使用 API

StorageGRID software

NetApp
July 23, 2026

目录

使用 API

使用 StorageGRID 网格管理 API

您可以使用网格管理 REST API 而不是 Grid Manager 用户界面来执行系统管理任务。例如，您可能希望使用 API 来自动化操作或更快地创建多个实体（例如用户）。

顶级资源

网格管理API可提供以下顶级资源：

- `/grid`：访问仅限于 Grid Manager 用户，并基于配置的组权限。
- `/org`：访问仅限于属于租户帐户的本地或联合 LDAP 组的用户。有关详细信息，请参见 "使用租户帐户"。
- `/private`：访问仅限于 Grid Manager 用户，并基于配置的组权限。私有 API 如有更改，恕不另行通知。StorageGRID 私有端点也会忽略请求的 API 版本。

发出 API 请求

网格管理 API 使用 Swagger 开源 API 平台。Swagger 提供了一个直观的用户界面，允许开发人员和非开发人员使用 API 在 StorageGRID 中执行实时操作。

Swagger 用户界面为每个 API 操作提供完整的详细信息和文档。

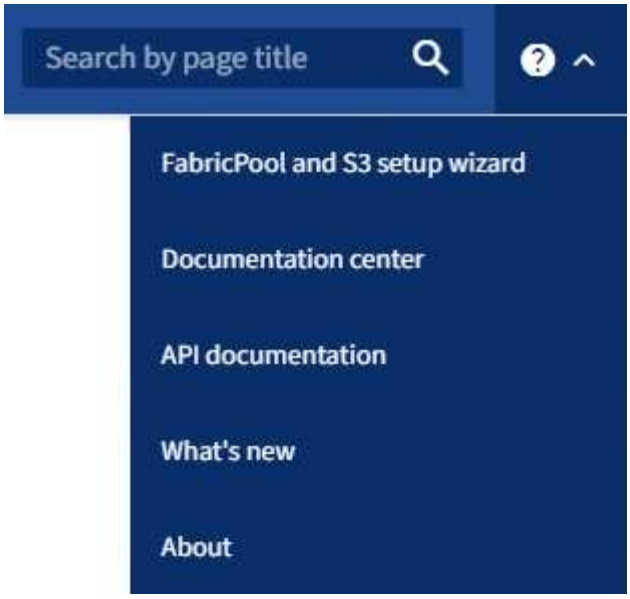
开始之前

- 您已使用 "支持的 Web 浏览器" 登录到网格管理器。
- 您有 "特定访问权限"。

