Die API verwenden

StorageGRID software

NetApp
July 23, 2026

Inhalt

Die API verwenden

Verwenden Sie die StorageGRID Grid Management API

Systemmanagement-Aufgaben können mithilfe der Grid Management REST API anstelle der Grid Manager Benutzeroberfläche durchgeführt werden. Beispielsweise kann die API verwendet werden, um Vorgänge zu automatisieren oder mehrere Entitäten, wie Benutzer, schneller zu erstellen.

Ressourcen der obersten Ebene

Die Grid Management API stellt die folgenden Ressourcen der obersten Ebene bereit:

- `/grid`: Der Zugriff ist auf Grid Manager Benutzer beschränkt und basiert auf den konfigurierten Gruppenberechtigungen.
- `/org`: Der Zugriff ist auf Benutzer beschränkt, die einer lokalen oder föderierten LDAP-Gruppe für ein Mandantenkonto angehören. Weitere Informationen finden sich unter "Ein Mandantenkonto verwenden".
- `/private`: Der Zugriff ist auf Grid Manager-Benutzer beschränkt und basiert auf den konfigurierten Gruppenberechtigungen. Die privaten APIs können sich ohne Vorankündigung ändern. StorageGRID private Endpunkte ignorieren außerdem die API-Version der Anfrage.

API-Anfragen ausführen

Die Grid Management API nutzt die Open-Source-API-Plattform Swagger. Swagger bietet eine intuitive Benutzeroberfläche, die es Entwicklern und Nicht-Entwicklern ermöglicht, Echtzeit-Operationen in StorageGRID mithilfe der API durchzuführen.

Die Swagger-Benutzeroberfläche bietet vollständige Details und Dokumentation für jede API-Operation.

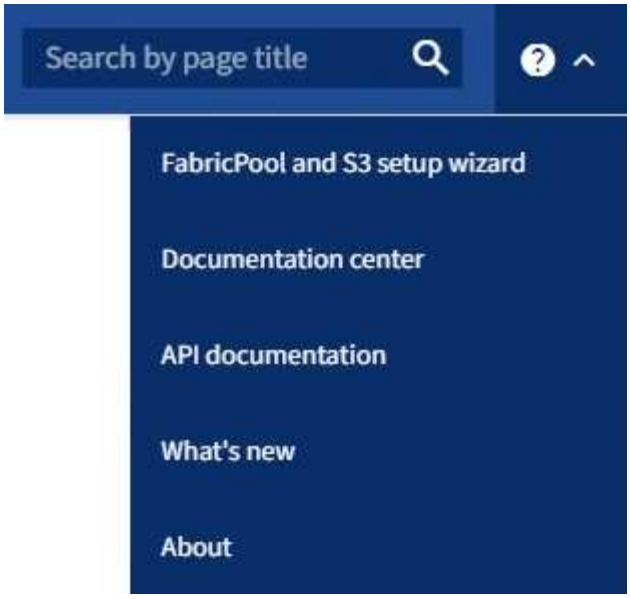
Bevor Sie beginnen

- Sie sind mit einem "unterstützte Webbrowser" beim Grid Manager angemeldet.
- Du hast "spezifische Zugriffsberechtigungen".

> Alle API-Operationen, die Sie über die API-Dokumentationswebseite durchführen, sind Live-Operationen. Es ist darauf zu achten, dass Konfigurationsdaten oder andere Daten nicht versehentlich erstellt, aktualisiert oder gelöscht werden.

