Use the API

StorageGRID software

NetApp
January 14, 2026

Table of Contents

Use the API

Use 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.

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 Use a tenant account.
- `/private`: Access is restricted to Grid Manager users and is based on the configured group permissions. The private APIs are subject to change without notice. StorageGRID private endpoints also ignore the API version of the request.

Issue API requests

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.

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

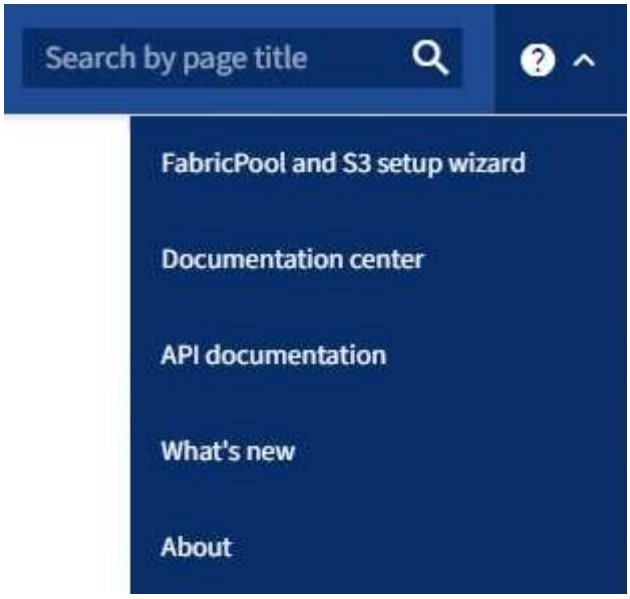
Before you begin

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

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

Steps

1. From the Grid Manager header, select the help icon and select **API documentation**.



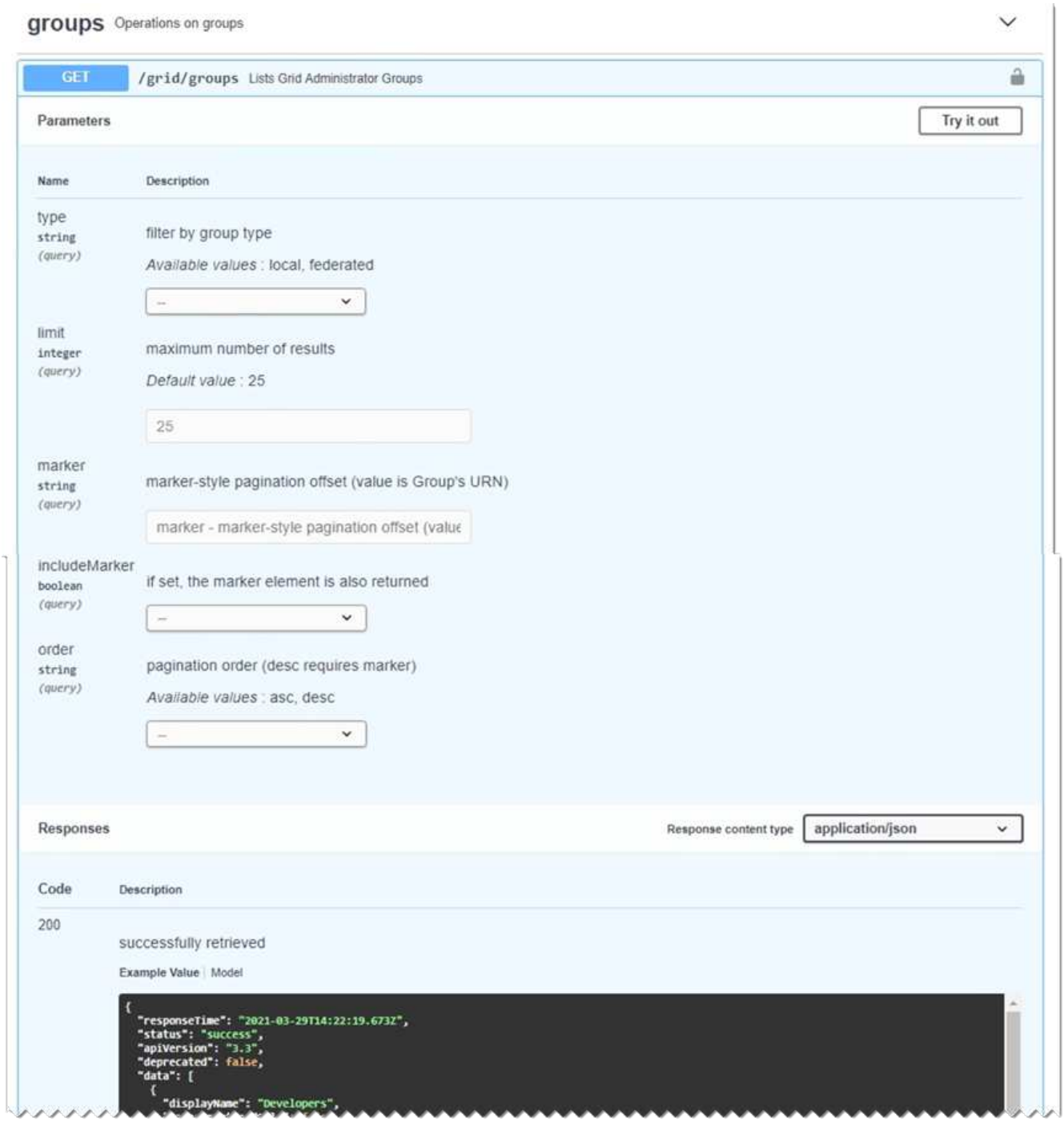
2. To perform an operation with the private API, select **Go to private API documentation** on the StorageGRID Management API page.