> 使用 API 文档网页执行的任何 API 操作都是实时操作。注意不要错误地创建、更新或删除配置数据或其他数据。

步骤

1. 从 Grid Manager 标题中，选择帮助图标，然后选择 **API documentation**。



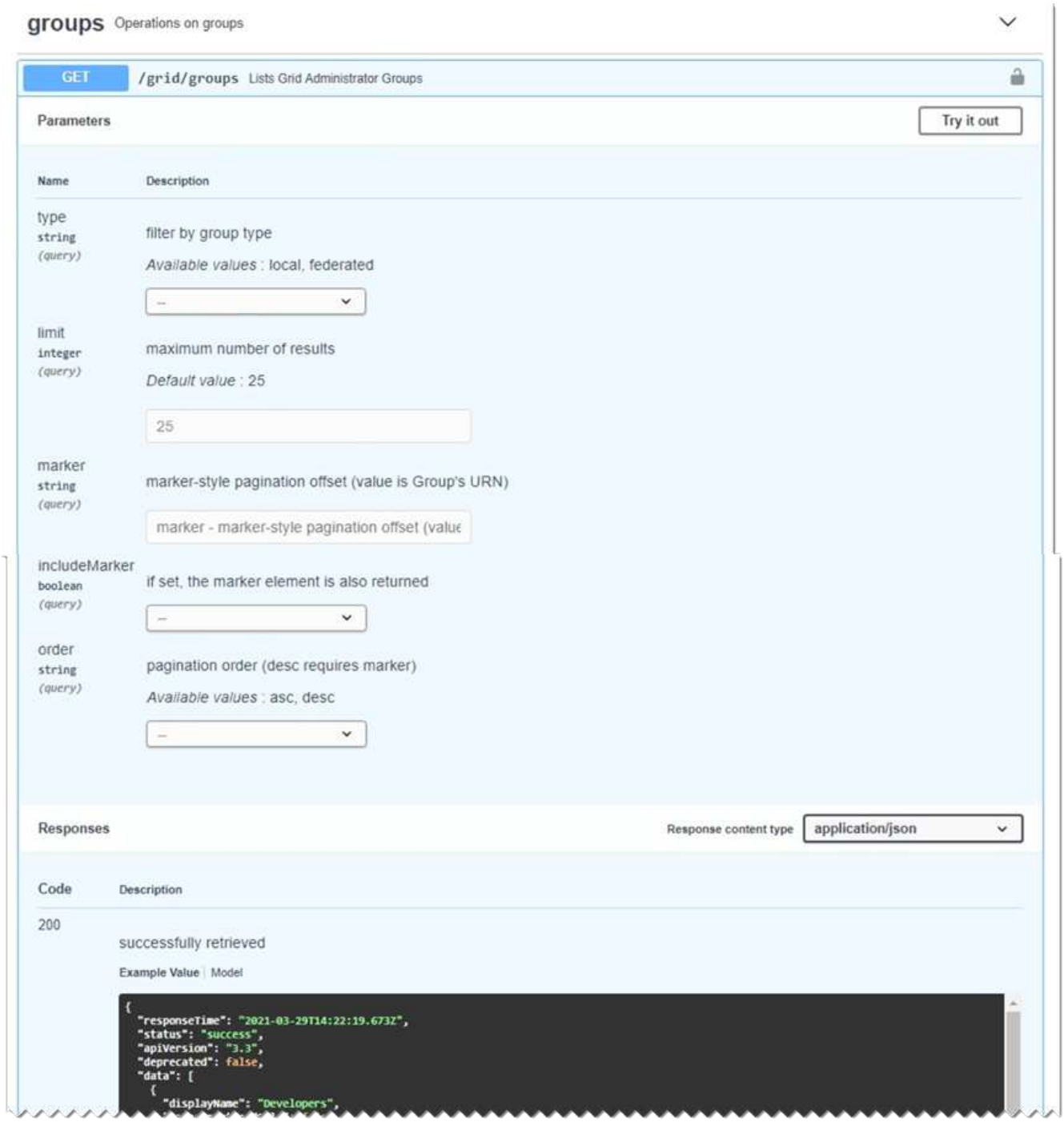
2. 要使用私有 API 执行操作，请在 StorageGRID Management API 页面上选择 转到私有 **API** 文档。

   私有 API 如有更改，恕不另行通知。StorageGRID 专用端点也会忽略请求的 API 版本。

3. 选择所需的操作。

   展开 API 操作时，可以查看可用的 HTTP 操作，例如 GET、PUT、UPDATE 和 DELETE。

4. 选择 HTTP 操作以查看请求详细信息，包括端点 URL、任何必需或可选参数的列表、请求正文示例（如果需要）以及可能的响应。



5. 确定请求是否需要其他参数，例如组或用户 ID。然后，获取这些值。您可能需要先发出其他 API 请求，才能获取所需信息。
6. 确定是否需要修改示例请求正文。如果是，您可以选择 **Model** 以了解每个字段的要求。
7. 选择*试用*。
8. 请提供任何所需的参数，或根据需要修改请求正文。
9. ${post_edited_translations.segment}
10. 查看响应代码以确定请求是否成功。

