使用 API

StorageGRID 11.7

NetApp
April 12, 2024

目录

使用 API

使用网格管理 API

您可以使用网格管理 REST API 执行系统管理任务，而不是使用网格管理器用户界面。例如，您可能希望使用 API 来自动执行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`：访问权限仅限于Grid Manager用户、并且取决于配置的组权限。
- `/org`：只有属于租户帐户的本地或联合LDAP组的用户才能访问。有关详细信息，请参见 "使用租户帐户"。
- `/private`：访问权限仅限于Grid Manager用户、并且取决于配置的组权限。专用 API 如有更改，恕不另行通知。StorageGRID 私有端点也会忽略此请求的 API 版本。

问题描述 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，使开发人员和非开发人员能够使用 API 在 StorageGRID 中执行实时操作。

Swagger 用户界面提供了每个 API 操作的完整详细信息和文档。

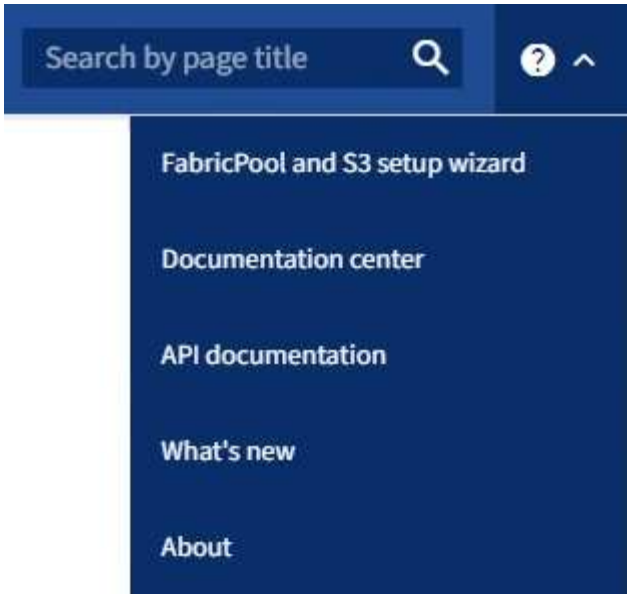
开始之前

- 您将使用登录到网格管理器 "支持的 Web 浏览器"。
- 您具有特定的访问权限。

> 使用 API 文档网页执行的任何 API 操作均为实时操作。请注意，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 documents*。