Schritte

1. Im Grid Manager Header das Hilfesymbol auswählen und anschließend **API-Dokumentation** wählen.



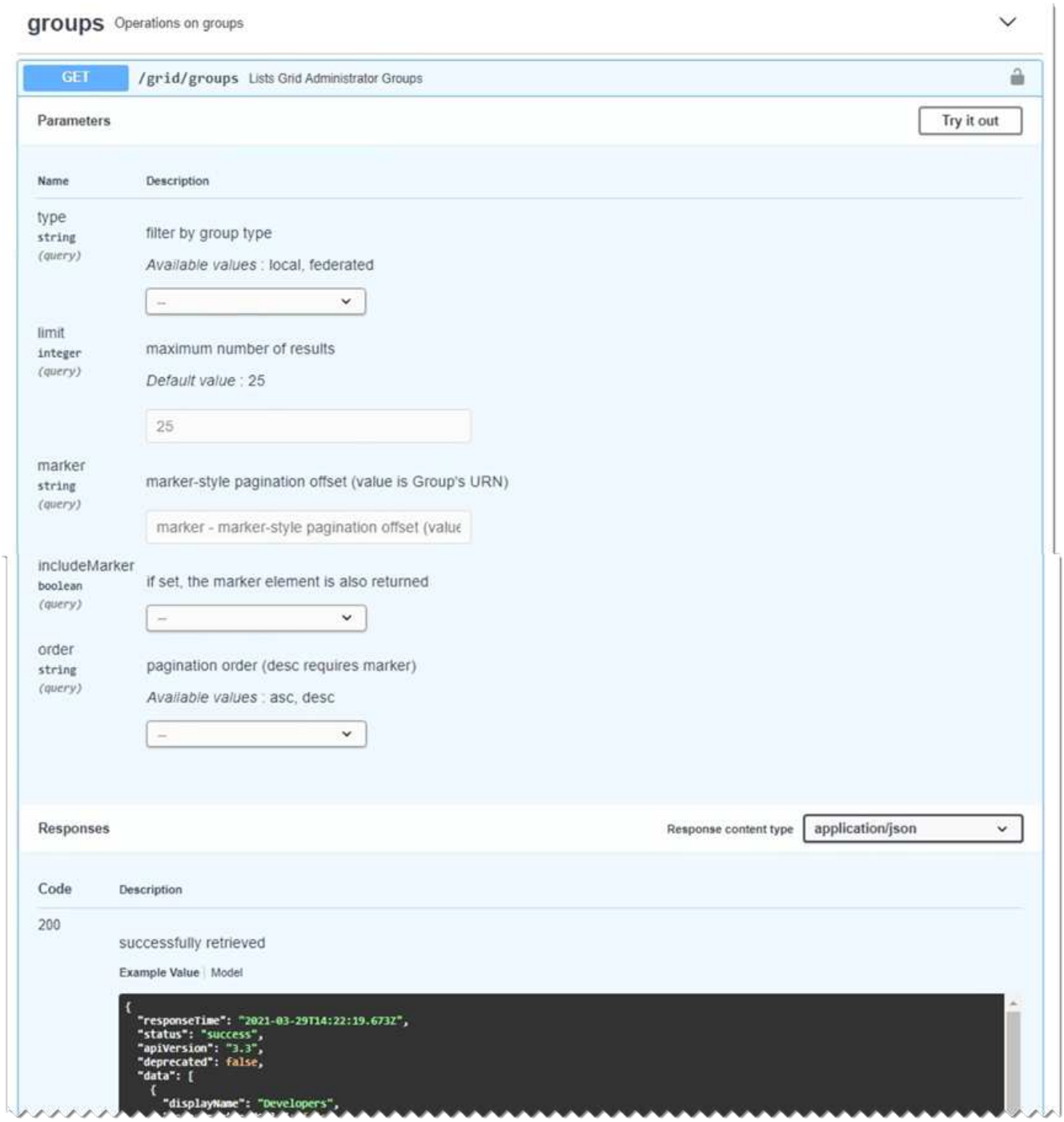
2. Um eine Operation mit der privaten API durchzuführen, auf der StorageGRID Management API-Seite **Zur Dokumentation der privaten API gehen** auswählen.

   Die privaten APIs können sich ohne Vorankündigung ändern. Private Endpunkte von StorageGRID ignorieren außerdem die API-Version der Anfrage.

3. Den gewünschten Vorgang auswählen.

   Wenn eine API-Operation erweitert wird, sind die verfügbaren HTTP-Aktionen wie GET, PUT, UPDATE und DELETE sichtbar.

4. Eine HTTP-Aktion kann ausgewählt werden, um die Anfragedetails einschließlich der Endpunkt-URL, einer Liste aller erforderlichen oder optionalen Parameter, eines Beispiels für den Anfragetext (falls erforderlich) und der möglichen Antworten anzuzeigen.



5. Es ist zu ermitteln, ob die Anforderung zusätzliche Parameter wie eine Gruppen- oder Benutzer-ID erfordert. Anschließend sind diese Werte abzurufen. Möglicherweise muss zuerst eine andere API-Anforderung gesendet werden, um die benötigten Informationen zu erhalten.
6. Es kann geprüft werden, ob der Beispiel-Anfragetext angepasst werden muss. Ist dies der Fall, steht **Model** zur Verfügung, um die Anforderungen für jedes Feld anzuzeigen.
7. **Ausprobieren** auswählen.
8. Alle erforderlichen Parameter können angegeben oder der Anfragetext nach Bedarf angepasst werden.
9. **Ausführen** auswählen.
10. Der Antwortcode gibt Aufschluss darüber, ob die Anfrage erfolgreich war.

Grid Management API-Operationen in StorageGRID

${post_edited_translations.segment}

Diese Liste enthält nur Operationen, die in der öffentlichen API verfügbar sind.

- **Konten**: Vorgänge zur Verwaltung von Speicher-Mandantenkonten, einschließlich der Erstellung neuer Konten und des Abrufs der Speichernutzung für ein bestimmtes Konto.
- ${post_edited_translations.segment}
- ${post_edited_translations.segment}
- ${post_edited_translations.segment}
- **alert-silences**: Vorgänge bei Alarm-Stummschaltungen.
- **Warnungen**: Vorgänge bei Warnmeldungen.
- **audit**: Vorgänge zum Auflisten und Aktualisieren der Audit-Konfiguration.
- **auth**: Operationen zur Durchführung der Benutzersitzungsauthentifizierung.

  Die Grid Management API unterstützt das Bearer Token Authentifizierungsschema. Für die Anmeldung werden ein Benutzername und ein Passwort im JSON-Body der Authentifizierungsanfrage angegeben (das heißt, `POST /api/v3/authorize`). Wenn der Benutzer erfolgreich authentifiziert wurde, wird ein Sicherheitstoken zurückgegeben. Dieses Token muss im Header nachfolgender API-Anfragen bereitgestellt werden ("Authorization: Bearer *token*"). Das Token läuft nach 16 Stunden ab.