StorageGRID 中的网格管理 API 操作

网格管理API将可用操作组织到以下部分。

此列表仅包括公共 API 中可用的操作。

- **accounts**：管理存储租户帐户的操作，包括创建新帐户和检索给定帐户的存储使用情况。
- **alert-history**：对已解决的警报进行操作。
- **alert-receivers**：对警报通知接收器（电子邮件）的操作。
- **alert-rules**：对警报规则的操作。
- **alert-silences**：对警报静默的操作。
- **alerts**：对警报的操作。
- **audit**：列出和更新审核配置的操作。
- **auth**：执行用户会话身份验证的操作。

  网格管理API支持承载令牌身份验证方案。若要登录，请在身份验证请求的JSON正文中提供用户名和密码（即 `POST /api/v3/authorize`）。如果用户成功通过身份验证，则返回安全令牌。此令牌必须在后续API请求的标头中提供（"Authorization: Bearer *token*"）。令牌将在16小时后过期。

  如果为 StorageGRID 系统启用了单点登录，则必须执行不同的步骤来进行身份验证。请参阅"如果启用了单点登录，则在 API 中进行身份验证。"

  有关提高身份验证安全性的信息，请参阅"防止跨站点请求伪造"。

- **client-certificates**：配置客户端证书的操作，以便使用外部监控工具安全地访问 StorageGRID。
- **config**：与产品发布和 Grid Management API 版本相关的操作。您可以列出产品发布版本以及该版本支持的 Grid Management API 的主要版本，也可以禁用 API 的已弃用版本。
- **deactivated-features**：查看可能已停用的功能的操作。
- **dns-servers**：列出和更改已配置的外部 DNS 服务器的操作。
- **drive-details**：针对特定存储设备型号的驱动器上的操作。
- **endpoint-domain-names**：用于列出和更改 S3 端点域名的操作。
- **erasure-coding**：对纠删码配置文件的操作。
- **expansion**：扩展操作（过程级）。
- **expansion-nodes**：扩展上的操作（节点级）。
- **expansion-sites**：扩展操作（站点级）。
- **grid-networks**：列出和更改网格网络列表的操作。
- **grid-passwords**：网格密码管理的操作。
- **groups**：用于管理本地网格管理员组以及从外部 LDAP 服务器检索联合网格管理员组的操作。
- **identity-source**：配置外部身份源并手动同步联合组和用户信息的操作。

- **ilm**：对信息生命周期管理 (ILM) 的操作。
- **in-progress-procedures**：检索当前正在进行的维护程序。
- **license**：检索和更新 StorageGRID 许可证的操作。
- **logs**：收集和下载日志文件的操作。v
- 指标：对 StorageGRID 指标的操作，包括在单个时间点的即时指标查询和在一定时间范围内的范围指标查询。网格管理 API 使用 Prometheus 系统监控工具作为后端数据源。有关构造 Prometheus 查询的信息，请参阅 Prometheus 网站。

> 名称中包含 *private* 的指标仅供内部使用。这些指标可能在不同 StorageGRID 版本之间发生变化，恕不另行通知。

- **node-details**：节点详细信息上的操作。
- **node-health**：对节点健康状态的操作。
- **node-storage-state**：对节点存储状态的操作。
- **ntp-servers**：用于列出或更新外部网络时间协议 (NTP) 服务器的操作。
- **objects**：对象和对象元数据的操作。
- **recovery**：恢复过程的操作。
- **recovery-package**：下载恢复包的操作。
- **regions**：用于查看和创建区域的操作。
- **s3-object-lock**：对全局 S3 Object Lock 设置执行的操作。
- **server-certificate**：用于查看和更新 Grid Manager 服务器证书的操作。
- **snmp**：当前 SNMP 配置上的操作。
- **storage-watermarks**：存储节点水印。
- **traffic-classes**：流量分类策略的操作。
- **untrusted-client-network**：对不受信任的客户端网络配置的操作。
- 用户：用于查看和管理 Grid Manager 用户的操作。