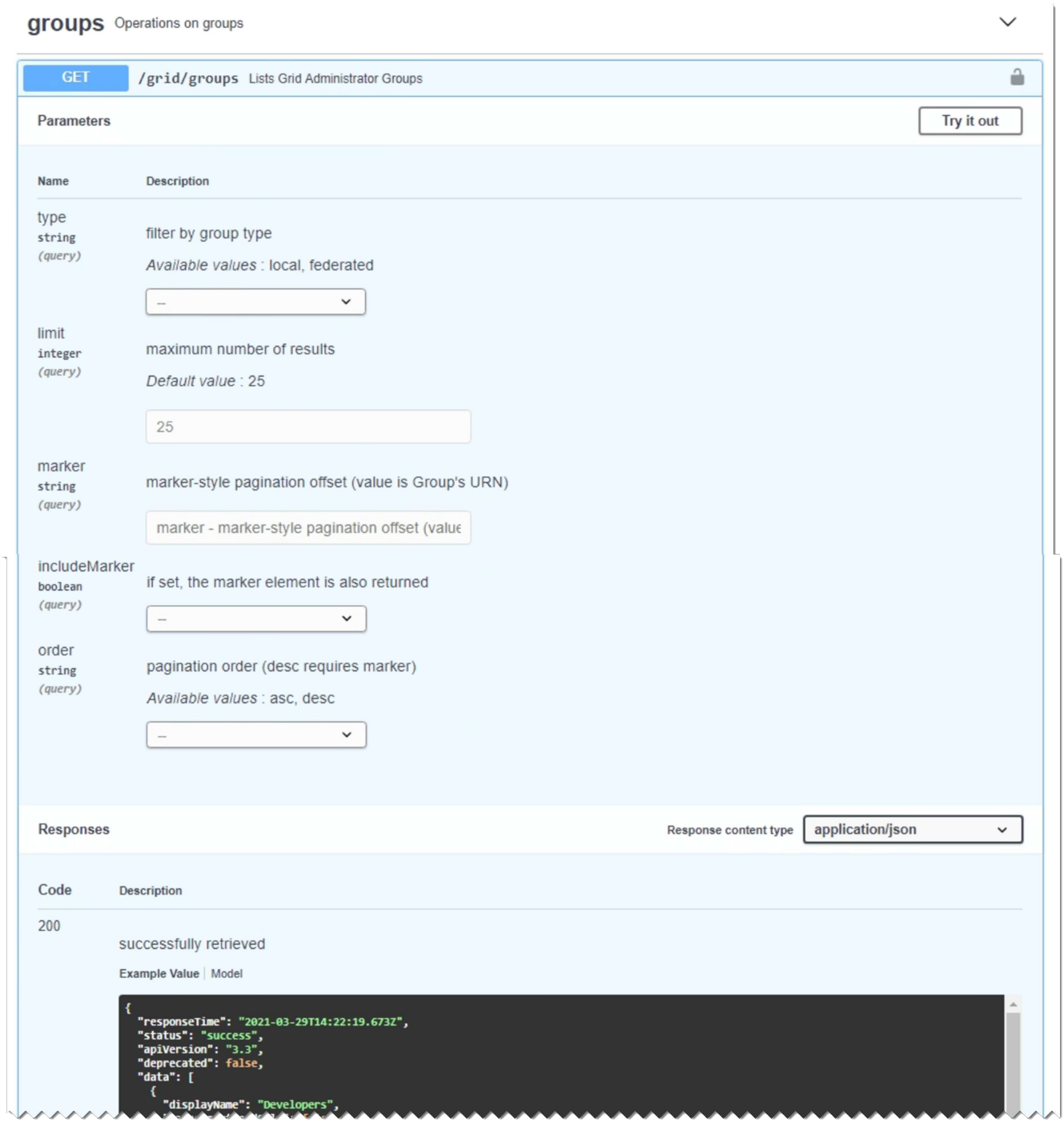
2. 要使用专用 API 执行操作，请在 StorageGRID 管理 API 页面上选择 * 转至专用 API 文档 * 。

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

3. 选择所需的操作。

   展开 API 操作时，您可以看到可用的 HTTP 操作，例如 GET ， PUT ， UPDATE 和 DELETE 。

4. 选择 HTTP 操作可查看请求详细信息，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. 确定此请求是否需要其他参数，例如组或用户 ID 。然后，获取这些值。您可能需要先对其他 API 请求进行问题描述 处理，以获取所需的信息。
6. 确定是否需要修改示例请求正文。如果是，您可以选择 * 型号 * 来了解每个字段的要求。
7. 选择 * 试用 * 。
8. 提供所需的任何参数，或根据需要修改请求正文。
9. 选择 * 执行 * 。
10. 查看响应代码以确定请求是否成功。

网格管理 API 操作

网格管理 API 将可用操作组织到以下部分中。

> 此列表仅包含公有 API 中可用的操作。

- 帐户：用于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量。
- 警报：用于列出当前警报(传统系统)并返回有关网格运行状况的信息(包括当前警报和节点连接状态摘要)的操作。
- **alerts**历史记录：对已解决的警报执行操作。
- 警报接收者：警报通知接收者操作(电子邮件)。
- 警报规则：对警报规则执行操作。
- 警报静音：警报静音操作。
- 警报：对警报执行操作。
- **audi**：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 支持不可承载令牌身份验证方案。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`）。如果用户已成功通过身份验证，则会返回一个安全令牌。此令牌必须在后续 API 请求的标题中提供（ "Authorization： bearer *token*" ）。

  > 如果为 StorageGRID 系统启用了单点登录，则必须执行不同的步骤进行身份验证。请参见 "` 在启用单点登录后对 API 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**：用于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**：与网格管理API的产品发行版和版本相关的操作。您可以列出该版本支持的网格管理 API 的产品版本和主要版本，并且可以禁用已弃用的 API 版本。
- *DEactive-Features *：用于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。
- **endpoint-domain-names**：列出和更改S3端点域名的操作。
- 纠删编码：对纠删编码配置文件的操作。
- 扩展：扩展操作(程序级)。
- 扩展节点：扩展操作(节点级)。
- 扩展站点：扩展操作(站点级)。
- *GRE-NETWORKS：列出和更改Grid Network List的操作。
- **GRID**密码：网格密码管理操作。
- 组：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。
- 身份源：用于配置外部身份源以及手动同步联盟组和用户信息的操作。