  Wenn Single Sign-On für das StorageGRID System aktiviert ist, sind andere Schritte zur Authentifizierung erforderlich. Siehe „Authentifizierung an der API, wenn Single Sign-On aktiviert ist“.

  Weitere Informationen zur Verbesserung der Sicherheit bei der Authentifizierung finden Sie unter „Schutz vor Cross-Site Request Forgery“.

- **Clientzertifikate**: Vorgänge zur Konfiguration von Clientzertifikaten, sodass StorageGRID mithilfe externer Überwachungstools sicher aufgerufen werden kann.
- **config**: Operationen im Zusammenhang mit der Produktfreigabe und den Versionen der Grid Management API. Es besteht die Möglichkeit, die Produktfreigabeversion sowie die von dieser Freigabe unterstützten Hauptversionen der Grid Management API aufzulisten und veraltete Versionen der API zu deaktivieren.
- **deactivated-features**: Vorgänge zum Anzeigen von Funktionen, die möglicherweise deaktiviert wurden.
- **dns-servers**: Vorgänge zum Auflisten und Ändern konfigurierter externer DNS-Server.
- ${post_edited_translations.segment}
- **endpoint-domain-names**: Operationen zum Auflisten und Ändern von S3-Endpunkt-Domänennamen.
- ${post_edited_translations.segment}
- **Expansion**: Operationen bei der Expansion (Prozedurebene).
- ${post_edited_translations.segment}
- ${post_edited_translations.segment}
- **grid-networks**: Vorgänge zum Auflisten und Ändern der Grid-Netzwerkliste.
- ${post_edited_translations.segment}

- ${post_edited_translations.segment}
- ${post_edited_translations.segment}
- **ilm**: Vorgänge im Bereich des Information Lifecycle Management (ILM).
- **in-progress-procedures**: Ruft die Wartungsverfahren ab, die derzeit in Bearbeitung sind.
- ${post_edited_translations.segment}
- **Protokolle**: Vorgänge zum Sammeln und Herunterladen von Protokolldateien.v
- **Metriken**: Operationen an StorageGRID-Metriken, einschließlich sofortiger Metrikabfragen zu einem bestimmten Zeitpunkt und Bereichsabfragen von Metriken über einen Zeitraum. Die Grid Management API verwendet das Systemüberwachungstool Prometheus als Backend-Datenquelle. Informationen zur Erstellung von Prometheus-Abfragen sind auf der Prometheus-Website verfügbar.

  > Metriken, die `private` in ihrem Namen enthalten, sind ausschließlich für den internen Gebrauch bestimmt. Diese Metriken können sich zwischen StorageGRID Versionen ohne Vorankündigung ändern.

- **Knotendetails**: Operationen an Knotendetails.
- **Knotengesundheit**: Vorgänge zum Status der Knotengesundheit.
- **node-storage-state**: Vorgänge zum Speicherstatus des Knotens.
- **ntp-servers**: Operationen zum Auflisten oder Aktualisieren externer Network Time Protocol (NTP) Server.
- **Objekte**: Vorgänge an Objekten und Objektmetadaten.
- **Wiederherstellung**: Vorgänge für das Wiederherstellungsverfahren.
- **Wiederherstellungspaket**: Vorgänge zum Herunterladen des Wiederherstellungspakets.
- **Regionen**: Vorgänge zum Anzeigen und Erstellen von Regionen.
- **s3-object-lock**: Vorgänge an globalen S3 Object Lock-Einstellungen.
- **Serverzertifikat**: Vorgänge zum Anzeigen und Aktualisieren von Grid Manager Serverzertifikaten.
- **snmp**: Vorgänge an der aktuellen SNMP Konfiguration.
- **storage-watermarks**: Wasserzeichen für Storage-Node.
- **traffic-classes**: Vorgänge für Richtlinien zur Verkehrsklassifizierung.
- **untrusted-client-network**: Vorgänge an der Konfiguration des nicht vertrauenswürdigen Client-Netzwerks.
- **Benutzer**: Vorgänge zum Anzeigen und Verwalten von Grid Manager Benutzern.

Versionierung der Grid Management API in StorageGRID

Die Grid Management API verwendet Versionierung, um unterbrechungsfreie Upgrades zu unterstützen.

Beispielsweise gibt diese Request-URL die Version 4 der API an.

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

Die Hauptversion der API wird erhöht, wenn Änderungen vorgenommen werden, die nicht mit älteren Versionen kompatibel sind. Die Nebenversion der API wird erhöht, wenn Änderungen vorgenommen werden, die mit älteren Versionen kompatibel sind. Kompatible Änderungen umfassen das Hinzufügen neuer

Endpunkte oder neuer Eigenschaften.

Das folgende Beispiel veranschaulicht, wie die API-Version je nach Art der vorgenommenen Änderungen erhöht wird.