   The private APIs are subject to change without notice. StorageGRID private endpoints also ignore the API version of the request.

3. Select the desired operation.

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

4. 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.



5. 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.
6. Determine if you need to modify the example request body. If so, you can select **Model** to learn the requirements for each field.
7. Select **Try it out**.
8. Provide any required parameters, or modify the request body as required.
9. Select **Execute**.
10. Review the response code to determine if the request was successful.

Grid Management API operations

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

> This list only includes operations available in the public API.

- **accounts**: Operations to manage storage tenant accounts, including creating new accounts and retrieving storage usage for a given account.
- **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*"). The token expires after 16 hours.

  > 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 about 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.
- **drive-details**: Operations on drives for specific storage appliance models.
- **endpoint-domain-names**: Operations to list and change S3 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).
- **in-progress-procedures**: Retrieves the maintenance procedures that are currently in progress.
- **license**: Operations to retrieve and update the StorageGRID license.
- **logs**: Operations for collecting and downloading log files.v
- **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-details**: Operations on node details.
- **node-health**: Operations on node health status.
- **node-storage-state**: Operations on node storage 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.
- **storage-watermarks**: Storage node watermarks.
- **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.

Grid Management API versioning

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

For example, this Request URL specifies version 4 of the API.

`https://hostname_or_ip_address/api/v4/authorize`

The major version of the API is bumped when changes are made that are *not compatible* with older versions. The minor version of the 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
	3.0	4.0

When you install StorageGRID software for the first time, only the most recent version of the 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 configure the supported versions. See the **config** section of the Swagger API documentation for the Grid Management API for more information. You should deactivate support for the older version after updating all 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 v2 API at POST "/api/v2/authorize"
```

Determine which API versions are supported in the current release

Use the `GET /versions` API request to return a list of the supported API major versions. This request is located in the **config** section of the Swagger API documentation.

```
GET https://{{IP-Address}}/api/versions
{
  "responseTime": "2023-06-27T22:13:50.750Z",
  "status": "success",
  "apiVersion": "4.0",
  "data": [
    2,
    3,
    4
  ]
}
```

Specify an API version for a request

You can specify the API version using a path parameter (`/api/v4`) or a header (`Api-Version: 4`). If you provide both values, the header value overrides the path value.

```
curl https://[IP-Address]/api/v4/grid/accounts

curl -H "Api-Version: 4" https://[IP-Address]/api/grid/accounts
```

Protect 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.

Use the API if single sign-on is enabled

Use the API if single sign-on is enabled (Active Directory)

If you have configured and enabled single sign-on (SSO) and you use Active Directory as the SSO provider, you must issue a series of API requests to obtain an authentication token that is valid for the Grid Management API or the Tenant Management API.

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

These instructions apply if you are using Active Directory as the SSO identity provider.

Before you begin

- 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 RHEL, `./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 method. Enter ADFS or adfs.
   - The SSO username
   - The domain where StorageGRID is installed
   - The address for StorageGRID
   - The tenant account ID, if you want to access the Tenant Management API.