- *ILM：有关信息生命周期管理(ILM)的操作。
- **license**：用于检索和更新StorageGRID 许可证的操作。
- **logs**：用于收集和下载日志文件的操作。
- **metrics**：对StorageGRID 指标的操作，包括在某一时间点的即时指标查询和在一段时间内的范围指标查询。网格管理 API 使用 Prometheus 系统监控工具作为后端数据源。有关构建 Prometheus 查询的信息，请参见 Prometheus 网站。

> 包括的指标 *private* 其名称仅供内部使用。这些指标可能会在 StorageGRID 版本之间发生更改，恕不另行通知。

- 节点详细信息：对节点详细信息执行的操作。
- 节点运行状况：对节点运行状况执行的操作。
- **NONE-storage-state**：对节点存储状态执行的操作。
- **ntp-server**：列出或更新外部网络时间协议(NTP)服务器的操作。
- 对象：对对象和对象元数据执行的操作。
- 恢复：恢复操作步骤 的操作。
- 恢复包：用于下载恢复软件包的操作。
- 区域：用于查看和创建区域的操作。
- **s3-object-lock**：对全局S3对象锁定设置执行操作。
- **server-certificates**：用于查看和更新Grid Manager服务器证书的操作。
- **SNMP**：对当前SNMP配置执行的操作。
- **Traffic Classes**：流量分类策略的操作。
- 不可信客户端网络：对不可信客户端网络配置执行的操作。
- 用户：用于查看和管理Grid Manager用户的操作。

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 版本 3 。

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

如果对旧版本进行了 * 不兼容 _* 的更改，则租户管理 API 的主要版本将发生递增。如果对 * 与旧版本兼容 _* 进行了更改，则租户管理 API 的次要版本将发生递增。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更改的类型对 API 版本进行递增。

API 的更改类型	旧版本	新版本
与旧版本兼容	2.1	2.2

API 的更改类型	旧版本	新版本
与旧版本不兼容	2.1	3.0

首次安装 StorageGRID 软件时，仅会启用最新版本的网格管理 API 。但是，在升级到 StorageGRID 的新功能版本时，您仍可以访问至少一个 StorageGRID 功能版本的旧版 API 。

> 您可以使用网格管理 API 配置受支持的版本。有关详细信息，请参见 Swagger API 文档中的 "`config` " 一节。在更新所有网格管理 API 客户端以使用较新版本后，您应停用对较旧版本的支持。

已过时的请求将通过以下方式标记为已弃用：

- 响应标头为 "depression: true"
- JSON 响应正文包含 "depressioned" ： true
- NMS.log 中会添加一个已弃用的警告。例如：

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

确定当前版本支持哪些 API 版本

请使用以下 API 请求返回受支持的 API 主要版本列表：

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

指定请求的 API 版本

您可以使用path参数指定API版本 (`/api/v3`)或标题 (`Api-Version: 3`）。如果同时提供这两个值，则标头值将覆盖路径值。

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

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

防止跨站点请求伪造（CSRF）

您可以通过使用 CSRF 令牌增强使用 Cookie 的身份验证，帮助防止 StorageGRID 受到跨站点请求伪造（CSRF）攻击。网格管理器和租户管理器会自动启用此安全功能；其他 API 客户端可以选择在登录时是否启用此功能。

如果攻击者可能触发对其他站点的请求（例如使用 HTTP 表单发布），则可以对使用已登录用户的 cookie 发出的某些请求进行发生原因 处理。

StorageGRID 可通过使用 CSRF 令牌帮助防止 CSRF 攻击。启用后，特定 Cookie 的内容必须与特定标题或特定后处理正文参数的内容匹配。

要启用此功能、请设置 `csrfToken` 参数设置为 `true` 身份验证期间。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、则为A `GridCsrfToken` Cookie会使用随机值设置为网格管理器和登录 `AccountCsrfToken` Cookie会使用随机值设置为登录到租户管理器。