StorageGRID 中的网格管理 API 版本控制

网格管理 API 使用版本控制来支持无中断升级。

例如，此请求 URL 指定 API 的版本 4。

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

当进行与旧版本_不兼容_的更改时，API 的主版本号将递增。当进行与旧版本_兼容_的更改时，API 的次版本号将递增。兼容的更改包括添加新端点或新属性。

以下示例说明了如何根据所做的更改类型提升 API 版本。

API 变更类型	旧版本	新版本
与旧版本兼容	2.1	2.2
与旧版本不兼容	2.1	3.0

首次安装 StorageGRID 软件时，仅启用最新版本的 API。但是，当您升级到 StorageGRID 的新功能版本时，您仍然可以访问至少一个 StorageGRID 功能版本的旧 API 版本。

> 您可以配置支持的版本。有关详细信息，请参见 Swagger API 文档的 "网格管理 API" 的 **config** 部分。将所有 API 客户端更新为使用较新版本后，应停用对较旧版本的支持。

过期的请求通过以下方式标记为已弃用：

- 响应标头为"Deprecated: true"
- JSON 响应正文包括 "deprecated": true
- 已将弃用警告添加到 nms.log。例如：

```
Received call to deprecated v2 API at POST "/api/v2/authorize"
```

确定当前版本支持的API版本

使用 `GET /versions` API 请求可返回支持的 API 主要版本的列表。此请求位于 Swagger API 文档的 **config** 部分。

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

为请求指定 API 版本

您可以使用路径参数(`/api/v4`）或标头(`Api-Version: 4`）指定 API 版本。如果同时提供这两个值，则标头值会覆盖路径值。

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

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

防止 StorageGRID 中的跨站请求伪造 (CSRF)

您可以通过使用 CSRF 令牌来增强使用 Cookie 的身份验证，从而帮助防止针对 StorageGRID 的跨站点请求伪造 (CSRF) 攻击。Grid Manager 和 Tenant Manager 会自动启用此安全功能；其他 API 客户端可以选择是否在登录时启用此功能。

可以触发对其他站点（例如使用 HTTP 表单 POST）的请求的攻击者，可以使用已登录用户的 Cookie 发出某些请求。

StorageGRID 通过使用 CSRF 令牌来帮助抵御 CSRF 攻击。启用后，特定 Cookie 的内容必须与特定标头或特定 POST 正文参数的内容匹配。

要启用此功能，请在身份验证期间将 `csrfToken`参数设置为 `true`。默认值为 `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"
```

如果为 true，则使用随机值设置 `GridCsrfToken` cookie 以登录到 Grid Manager，并使用随机值设置 `AccountCsrfToken` cookie 以登录到 Tenant Manager。

如果存在 Cookie，则可以修改系统状态（POST、PUT、PATCH、DELETE）的所有请求都必须包括以下内容之一：

- The `X-Csrf-Token` 标头，标头的值设置为 CSRF 令牌 cookie 的值。
- 对于接受表单编码正文的端点：一个 `csrfToken`表单编码的请求正文参数。

有关其他示例和详细信息，请参见在线 API 文档。

> 设置了 CSRF 令牌 cookie 的请求还将对任何期望 JSON 请求正文的请求强制执行"Content-Type: application/json"标头，作为针对 CSRF 攻击的额外保护。

如果启用了单一登录，请使用 API

如果已启用单一登录，请使用 StorageGRID API（Active Directory）

如果已启用 "已配置并启用单点登录(SSO)" 并将 Active Directory 用作 SSO 提供程序，则必须发出一系列 API 请求，以获取对网格管理 API 或租户管理 API 有效的身份验证令牌。

如果启用了单点登录，请登录到 API

如果您使用 Active Directory 作为 SSO 身份提供程序，则适用以下说明。

开始之前

- 您知道属于 StorageGRID 用户组的联合用户的 SSO 用户名和密码。
- 如果要访问租户管理 API，则需要知道租户帐户 ID。

关于此任务

可以使用以下示例之一获取身份验证令牌：

- 该 `storagegrid-ssoauth.py` Python 脚本位于 StorageGRID 安装文件目录中（RHEL 为(`./rpms`，`./debs` 适用于 Ubuntu 或 Debian，VMware 为 `./vsphere`）。
- curl 请求的示例工作流。

  如果执行太慢，curl 工作流可能会超时。您可能会看到以下错误： `A valid SubjectConfirmation was not found on this Response`。

  > 示例 curl 工作流程不会保护密码免遭其他用户查看。

如果您遇到 URL 编码问题，则可能会看到错误： `Unsupported SAML version`。

步骤

1. 选择以下方法之一以获取身份验证令牌：
   - 使用 `storagegrid-ssoauth.py` Python 脚本。转至第 2 步。
   - 使用 curl 请求。转到步骤 3。
2. 如果要使用 `storagegrid-ssoauth.py`脚本，请将脚本传递给 Python 解释器并运行脚本。

   出现提示时，请输入以下参数的值：
   - SSO 方法。输入 ADFS 或 adfs。
   - SSO 用户名
   - 安装 StorageGRID 的域
   - StorageGRID 的地址
   - 租户帐户 ID（如果要访问租户管理 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
```