Art der Änderung an der API	${post_edited_translations.segment}	Neue Version
${post_edited_translations.segment}	2,1	2,2
Nicht kompatibel mit älteren Versionen	2,1	3,0

Bei der erstmaligen Installation der StorageGRID Software ist nur die neueste Version der API aktiviert. Wenn jedoch ein Upgrade auf eine neue Funktionsversion von StorageGRID erfolgt, bleibt der Zugriff auf die ältere API-Version für mindestens eine StorageGRID Funktionsversion erhalten.

> Sie können die unterstützten Versionen konfigurieren. Im Abschnitt **config** der Swagger API-Dokumentation "Grid Management API" finden sich weitere Informationen. Die Unterstützung für die ältere Version sollte nach der Aktualisierung aller API-Clients auf die neuere Version deaktiviert werden.

Veraltete Anfragen werden auf folgende Weise als veraltet gekennzeichnet:

- Der Antwortheader ist „Deprecated: true“.
- Der JSON-Antworttext enthält "deprecated": true
- Eine Warnung bezüglich veralteter Technologien wird in nms.log hinzugefügt. Zum Beispiel:

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

${post_edited_translations.segment}

Die `GET /versions`API-Anfrage gibt eine Liste der unterstützten API-Hauptversionen zurück. Diese Anfrage befindet sich im Abschnitt **config** der Swagger API-Dokumentation.

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

Eine API-Version für eine Anfrage angeben

Sie können die API-Version mit einem Pfadparameter (`/api/v4`) oder einem Header (`Api-Version: 4`) angeben. Wenn beide Werte angegeben werden, überschreibt der Wert im Header den Wert im Pfad.

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

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

Schutz vor Cross-Site Request Forgery (CSRF) in StorageGRID

Sie können zum Schutz vor Cross-Site-Request-Forgery-Angriffen (CSRF) auf StorageGRID beitragen, indem Sie CSRF-Token verwenden, um die Authentifizierung über Cookies zu verbessern. Der Grid Manager und der Tenant Manager aktivieren diese Sicherheitsfunktion automatisch; andere API-Clients können beim Anmelden auswählen, ob sie diese aktivieren möchten.

${post_edited_translations.segment}

StorageGRID schützt vor CSRF-Angriffen durch die Verwendung von CSRF-Tokens. Wenn diese Option aktiviert ist, muss der Inhalt eines bestimmten Cookies mit dem Inhalt eines bestimmten Headers oder eines bestimmten POST-Body-Parameters übereinstimmen.

Um die Funktion zu aktivieren, legen Sie den Parameter `csrfToken` während der Authentifizierung auf `true` fest. Der Standardwert ist `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"
```

Wenn dieser Wert auf „true“ festgelegt ist, wird ein `GridCsrfToken`-Cookie mit einem zufälligen Wert für Anmeldungen beim Grid Manager gesetzt, und das `AccountCsrfToken`-Cookie wird mit einem zufälligen Wert für Anmeldungen beim Tenant Manager gesetzt.

${post_edited_translations.segment}

- Der `X-Csrf-Token`-Header, wobei der Wert des Headers auf den Wert des CSRF-Token-Cookies festgelegt ist.
- Für Endpunkte, die einen formularcodierten Body akzeptieren: Ein `csrfToken` formularcodierter Anfragekörperparameter.

Zusätzliche Beispiele und Details sind in der Online-API-Dokumentation zu finden.

${post_edited_translations.segment}

Die API kann verwendet werden, wenn Single Sign-On aktiviert ist.

Verwenden Sie die StorageGRID API, wenn Single Sign-On aktiviert ist (Active Directory)

Wenn Sie "Single Sign-On (SSO) konfiguriert und aktiviert" haben und Active Directory als SSO-Anbieter verwenden, müssen Sie eine Reihe von API-Anfragen senden, um ein Authentifizierungstoken zu erhalten, das für die Grid Management API oder die Tenant Management API gültig ist.

Bei aktivierter Single Sign-On-Funktion erfolgt die Anmeldung an der API.

${post_edited_translations.segment}

Bevor Sie beginnen

- ${post_edited_translations.segment}
- ${post_edited_translations.segment}

Über diese Aufgabe

Um ein Authentifizierungs-Token zu erhalten, kann eines der folgenden Beispiele verwendet werden:

- Das `storagegrid-ssoauth.py` Python-Skript befindet sich im StorageGRID Installationsverzeichnis (`./rpms` für RHEL, `./debs` für Ubuntu oder Debian und `./vsphere` für VMware.

- Ein Beispielworkflow für curl-Anfragen.

  Der curl Workflow kann zu einem Timeout führen, wenn er zu langsam ausgeführt wird. Möglicherweise wird folgender Fehler angezeigt: `A valid SubjectConfirmation was not found on this Response`.

  > Der im Beispiel gezeigte curl Workflow schützt das Passwort nicht davor, von anderen Benutzern eingesehen zu werden.

Wenn ein Problem mit der URL-Codierung vorliegt, wird möglicherweise folgender Fehler angezeigt: `Unsupported SAML version`.

Schritte

1. Wählen Sie eine der folgenden Methoden, um ein Authentifizierungstoken zu erhalten:
   - Das Python-Skript `storagegrid-ssoauth.py` verwenden. Weiter mit Schritt 2.
   - Verwenden Sie curl Anfragen. Wechseln Sie zu Schritt 3.
2. Wenn Sie das `storagegrid-ssoauth.py` Skript verwenden möchten, übergeben Sie das Skript an den Python-Interpreter und führen das Skript aus.

   Geben Sie bei Aufforderung Werte für die folgenden Argumente ein:

   - Die SSO-Methode. ADFS oder adfs eingeben.
   - Der SSO Benutzername
   - Die Domäne, in der StorageGRID installiert ist
   - Die Adresse für StorageGRID
   - Die Mandantenkonto-ID, falls Sie auf die Tenant Management API zugreifen möchten.

   ```
   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
   ```

   Das StorageGRID Autorisierungstoken wird in der Ausgabe bereitgestellt. Sie können das Token nun für andere Anfragen verwenden, ähnlich wie Sie die API verwenden würden, wenn kein SSO verwendet wird.

3. ${post_edited_translations.segment}

   a. ${post_edited_translations.segment}

