to sign out of the Grid Management API or the Tenant Management API.

About this task

If required, you can sign out of the StorageGRID API simply by logging out from your organization's single logout page. Or, you can trigger single logout (SLO) from StorageGRID, which requires a valid StorageGRID bearer token.

Steps

1. To generate a signed logout request, pass cookie "sso=true" to the SLO API:

```
curl -k -X DELETE "https://$STORAGEGRID_ADDRESS/api/v3/authorize" \
-H "accept: application/json" \
-H "Authorization: Bearer $MYTOKEN" \
--cookie "sso=true" \
| python -m json.tool
```

A logout URL is returned:

```
{
  "apiVersion": "3.0",
  "data":
    "https://ads.example.com/ads/ls/?SAMLRequest=fZDNboMwEIRfhZ...HcQ%3D%3D",
  "responseTime": "2018-11-20T22:20:30.839Z",
  "status": "success"
}
```

2. Save the logout URL.

```
export  
LOGOUT_REQUEST='https://adfs.example.com/adfs/ls/?SAMLRequest=fZDNboMwEIRfhZ...HcQ%3D%3D'
```

3. Send a request to the logout URL to trigger SLO and to redirect back to StorageGRID.

```
curl --include "$LOGOUT_REQUEST"
```

The 302 response is returned. The redirect location is not applicable to API-only logout.

```
HTTP/1.1 302 Found  
Location: https://$STORAGEGRID_ADDRESS:443/api/saml-  
logout?SAMLResponse=fVLLasMwEPwVo7ss%...%23rsa-sha256  
Set-Cookie: MSISSignoutProtocol=U2FtbA==; expires=Tue, 20 Nov 2018  
22:35:03 GMT; path=/adfs; HttpOnly; Secure
```

4. Delete the StorageGRID bearer token.

Deleting the StorageGRID bearer token works the same way as without SSO. If cookie "sso=true" is not provided, the user is logged out of StorageGRID without affecting the SSO state.

```
curl -X DELETE "https://$STORAGEGRID_ADDRESS/api/v3/authorize" \  
-H "accept: application/json" \  
-H "Authorization: Bearer $MYTOKEN" \  
--include
```

A 204 No Content response indicates the user is now signed out.

```
HTTP/1.1 204 No Content
```

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.