StorageGRID 授权令牌在输出中提供。您现在可以将令牌用于其他请求，类似于在未使用 SSO 的情况下使用 API 的方式。

3. 如果要使用 curl 请求，请使用以下过程。

   a. 声明登录所需的变量。

```
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'
```

   要访问 Grid Management API，请使用 0 作为 `TENANTACCOUNTID`。

   b. 若要接收已签名的身份验证 URL，请向 `/api/v3/authorize-saml`发出 POST 请求，并从响应中删除其他 JSON 编码。

   此示例显示了针对 `TENANTACCOUNTID`的已签名身份验证 URL 的 POST 请求。结果将传递给 `python -m json.tool`以删除 JSON 编码。

```
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
```

   此示例的响应包括经过 URL 编码的签名 URL，但不包括附加的 JSON 编码层。

```
{
    "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. 从响应中保存 `SAMLRequest`，以便在后续命令中使用。

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

d. 从 AD FS 获取包含客户端请求 ID 的完整 URL。

一种选择是使用上一个响应中的 URL 请求登录表单。

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

响应包括客户端请求 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. 从响应中保存客户端请求 ID。

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

f. 将您的凭据发送到上一个响应的表单操作。

```
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返回302重定向，标头中包含其他信息。

> 如果您的 SSO 系统启用了多因素身份验证 (MFA)，表单提交内容中也会包含第二个密码或其他凭据。

```
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. 从响应中保存 `MSISAuth` Cookie。

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

h. 使用来自身份验证 POST 的 Cookie 将 GET 请求发送到指定位置。

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

响应标头将包含 AD FS 会话信息，以供以后注销使用，响应正文在隐藏表单字段中包含 SAMLResponse。

```
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. 从隐藏字段中保存 `SAMLResponse`：

```
export SAMLResponse='PHNhbWxwOlJlc3BvbnN...1scDpSZXNwb25zZT4='
```

j. 使用保存的 `SAMLResponse`，发出 StorageGRID `/api/saml-response` 请求以生成 StorageGRID 身份验证令牌。

对于 `RelayState`，请使用租户帐户 ID，或者如果要登录到 Grid Management API，请使用 0。

```
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
```

响应包括身份验证令牌。

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

a. 将响应中的身份验证令牌另存为 `MYTOKEN`。

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

现在，您可以将 `MYTOKEN`用于其他请求，类似于在未使用 SSO 的情况下使用 API 的方式。

如果已启用单一登录，则注销 API

如果已启用单点登录 (SSO)，则必须发出一系列 API 请求以注销网格管理 API 或租户管理 API。如果您将 Active Directory 用作 SSO 身份提供程序，则适用以下说明