```
python3 storagegrid-ssoauth.py
sso_method: adfs
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=fZHRToMwFIZfhb...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=FormsAuthentication" --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=fZHRToMwFIZfhb...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
XFxVWx3bkl1MnFuUSUzZCUzZCYmJiYmXzE3MjAyZTA5LThmMDgtNDRkZC04Yzg5LTQ3ND
UxYzA3ZjkzYw==; path=/adfs; HttpOnly; Secure
Set-Cookie: MSISAuthenticated=MTEvNy8yMDE4IDQ6MzI6NTkgUE0=;
path=/adfs; HttpOnly; Secure
Set-Cookie: MSISLoopDetectionCookie=MjAxOC0xMS0wNzoxNjozMjo1OVpcMQ==;
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="PHNhbWxwOlJlc3BvbnN...1scDpSZXNwb25zZT4=" /><input
type="hidden" name="RelayState" value="12345" />
```

i. Save the `SAMLResponse` from the hidden field:

```
export SAMLResponse='PHNhbWxwOlJlc3BvbnN...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.

Sign 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. These instructions apply if you are using Active Directory as the SSO identity provider

About this task

If required, you can sign out of the StorageGRID API 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://adfs.example.com/adfs/ls/?SAMLRequest=fZDNboMwEIRfhZ...HcQ%3D%3
D",
    "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
```

Use the API if single sign-on is enabled (Entra ID)

If you have configured and enabled single sign-on (SSO) and you use Entra ID as the SSO provider, you can use two example scripts to obtain an authentication token that is valid for the Grid Management API or the Tenant Management API.

Sign in to the API if Entra ID single sign-on is enabled

These instructions apply if you are using Entra ID as the SSO identity provider

Before you begin

- You know the SSO email address 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 the following example scripts:

- The `storagegrid-ssoauth-azure.py` Python script
- The `storagegrid-ssoauth-azure.js` Node.js script

Both scripts are located in the StorageGRID installation files directory (`./rpms` for RHEL, `./debs` for Ubuntu or Debian, and `./vsphere` for VMware).

To write your own API integration with Entra ID, see the `storagegrid-ssoauth-azure.py` script. The Python script makes two requests to StorageGRID directly (first to get the SAMLRequest, and later to get the authorization token), and also calls the Node.js script to interact with Entra ID to perform the SSO operations.

SSO operations can be executed using a series of API requests, but doing so is not straightforward. The Puppeteer Node.js module is used to scrape the Entra ID SSO interface.

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

Steps

1. Install the required dependencies, as follows:
   a. Install Node.js (see https://nodejs.org/en/download/).
   b. Install the required Node.js modules (puppeteer and jsdom):

      ```
      npm install -g <module>
      ```

2. Pass the Python script to the Python interpreter to run the script.

   The Python script will then call the corresponding Node.js script to perform the Entra ID SSO interactions.

3. When prompted, enter values for the following arguments (or pass them in using parameters):
   - The SSO email address used to sign in to Entra ID
   - The address for StorageGRID
   - The tenant account ID, if you want to access the Tenant Management API
4. When prompted, enter the password and be prepared to provide an MFA authorization to Entra ID if

requested.

```
c:\Users\user\Documents\azure_sso>py storagegrid-azure-ssoauth.py --sso-email-address user@my-domain.com
--sg-address storagegrid.examp.e.com --tenant-account-id 0
Enter the user's SSO password:
********************************************************************************

Watch for and approve a 2FA authorization request
********************************************************************************
StorageGRID Auth Token: {'responseTime': '2021-10-04T21:30:48.807Z', 'status': 'success', 'apiVersion':
'3.4', 'data': '4807d93e-a3df-48f2-9680-906cd255979e'}
```

> The script assumes MFA is done using Microsoft Authenticator. You might need to modify the script to support other forms of MFA (such as entering a code received in a text message).

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.

Use the API if single sign-on is enabled (PingFederate)

If you have configured and enabled single sign-on (SSO) and you use PingFederate as the SSO provider, you must issue a series of API requests to obtain an authentication token that is valid for the Grid Management API or the Tenant Management API.

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

These instructions apply if you are using PingFederate as the SSO identity provider

Before you begin

- 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 RHEL, `./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 method. You can enter any variation of "pingfederate" (PINGFEDERATE, pingfederate, and so on).
   - The SSO username
   - The domain where StorageGRID is installed. This field is not used for PingFederate. You can leave it blank or enter any value.
   - The address for StorageGRID
   - The tenant account ID, if you want to access the Tenant Management API.

   ```
   python3 storagegrid-ssoauth.py
   sso_method: pingfederate
   saml_user: my-sso-username
   saml_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 TENANTACCOUNTID='12345'
   export STORAGEGRID_ADDRESS='storagegrid.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://my-pf-baseurl/idp/SSO.saml2?...",
    "responseTime": "2018-11-06T16:30:23.355Z",
    "status": "success"
}
```

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