如果存在 Cookie ，则可以修改系统状态的所有请求（POST， PUT， patch， delete）都必须包括以下项之一：

- 。 `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：A `csrfToken` 表单编码的请求正文参数。

有关其他示例和详细信息，请参见联机 API 文档。

> 设置了CSRF令牌Cookie的请求也将强制实施 `"Content-Type: application/json"` 任何请求的标头、如果希望JSON请求正文作为对CSRF攻击的额外保护、

如果启用了单点登录，请使用 API

如果启用了单点登录，请使用 API （Active Directory）

如果您有 "已配置并启用单点登录（SSO）" 如果您使用 Active Directory 作为 SSO 提供程序，则必须对一系列 API 请求进行问题描述 处理，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，则以下说明适用。

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 ，您知道租户帐户 ID 。

关于此任务

要获取身份验证令牌，可以使用以下示例之一：

- 。 `storagegrid-ssoauth.py` Python脚本、位于StorageGRID 安装文件目录中 (`./rpms` 对于Red Hat Enterprise Linux或CentOS、 `./debs` 适用于Ubuntu或Debian、和 `./vsphere` 适用于VMware)。
- cURL 请求的示例工作流。

  如果执行速度过慢，则卷曲工作流可能会超时。您可能会看到以下错误： `A valid SubjectConfirmation was not found on this Response`。

  > 示例 cURL 工作流不会保护密码不会被其他用户看到。

如果您使用的是URL编码问题描述 、则可能会看到以下错误： `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。转至步骤 2 。
   - 使用 curl 请求。转至步骤 3 。
2. 如果要使用 `storagegrid-ssoauth.py` 脚本、将脚本传递给Python解释器并运行脚本。

   出现提示时，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
   - 要访问租户管理 API 的租户帐户 ID 。

```
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
```

输出中提供了 StorageGRID 授权令牌。现在，您可以将令牌用于其他请求，类似于在未使用 SSO 时使用 API 的方式。

3. 如果要使用 curl 请求，请使用以下操作步骤 。

   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'
```

> 要访问网格管理API、请使用0作为 `TENANTACCOUNTID`。

   b. 要接收签名身份验证URL、问题描述 请将POST请求发送到 `/api/v3/authorize-saml`、并从响应中删除其他JSON编码。

   此示例显示了的已签名身份验证URL的POST请求 `TENANTACCOUNTID`。结果将传递到 `python -m json.tool` 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
```

   此示例的响应包括一个 URL 编码的签名 URL ，但不包括额外的 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. 保存 `SAMLRequest` 从响应中获取、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 。

一种方法是使用上一响应中的 URL 请求登录表单。

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

此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. 保存响应中的客户端请求 ID 。

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

f. 将您的凭据发送到上一响应中的表单操作。

```
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，并在标题中显示追加信息 。

> 如果为 SSO 系统启用了多因素身份验证（ MFA ），则此表单发布还将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. 保存 `MSISAuth` 响应中的cookie。

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

h. 使用身份验证 POST 中的 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，以便日后注销时使用，而响应正文将 SAMLResponse 隐藏在一个格式的字段中。

```
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. 保存 `SAMLResponse` 在隐藏字段中：

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

j. 使用已保存的 `SAMLResponse`、创建StorageGRID`/api/saml-response` 生成StorageGRID 身份验证令牌的请求。

适用于 `RelayState`、请使用租户帐户ID或如果要登录到网格管理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"
```

您现在可以使用 `MYTOKEN` 对于其他请求、类似于未使用SSO时使用API的方式。

如果启用了单点登录，请注销 API

如果已启用单点登录（ Single Sign-On ， SSO ），则必须对一系列 API 请求进行问题描述，才能注销网格管理 API 或租户管理 API 。如果您使用 Active Directory 作为 SSO 身份提供程序，则以下说明适用

关于此任务

如果需要、您可以从组织的单点注销页面注销、以注销StorageGRID API。或者，您也可以从 StorageGRID 触发单点注销（ SLO ），这需要有效的 StorageGRID 令牌。

步骤

1. 要生成签名注销请求、请传递 `cookie "sso=true"` 至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。此重定向位置不适用于纯 API 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不使用 SSO 相同。条件 `cookie "sso=true"` 如果未提供、则用户将从StorageGRID 中注销、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
```