关于此任务

如果需要，您可以通过从组织的单点注销页面注销来退出 StorageGRID API。或者，您可以从 StorageGRID 触发单点注销 (SLO)，这需要有效的 StorageGRID 承载令牌。

步骤

1. 要生成已签名的注销请求，请将 `cookie "sso=true"` 传递给 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
```

返回注销 URL：

```
{
    "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. 保存注销 URL。

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

3. 向注销 URL 发送请求以触发 SLO 并重定向回 StorageGRID。

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

返回 302 响应。重定向位置不适用于仅限 API 的注销。

```
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. 删除 StorageGRID 承载令牌。

删除 StorageGRID 承载令牌的工作方式与没有 SSO 的工作方式相同。如果未提供 `cookie "sso=true"，则用户将从 StorageGRID 注销，而不会影响 SSO 状态。

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

A `204 No Content` 响应表示用户现已注销。

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

如果已启用单点登录，请使用 StorageGRID API（Entra ID）

如果有"已配置并启用单点登录(SSO)"并将 Entra ID 用作 SSO 提供程序，则可以使用两个示例脚本获取对网格管理 API 或租户管理 API 有效的身份验证令牌。

如果启用了 Entra ID 单点登录，请登录到 API

如果您使用 Entra ID 作为 SSO 身份提供商，则适用以下说明

开始之前

- 您知道属于 StorageGRID 用户组的联合用户的 SSO 电子邮件地址和密码。
- 如果要访问租户管理 API，则需要知道租户帐户 ID。

关于此任务

可以使用以下示例脚本获取身份验证令牌：

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

这两个脚本都位于 StorageGRID 安装文件目录中（RHEL 为 (`./rpms`，Ubuntu 或 Debian 为 `./debs`，VMware 为 `./vsphere`）。

要使用 Entra ID 编写自己的 API 集成，请参见 `storagegrid-ssoauth-azure.py` 脚本。Python 脚本直接向 StorageGRID 发出两个请求（首先获取 SAMLRequest，然后获取授权令牌），并调用 Node.js 脚本与 Entra ID 交互以执行 SSO 操作。

SSO 操作可以使用一系列 API 请求执行，但这样做并不简单。Puppeteer Node.js 模块用于抓取 Entra ID SSO 接口。

如果您遇到 URL 编码问题，则可能会看到错误： `Unsupported SAML version`。

步骤

1. 安装所需的依赖项，如下：
   a. 安装 Node.js（请参见 "https://nodejs.org/en/download/"）。
   b. 安装所需的 Node.js 模块（puppeteer 和 jsdom）：

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

2. 将 Python 脚本传递给 Python 解释器以运行脚本。

   然后，Python 脚本将调用相应的 Node.js 脚本来执行 Entra ID SSO 交互。

3. 出现提示时，输入以下参数的值（或使用参数传入）：
   - 用于登录到 Entra ID 的 SSO 电子邮件地址
   - StorageGRID 的地址
   - 租户帐户 ID（如果要访问租户管理 API）
4. 出现提示时，请输入密码，并准备好在收到请求时向 Entra ID 提供 MFA 授权。