```
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'
```

> Für den Zugriff auf die Grid Management API ist 0 als `TENANTACCOUNTID` zu verwenden.

b. Um eine signierte Authentifizierungs-URL zu erhalten, wird eine POST-Anfrage an `/api/v3/authorize-saml` gesendet und die zusätzliche JSON-Codierung aus der Antwort entfernt.

Dieses Beispiel zeigt eine POST-Anforderung für eine signierte Authentifizierungs-URL für `TENANTACCOUNTID`. Die Ergebnisse werden an `python -m json.tool` übergeben, um die JSON-Codierung zu entfernen.

```
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
```

Die Antwort für dieses Beispiel enthält eine signierte URL, die URL-codiert ist, aber nicht die zusätzliche JSON-Codierungsschicht.

```
{
    "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. Den `SAMLRequest` aus der Antwort für die Verwendung in nachfolgenden Befehlen speichern.

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

d. Eine vollständige URL mit der Client Request-ID von AD FS erhalten.

Eine Möglichkeit besteht darin, das Anmeldeformular mit der URL aus der vorherigen Antwort anzufordern.

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

Die Antwort enthält die Client-Anforderungs-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. Die Client-Anforderungs-ID aus der Antwort speichern.

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

f. Ihre Zugangsdaten werden an die Formularaktion aus der vorherigen Antwort gesendet.

```
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 gibt eine 302-Weiterleitung zurück, mit zusätzlichen Informationen in den Headern.

Wenn für Ihr SSO-System die Multi-Faktor-Authentifizierung (MFA) aktiviert ist, enthält das Formular ebenfalls das zweite Passwort oder andere Anmeldeinformationen.

```
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. Den `MSISAuth`Cookie aus der Antwort speichern.

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

h. Eine GET-Anfrage wird an den angegebenen Speicherort mit den Cookies aus dem Authentifizierung-POST gesendet.

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

Die Antwortheader enthalten AD FS-Sitzungsinformationen für die spätere Abmeldung, und der Antworttext enthält die SAMLResponse in einem versteckten Formularfeld.

```
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. Die Daten `SAMLResponse` aus dem versteckten Feld speichern:

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

j. Mit den gespeicherten `SAMLResponse` kann eine StorageGRID`/api/saml-response`-Anfrage gestellt werden, um ein StorageGRID Authentifizierungs-Token zu generieren.

   Für `RelayState` verwenden Sie die Mandantenkonto-ID oder 0, wenn Sie sich bei der Grid Management API anmelden möchten.

```
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
```

   Die Antwort enthält das Authentifizierungs-Token.

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

a. Das Authentifizierungstoken in der Antwort als `MYTOKEN` speichern.

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

   Sie können jetzt `MYTOKEN` auch für andere Anfragen verwenden, ähnlich wie Sie die API verwenden würden, wenn SSO nicht verwendet wird.

${post_edited_translations.segment}

Wenn Single Sign-On (SSO) aktiviert wurde, müssen Sie eine Reihe von API-Anfragen senden, um sich von der Grid Management API oder der Tenant Management API abzumelden. Diese Anweisungen gelten, wenn Sie Active Directory als SSO-Identitätsanbieter verwenden

Über diese Aufgabe

Bei Bedarf können Sie sich von der StorageGRID API abmelden, indem Sie sich über die Single-Logout-Seite Ihrer Organisation abmelden. Alternativ können Sie das Single Logout (SLO) von StorageGRID auslösen, wofür ein gültiges StorageGRID Bearer-Token erforderlich ist.

Schritte

1. Um eine signierte Abmeldeanfrage zu generieren, wird `cookie "sso=true"` an die SLO API übergeben:

```
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
```

Eine Abmelde-URL wird zurückgegeben:

```
{
    "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. ${post_edited_translations.segment}