答 `204 No Content` 响应指示用户现在已注销。

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

如果启用了单点登录，请使用 API （Azure）

如果您有 "已配置并启用单点登录（SSO）" 使用 Azure 作为 SSO 提供程序时，您可以使用两个示例脚本来获取对网格管理 API 或租户管理 API 有效的身份验证令牌。

如果启用了 Azure 单点登录，请登录到 API

如果您使用 Azure 作为 SSO 身份提供程序，则以下说明适用

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 ，您知道租户帐户 ID 。

关于此任务

要获取身份验证令牌，可以使用以下示例脚本：

- 。 `storagegrid-ssoauth-azure.py` Python 脚本
- 。 `storagegrid-ssoauth-azure.js` 节点.js脚本

这两个脚本都位于StorageGRID 安装文件目录中 (`./rpms` 对于Red Hat Enterprise Linux或CentOS、 `./debs` 适用于Ubuntu或Debian、和 `./vsphere` 适用于VMware)。

要编写您自己的与Azure的API集成、请参见 `storagegrid-ssoauth-azure.py` 脚本。Python 脚本会直接向 StorageGRID 发出两个请求（首先获取 SAMLRequest ，然后再获取授权令牌），同时还会调用 Node.js 脚本与 Azure 交互以执行 SSO 操作。

可以使用一系列 API 请求执行 SSO 操作，但这样做并不简单。puppeteer Node.js 模块用于擦除 Azure SSO 接口。

如果您使用的是URL编码问题描述 、则可能会看到以下错误： `Unsupported SAML version`。

步骤

1. 安装所需的依赖关系，如下所示：
   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 解释器以运行此脚本。

   然后， Python 脚本将调用相应的 Node.js 脚本以执行 Azure SSO 交互。

3. 出现提示时，输入以下参数的值（或使用参数传递这些值）：
   - 用于登录到 Azure 的 SSO 电子邮件地址
   - StorageGRID 的地址
   - 要访问租户管理 API 的租户帐户 ID
4. 出现提示时，输入密码，并在收到请求时准备向 Azure 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'}
```

> 此脚本假定 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、输入在文本消息中收到的代码)。

输出中提供了 StorageGRID 授权令牌。现在，您可以将令牌用于其他请求，类似于在未使用 SSO 时使用 API 的方式。

如果启用了单点登录，请使用 API （ PingFederate ）

如果您有 "已配置并启用单点登录（ SSO ）" 如果使用 PingFederate 作为 SSO 提供程序，则必须对一系列 API 请求进行问题描述 处理，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 身份提供程序，则以下说明适用

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 ，您知道租户帐户 ID 。

关于此任务

要获取身份验证令牌，可以使用以下示例之一：

- 。 `storagegrid-ssoauth.py` Python脚本、位于StorageGRID 安装文件目录中 (`./rpms` 对于Red Hat Enterprise Linux或CentOS、 `./debs` 适用于Ubuntu或Debian、和 `./vsphere` 适用于VMware)。
- cURL 请求的示例工作流。

  如果执行速度过慢，则卷曲工作流可能会超时。您可能会看到以下错误： `A valid SubjectConfirmation was not found on this Response`。

  > 示例 cURL 工作流不会保护密码不会被其他用户看到。

  如果您使用的是URL编码问题描述 、则可能会看到以下错误： `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。转至步骤 2 。
   - 使用 curl 请求。转至步骤 3 。
2. 如果要使用 `storagegrid-ssoauth.py` 脚本、将脚本传递给Python解释器并运行脚本。

   出现提示时，输入以下参数的值：

   - SSO 方法。您可以输入 "`pingFederate` " 的任何变体（ PNGFEDERATE ， 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
- 要访问租户管理 API 的租户帐户 ID 。

```
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
```

输出中提供了 StorageGRID 授权令牌。现在，您可以将令牌用于其他请求，类似于在未使用 SSO 时使用 API 的方式。

3. 如果要使用 curl 请求，请使用以下操作步骤 。

   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'
```