```
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'}
```

该脚本假设 MFA 是使用 Microsoft Authenticator 完成的。您可能需要修改脚本以支持其他形式的 MFA（例如输入在短信中收到的代码）。

StorageGRID 授权令牌在输出中提供。您现在可以将令牌用于其他请求，类似于在未使用 SSO 的情况下使用 API 的方式。

如果已启用单一登录，请使用 **StorageGRID API (PingFederate)**

如果具有"已配置并启用单点登录(SSO)"并使用 PingFederate 作为 SSO 提供程序，则必须发出一系列 API 请求，以获取对网格管理 API 或租户管理 API 有效的身份验证令牌。

如果启用了单点登录，请登录到 **API**

如果您使用 PingFederate 作为 SSO 身份提供商，则适用以下说明

开始之前

- 您知道属于 StorageGRID 用户组的联合用户的 SSO 用户名和密码。
- 如果要访问租户管理 API，则需要知道租户帐户 ID。

关于此任务

可以使用以下示例之一获取身份验证令牌：

- 该 `storagegrid-ssoauth.py` Python 脚本位于 StorageGRID 安装文件目录中（RHEL 为(`./rpms`，`./debs` 适用于 Ubuntu 或 Debian，VMware 为 `./vsphere`）。
- curl 请求的示例工作流。

  如果执行太慢，curl 工作流可能会超时。您可能会看到以下错误： `A valid SubjectConfirmation was not found on this Response`。

  示例 curl 工作流程不会保护密码免遭其他用户查看。

  如果您遇到 URL 编码问题，则可能会看到错误： `Unsupported SAML version`。

步骤

1. 选择以下方法之一以获取身份验证令牌：
   - 使用 `storagegrid-ssoauth.py` Python 脚本。转至第 2 步。
   - 使用 curl 请求。转到步骤 3。
2. 如果要使用 `storagegrid-ssoauth.py`脚本，请将脚本传递给 Python 解释器并运行脚本。

   出现提示时，请输入以下参数的值：

   - SSO 方法。您可以输入"pingfederate"的任何变体（PINGFEDERATE、pingfederate 等）。
   - SSO 用户名
   - 安装 StorageGRID 的域。此字段不用于 PingFederate。您可以将其留空或输入任何值。

- StorageGRID 的地址
- 租户帐户 ID（如果要访问租户管理 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
```

StorageGRID 授权令牌在输出中提供。您现在可以将令牌用于其他请求，类似于在未使用 SSO 的情况下使用 API 的方式。

3. 如果要使用 curl 请求，请使用以下过程。

   a. 声明登录所需的变量。

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

   要访问 Grid Management API，请使用 0 作为 `TENANTACCOUNTID`。

   b. 若要接收已签名的身份验证 URL，请向 `/api/v3/authorize-saml`发出 POST 请求，并从响应中删除其他 JSON 编码。

   此示例显示了对 TENANTACCOUNTID 的签名身份验证 URL 的 POST 请求。结果将传递到 python -m json.tool 以删除 JSON 编码。

```
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
```

   此示例的响应包括经过 URL 编码的签名 URL，但不包括附加的 JSON 编码层。

```
{
    "apiVersion": "3.0",
    "data": "https://my-pf-baseurl/idp/SSO.saml2?...",
    "responseTime": "2018-11-06T16:30:23.355Z",
    "status": "success"
}
```

c. 从响应中保存 `SAMLRequest`，以便在后续命令中使用。

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

d. 导出响应和 cookie，并回显响应：

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

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

e. 导出"pf.adapterId"值，并回显响应：

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

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

f. 导出"href"值（删除尾部斜杠 /），并回显响应：

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

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

g. 导出"action"值：

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

h. 将Cookie与凭据一起发送：

```
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. 从隐藏字段中保存 `SAMLResponse`：

```
export SAMLResponse='PHNhbWxwOlJlc3BvbnN...1scDpSZXNwb25zZT4='
```

j. 使用保存的 `SAMLResponse`，发出 StorageGRID `/api/saml-response` 请求以生成 StorageGRID 身份验证令牌。

对于 `RelayState`，请使用租户帐户 ID，或者如果要登录到 Grid Management API，请使用 0。