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

3. Eine Anfrage an die Logout-URL senden, um SLO auszulösen und zurück zu StorageGRID zu gelangen.

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

Es wird der Statuscode 302 zurückgegeben. Die Weiterleitungsadresse gilt nicht für die Abmeldung ausschließlich über die API.

```
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. ${post_edited_translations.segment}

Das Löschen des StorageGRID Bearer-Tokens funktioniert auf dieselbe Weise wie ohne SSO. Wenn das Cookie "sso=true" nicht bereitgestellt wird, wird der Benutzer von StorageGRID abgemeldet, ohne dass der SSO-Status beeinträchtigt wird.

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

Eine `204 No Content`Antwort signalisiert, dass der Benutzer jetzt abgemeldet ist.

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

Verwenden Sie die StorageGRID API, wenn Single Sign-On aktiviert ist (Entra ID)

Wenn Sie über "Single Sign-On (SSO) konfiguriert und aktiviert" verfügen und Entra ID als SSO-Anbieter verwenden, können Sie zwei Beispielskripte verwenden, um ein Authentifizierungstoken abzurufen, das für die Grid Management API oder die Tenant Management API gültig ist.

Anmeldung bei der API, wenn die Entra ID Single Sign-on-Funktion aktiviert ist

Diese Anweisungen gelten, wenn Sie Entra ID als SSO-Identitätsanbieter verwenden

Bevor Sie beginnen

- ${post_edited_translations.segment}
- ${post_edited_translations.segment}

Über diese Aufgabe

${post_edited_translations.segment}

- Das `storagegrid-ssoauth-azure.py`Python Skript
- Das `storagegrid-ssoauth-azure.js`Node.js Skript

Beide Skripte befinden sich im StorageGRID Installationsverzeichnis (`./rpms` für RHEL, `./debs` für Ubuntu oder Debian und `./vsphere` für VMware).

Um eine eigene API-Integration mit Entra ID zu erstellen, siehe das `storagegrid-ssoauth-azure.py` Skript. Das Python-Skript sendet zwei Anfragen direkt an StorageGRID (zuerst zum Abrufen der SAMLRequest und später zum Abrufen des Autorisierungstokens) und ruft außerdem das Node.js-Skript auf, um mit Entra ID zu interagieren und die SSO-Operationen auszuführen.

SSO-Operationen können mithilfe einer Reihe von API-Anfragen ausgeführt werden, jedoch ist dies nicht unkompliziert. Das Puppeteer Node.js Modul wird verwendet, um die Entra ID SSO-Oberfläche auszulesen.

Wenn ein Problem mit der URL-Codierung vorliegt, wird möglicherweise folgender Fehler angezeigt: `Unsupported SAML version.`

Schritte

1. ${post_edited_translations.segment}

a. Node.js installieren (siehe "https://nodejs.org/en/download/").

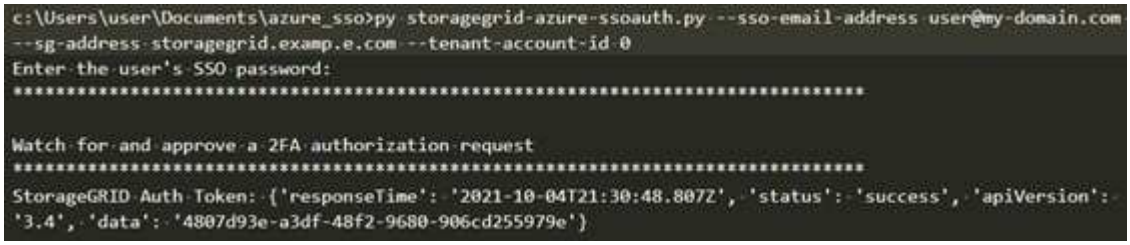
b. Die erforderlichen Node.js Module (puppeteer und jsdom) installieren:

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

2. Das Python-Skript wird dem Python-Interpreter übergeben, um das Skript auszuführen.

   Anschließend ruft das Python-Skript das entsprechende Node.js-Skript auf, um die Entra ID SSO-Interaktionen durchzuführen.

3. ${post_edited_translations.segment}
   - Die SSO-E-Mail-Adresse, die für die Anmeldung bei Entra ID verwendet wird
   - Die Adresse für StorageGRID
   - Die Mandantenkonto-ID, wenn auf die Tenant Management API zugegriffen werden soll
4. ${post_edited_translations.segment}



> Das Skript geht davon aus, dass die Multi-Faktor-Authentifizierung (MFA) mit Microsoft Authenticator durchgeführt wird. Möglicherweise ist eine Anpassung des Skripts erforderlich, um andere Formen der MFA zu unterstützen (wie zum Beispiel die Eingabe eines per SMS erhaltenen Codes).

Das StorageGRID Autorisierungstoken wird in der Ausgabe bereitgestellt. Sie können das Token nun für andere Anfragen verwenden, ähnlich wie Sie die API verwenden würden, wenn kein SSO verwendet wird.

Verwenden Sie die StorageGRID API, wenn Single Sign-On aktiviert ist (PingFederate)

Wenn Sie "Single Sign-On (SSO) konfiguriert und aktiviert" haben und PingFederate als SSO-Anbieter verwenden, muss eine Reihe von API-Anfragen gestellt werden, um ein Authentifizierungs-Token zu erhalten, das für die Grid Management API oder die Tenant Management API gültig ist.

Bei aktivierter Single Sign-On-Funktion erfolgt die Anmeldung an der API.

Diese Anweisungen gelten, wenn Sie PingFederate als SSO-Identitätsanbieter verwenden.

Bevor Sie beginnen

- ${post_edited_translations.segment}
- ${post_edited_translations.segment}

Über diese Aufgabe

Um ein Authentifizierungs-Token zu erhalten, kann eines der folgenden Beispiele verwendet werden:

- Das `storagegrid-ssoauth.py` Python-Skript befindet sich im StorageGRID Installationsverzeichnis (`./rpms` für RHEL, `./debs` für Ubuntu oder Debian und `./vsphere` für VMware.
- Ein Beispielworkflow für curl-Anfragen.

  Der curl Workflow kann zu einem Timeout führen, wenn er zu langsam ausgeführt wird. Möglicherweise wird folgender Fehler angezeigt: `A valid SubjectConfirmation was not found on this Response`.

  > Der im Beispiel gezeigte curl Workflow schützt das Passwort nicht davor, von anderen Benutzern eingesehen zu werden.

  Wenn ein Problem mit der URL-Codierung vorliegt, wird möglicherweise folgender Fehler angezeigt: `Unsupported SAML version`.

Schritte

1. Wählen Sie eine der folgenden Methoden, um ein Authentifizierungstoken zu erhalten:
   - Das Python-Skript `storagegrid-ssoauth.py` verwenden. Weiter mit Schritt 2.
   - Verwenden Sie curl Anfragen. Wechseln Sie zu Schritt 3.
2. Wenn Sie das `storagegrid-ssoauth.py` Skript verwenden möchten, übergeben Sie das Skript an den Python-Interpreter und führen das Skript aus.

   Geben Sie bei Aufforderung Werte für die folgenden Argumente ein:

   - Die SSO-Methode. Es kann jede beliebige Variante von "pingfederate" eingegeben werden (PINGFEDERATE, pingfederate und so weiter).
   - Der SSO Benutzername
   - Die Domäne, in der StorageGRID installiert ist. Dieses Feld wird nicht für PingFederate verwendet. Das Feld kann leer bleiben oder es kann ein beliebiger Wert eingegeben werden.
   - Die Adresse für StorageGRID
   - Die Mandantenkonto-ID, falls Sie auf die Tenant Management API zugreifen möchten.

   ```
   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
   ```

   Das StorageGRID Autorisierungstoken wird in der Ausgabe bereitgestellt. Sie können das Token nun für andere Anfragen verwenden, ähnlich wie Sie die API verwenden würden, wenn kein SSO verwendet wird.

3. ${post_edited_translations.segment}

   a. ${post_edited_translations.segment}

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

> Für den Zugriff auf die Grid Management API ist 0 als `TENANTACCOUNTID` zu verwenden.

   b. Um eine signierte Authentifizierungs-URL zu erhalten, wird eine POST-Anfrage an `/api/v3/authorize-saml` gesendet und die zusätzliche JSON-Codierung aus der Antwort entfernt.

   Dieses Beispiel zeigt eine POST-Anfrage für eine signierte Authentifizierungs-URL für TENANTACCOUNTID. Die Ergebnisse werden an `python -m json.tool` übergeben, um die JSON-Kodierung zu entfernen.