```
export SAMLREQUEST="https://my-pf-baseurl/idp/SSO.saml2?..."
```

d. Export the response and cookie, and echo the response:

```
RESPONSE=$(curl -c - "$SAMLREQUEST")
```

```
echo "$RESPONSE" | grep 'input type="hidden" name="pf.adapterId"
id="pf.adapterId"'
```

e. Export the 'pf.adapterId' value, and echo the response:

```
export ADAPTER='myAdapter'
```

```
echo "$RESPONSE" | grep 'base'
```

f. Export the 'href' value (remove the trailing slash /), and echo the response:

```
export BASEURL='https://my-pf-baseurl'
```

```
echo "$RESPONSE" | grep 'form method="POST"'
```

g. Export the 'action' value:

```
export SSOPING='/idp/.../resumeSAML20/idp/SSO.ping'
```

h. Send cookies along with credentials:

```
curl -b <(echo "$RESPONSE") -X POST "$BASEURL$SSOPING" \
--data "pf.username=$SAMLUSER&pf.pass=
$SAMLPASSWORD&pf.ok=clicked&pf.cancel=&pf.adapterId=$ADAPTER"
--include
```

i. Save the `SAMLResponse` from the hidden field:

```
export SAMLResponse='PHNhbWxwOlJlc3BvbnN...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.

Sign 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. These instructions apply if you are using PingFederate as the SSO identity provider

About this task

If required, you can sign out of the StorageGRID API 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://my-ping-
url/idp/SLO.saml2?SAMLRequest=fZDNboMwEIRfhZ...HcQ%3D%3D",
    "responseTime": "2021-10-12T22:20:30.839Z",
    "status": "success"
}
```

2. Save the logout URL.

```
export LOGOUT_REQUEST='https://my-ping-
url/idp/SLO.saml2?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: PF=QoKs...SgCC; Path=/; Secure; HttpOnly; SameSite=None
```

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
```

Deactivate features with the API

You can use the Grid Management API to completely deactivate certain features in the StorageGRID system. When a feature is deactivated, no one can be assigned permissions to perform the tasks related to that feature.

About this task

The Deactivated Features system allows you to prevent access to certain features in the StorageGRID system. Deactivating a feature is the only way to prevent the root user or users who belong to admin groups with **Root access** permission from being able to use that feature.

To understand how this functionality might be useful, consider the following scenario:

Company A is a service provider who leases the storage capacity of their StorageGRID system by creating tenant accounts. To protect the security of their leaseholders' objects, Company A wants to ensure that its own employees can never access any tenant account after the account has been deployed.

*Company A can accomplish this goal by using the Deactivate Features system in the Grid Management API. By completely deactivating the **Change tenant root password** feature in the Grid Manager (both the UI and the API), Company A ensures that Admin users—including the root user and users belonging to groups with the **Root access** permission—can't change the password for any tenant account's root user.*

Steps

1. Access the Swagger documentation for the Grid Management API. See Use the Grid Management API.

2. Locate the Deactivate Features endpoint.
3. To deactivate a feature, such as Change tenant root password, send a body to the API like this:

```
{ "grid": {"changeTenantRootPassword": true} }
```

When the request is complete, the Change tenant root password feature is disabled. The **Change tenant root password** management permission no longer appears in the user interface, and any API request that attempts to change the root password for a tenant will fail with "403 Forbidden."

Reactivate deactivated features

By default, you can use the Grid Management API to reactivate a feature that has been deactivated. However, if you want to prevent deactivated features from ever being reactivated, you can deactivate the **activateFeatures** feature itself.

The **activateFeatures** feature can't be reactivated. If you decide to deactivate this feature, be aware that you will permanently lose the ability to reactivate any other deactivated features. You must contact technical support to restore any lost functionality.

Steps

1. Access the Swagger documentation for the Grid Management API.
2. Locate the Deactivate Features endpoint.
3. To reactivate all features, send a body to the API like this:

```
{ "grid": null }
```

When this request is complete, all features, including the Change tenant root password feature, are reactivated. The **Change tenant root password** management permission now appears in the user interface, and any API request that attempts to change the root password for a tenant will succeed, assuming the user has the **Root access** or **Change tenant root password** management permission.

The previous example causes *all* deactivated features to be reactivated. If other features have been deactivated that should remain deactivated, you must explicitly specify them in the PUT request. For example, to reactivate the Change tenant root password feature and continue to deactivate the storageAdmin management permission, send this PUT request:

```
{ "grid": {"storageAdmin": true} }
```

Copyright information

Trademark information