```
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
```

响应包括身份验证令牌。

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

a. 将响应中的身份验证令牌另存为 `MYTOKEN`。

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

现在，您可以将 `MYTOKEN`用于其他请求，类似于在未使用 SSO 的情况下使用 API 的方式。

如果已启用单一登录，则注销 API

如果已启用单点登录 (SSO)，则必须发出一系列 API 请求以注销网格管理 API 或租户管理 API。如果您使用 PingFederate 作为 SSO 身份提供商，则适用以下说明

关于此任务

如果需要，您可以通过从组织的单点注销页面注销来退出 StorageGRID API。或者，您可以从 StorageGRID 触发单点注销 (SLO)，这需要有效的 StorageGRID 承载令牌。

步骤

1. 要生成已签名的注销请求，请将 `cookie "sso=true"` 传递给 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
```

返回注销 URL：

```
{
    "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. 保存注销 URL。

```
export LOGOUT_REQUEST='https://my-ping-
url/idp/SLO.saml2?SAMLRequest=fZDNboMwEIRfhZ...HcQ%3D%3D'
```

3. 向注销 URL 发送请求以触发 SLO 并重定向回 StorageGRID。

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

返回 302 响应。重定向位置不适用于仅限 API 的注销。

```
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. 删除 StorageGRID 承载令牌。

删除 StorageGRID 承载令牌的工作方式与没有 SSO 的工作方式相同。如果未提供 `cookie "sso=true"，则用户将从 StorageGRID 注销，而不会影响 SSO 状态。

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

A `204 No Content` 响应表示用户现已注销。

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

使用 API 停用 StorageGRID 功能

您可以使用网格管理 API 完全停用 StorageGRID 系统中的某些功能。停用某项功能后，无法向任何人分配执行与该功能相关的任务的权限。

关于此任务

通过停用功能系统，您可以阻止访问 StorageGRID 系统中的某些功能。停用功能是阻止 root 用户或属于具有 **Root access** 权限的管理员组的用户使用该功能的唯一方法。

要了解此功能的实用性，请考虑以下情形：

A 公司是一家服务提供商，通过创建租户帐户来租赁其 *StorageGRID* 系统的存储容量。为了保护租户对象的安全，*A* 公司希望确保其员工在部署帐户后永远无法访问任何租户帐户。

A 公司可以通过使用网格管理 *API* 中的停用功能系统来实现此目标。通过完全停用网格管理器（*UI* 和 *API*）中的*更改租户 *root* 密码*功能，*A* 公司确保管理员用户（包括 *root* 用户和属于具有*根访问权限*的组的用户）无法更改任何租户帐户的 *root* 用户密码。

步骤

1. 访问网格管理 API 的 Swagger 文档。请参阅 "使用 Grid Management API"。
2. 找到"停用功能"端点。
3. 要停用某项功能（例如 Change tenant root password），请按如下所示向 API 发送正文：

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

   请求完成后，将禁用更改租户根密码功能。用户界面中不再显示 **Change tenant root password** 管理权限，任何尝试更改租户 root 用户密码的 API 请求都将失败并显示"403 Forbidden"。

重新激活已停用的功能

默认情况下，您可以使用 Grid Management API 重新激活已停用的功能。但是，如果您想防止已停用的功能被重新激活，您可以停用 **activateFeatures** 功能本身。

activateFeatures 功能无法重新激活。如果您决定停用此功能，请注意，您将永久失去重新激活任何其他已停用功能的能力。您必须联系技术支持以恢复任何丢失的功能。

步骤

1. 访问网格管理 API 的 Swagger 文档。
2. 找到"停用功能"端点。
3. 要重新激活所有功能，请将正文发送到 API，如下所示：

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

此请求完成后，将重新激活所有功能，包括更改租户 root 密码功能。*更改租户 root 密码*管理权限现在显示在用户界面中，任何尝试更改租户 root 密码的 API 请求都将成功，前提是用户具有 *Root 访问权限*或*更改租户 root 密码*管理权限。

前面的示例会导致_all_已停用的功能重新激活。如果其他应保持停用状态的功能已被停用，则必须在 PUT 请求中明确指定它们。例如，要重新激活"更改租户 root 用户密码"功能并继续停用 storageAdmin 管理权限，请发送此 PUT 请求：

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