```
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
```

   Die Antwort für dieses Beispiel enthält eine signierte URL, die URL-codiert ist, aber nicht die zusätzliche JSON-Codierungsschicht.

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

   c. Den `SAMLRequest` aus der Antwort für die Verwendung in nachfolgenden Befehlen speichern.

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

   d. Die Antwort und den Cookie exportieren und die Antwort ausgeben:

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

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

e. Den Wert 'pf.adapterId' exportieren und die Antwort ausgeben:

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

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

f. Exportieren Sie den 'href'-Wert (entfernen Sie den abschließenden Schrägstrich /), und geben Sie die Antwort aus:

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

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

g. Den Wert „action“ exportieren:

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

h. Cookies zusammen mit den Anmeldeinformationen senden:

```
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. Die Daten `SAMLResponse` aus dem versteckten Feld speichern:

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

j. Mit den gespeicherten `SAMLResponse` kann eine StorageGRID`/api/saml-response`-Anfrage gestellt werden, um ein StorageGRID Authentifizierungs-Token zu generieren.

   Für `RelayState` verwenden Sie die Mandantenkonto-ID oder 0, wenn Sie sich bei der Grid Management API anmelden möchten.

```
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
```

Die Antwort enthält das Authentifizierungs-Token.

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

a. Das Authentifizierungstoken in der Antwort als `MYTOKEN` speichern.

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

Sie können jetzt `MYTOKEN` auch für andere Anfragen verwenden, ähnlich wie Sie die API verwenden würden, wenn SSO nicht verwendet wird.

${post_edited_translations.segment}

Wenn Single Sign-On (SSO) aktiviert wurde, müssen Sie eine Reihe von API-Anfragen senden, um sich von der Grid Management API oder der Tenant Management API abzumelden. Diese Anweisungen gelten, wenn Sie PingFederate als SSO-Identitätsanbieter verwenden

Über diese Aufgabe

Bei Bedarf können Sie sich von der StorageGRID API abmelden, indem Sie sich über die Single-Logout-Seite Ihrer Organisation abmelden. Alternativ können Sie das Single Logout (SLO) von StorageGRID auslösen, wofür ein gültiges StorageGRID Bearer-Token erforderlich ist.