> 要访问网格管理API、请使用0作为 `TENANTACCOUNTID`。

   b. 要接收签名身份验证URL、问题描述 请将POST请求发送到 `/api/v3/authorize-saml`、并从响应中删除其他JSON编码。

   此示例显示了一个 POST 请求，用于为 TENANTACCOBTID 提供签名身份验证 URL 。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
```

   此示例的响应包括一个 URL 编码的签名 URL ，但不包括额外的 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. 保存 `SAMLRequest` 从响应中获取、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 ，并对响应执行回显：

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

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

e. 导出 "pf.adapterId" 值，并对响应执行回显：

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

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

f. 导出 "href" 值（删除后斜杠 / ），并对响应执行回显：

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

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

g. 导出 "act" 值：

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

h. 发送 Cookie 以及凭据：

```
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. 保存 `SAMLResponse` 在隐藏字段中：

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

j. 使用已保存的 `SAMLResponse`、创建StorageGRID`/api/saml-response` 生成StorageGRID 身份验证令牌的请求。

适用于 `RelayState`、请使用租户帐户ID或如果要登录到网格管理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"
```

您现在可以使用 `MYTOKEN` 对于其他请求、类似于未使用SSO时使用API的方式。

如果启用了单点登录，请注销 API

如果已启用单点登录（ Single Sign-On ， SSO ），则必须对一系列 API 请求进行问题描述 ，才能注销网格管理 API 或租户管理 API 。如果您使用 PingFederate 作为 SSO 身份提供程序，则以下说明适用

关于此任务

如果需要、您可以从组织的单点注销页面注销、以注销StorageGRID API。或者，您也可以从 StorageGRID 触发单点注销（ SLO ），这需要有效的 StorageGRID 令牌。

步骤

1. 要生成签名注销请求、请传递 `cookie "sso=true"` 至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。此重定向位置不适用于纯 API 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不使用 SSO 相同。条件 `cookie "sso=true"` 如果未提供、则用户将从StorageGRID 中注销、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
```

答 `204 No Content` 响应指示用户现在已注销。

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

使用 API 停用功能

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

关于此任务

停用的功能系统允许您阻止访问 StorageGRID 系统中的某些功能。停用某个功能是防止 root 用户或具有 * root 访问权限 * 的管理组中的用户能够使用该功能的唯一方法。

要了解此功能的有用程度，请考虑以下情形：

_Company A 是一家服务提供商，通过创建租户帐户租用其 StorageGRID 系统的存储容量。为了保护租户对象的安全， A 公司希望确保自己的员工在部署帐户后永远不能访问任何租户帐户。 _

_Company A 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中的 * 更改租户根密码 * 功能（ UI 和 API ），公司 A 可以确保任何管理员用户（包括 root 用户和属于具有 * root 访问权限 * 组的用户）都不能更改任何租户帐户的 root 用户的密码

步骤

1. 访问网格管理 API 的 Swagger 文档。请参见 "使用网格管理 API"。
2. 找到停用功能端点。
3. 要停用更改租户 root 密码等功能，请按如下所示向 API 发送正文：

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

   请求完成后，更改租户根密码功能将被禁用。用户界面中不再显示 * 更改租户根密码 * 管理权限，尝试更改租户根密码的任何 API 请求将失败，并显示 "`403 For禁用 。`

重新激活已停用的功能

默认情况下，您可以使用网格管理 API 重新激活已停用的功能。但是，如果要防止重新激活已停用的功能，则可以停用 * 激活功能 * 功能本身。

> ⚠ 无法重新激活*activateFeature*功能。如果您决定停用此功能，请注意，您将永远无法重新激活任何其他已停用的功能。要还原任何丢失的功能，您必须联系技术支持。

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，请按如下所示将正文发送到 API：

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

   此请求完成后，包括更改租户 root 密码功能在内的所有功能都将重新激活。现在，" 更改租户根密码 " 管理权限将显示在用户界面中，如果用户具有 * 根访问权限 * 或 * 更改租户根密码 * 管理权限，则尝试更改租户根密码的任何 API 请求都将成功。

   > 上一示例将重新激活 *all* 已停用的功能。如果其他功能已停用，而这些功能应保持停用状态，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 密码功能并继续停用警报确认功能，请发送此 PUT 请求：
   >
   > ```
   > { "grid": { "alarmAcknowledgment": true } }
   > ```