Schritte

1. Um eine signierte Abmeldeanfrage zu generieren, wird `cookie "sso=true"` an die SLO API übergeben:

```
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
```

Eine Abmelde-URL wird zurückgegeben:

```
{
    "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. ${post_edited_translations.segment}

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

3. Eine Anfrage an die Logout-URL senden, um SLO auszulösen und zurück zu StorageGRID zu gelangen.

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

Es wird der Statuscode 302 zurückgegeben. Die Weiterleitungsadresse gilt nicht für die Abmeldung ausschließlich über die API.

```
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. ${post_edited_translations.segment}

Das Löschen des StorageGRID Bearer-Tokens funktioniert auf dieselbe Weise wie ohne SSO. Wenn das Cookie "sso=true" nicht bereitgestellt wird, wird der Benutzer von StorageGRID abgemeldet, ohne dass der SSO-Status beeinträchtigt wird.

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

Eine `204 No Content`Antwort signalisiert, dass der Benutzer jetzt abgemeldet ist.

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

StorageGRID-Funktionen mit der API deaktivieren

Mithilfe der Grid Management API kann die vollständige Deaktivierung bestimmter Funktionen im StorageGRID System erfolgen. Wenn eine Funktion deaktiviert ist, kann niemand Berechtigungen zur Ausführung der mit dieser Funktion verbundenen Aufgaben erhalten.

Über diese Aufgabe

Das System „Deaktivierte Funktionen“ ermöglicht es, den Zugriff auf bestimmte Funktionen im StorageGRID System zu verhindern. Die Deaktivierung einer Funktion ist die einzige Möglichkeit, den Root-Benutzer oder Benutzer, die Administratorgruppen mit **Root access**-Berechtigung angehören, an der Nutzung dieser Funktion zu hindern.

Um zu verstehen, wie diese Funktionalität nützlich sein kann, ist folgendes Szenario zu berücksichtigen:

Unternehmen A ist ein Dienstleister, der die Speicherkapazität seines StorageGRID Systems durch die Erstellung von Mandantenkonten vermietet. Um die Sicherheit der Objekte seiner Mieter zu schützen, möchte Unternehmen A sicherstellen, dass seine eigenen Mitarbeiter nach der Bereitstellung eines Mandantenkontos niemals auf ein Mandantenkonto zugreifen können.

Unternehmen A kann dieses Ziel erreichen, indem die Funktion „Funktionen deaktivieren“ im Grid Management API verwendet wird. Durch die vollständige Deaktivierung der Funktion **Change tenant root password** im Grid Manager (sowohl in der UI als auch in der API) wird sichergestellt, dass Admin-Benutzer – einschließlich des Root-Benutzers und von Benutzern, die Gruppen mit der Berechtigung **Root access** angehören – das Passwort für den Root-Benutzer eines beliebigen Mandantenkontos nicht ändern können.

Schritte

1. Auf die Swagger-Dokumentation für die Grid Management API kann zugegriffen werden. Siehe "Die Grid Management API verwenden".
2. Den Endpunkt „Funktionen deaktivieren“ finden.
3. Um eine Funktion wie das Ändern des Root-Benutzer-Passworts des Mandanten zu deaktivieren, wird ein Body wie folgt an die API gesendet:

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

   Nach Abschluss der Anfrage wird die Funktion „Change tenant root password“ deaktiviert. Die Verwaltungsberechtigung **Change tenant root password** wird nicht mehr in der Benutzeroberfläche angezeigt, und jede API-Anfrage, die versucht, das Root-Passwort eines Mandanten zu ändern, schlägt mit „403 Forbidden“ fehl.

Deaktivierte Funktionen reaktivieren

Standardmäßig kann die Grid Management API verwendet werden, um eine deaktivierte Funktion zu reaktivieren. Wenn jedoch verhindert werden soll, dass deaktivierte Funktionen jemals wieder aktiviert werden, kann die Funktion **activateFeatures** selbst deaktiviert werden.

Die Funktion **activateFeatures** kann nicht reaktiviert werden. Wenn Sie sich entscheiden, diese Funktion zu deaktivieren, ist zu beachten, dass die Möglichkeit, andere deaktivierte Funktionen erneut zu aktivieren, dauerhaft verloren geht. Um verlorene Funktionen wiederherzustellen, ist der technische Support zu kontaktieren.

Schritte

1. Auf die Swagger-Dokumentation für die Grid Management API kann zugegriffen werden.
2. Den Endpunkt „Funktionen deaktivieren“ finden.
3. Um alle Funktionen wieder zu aktivieren, wird ein Body wie folgt an die API gesendet:

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

   Nach Abschluss dieser Anfrage werden alle Funktionen, einschließlich der Funktion zum Ändern des Mandanten-Root-Passworts, reaktiviert. Die Verwaltungsberechtigung **Change tenant root password** wird nun in der Benutzeroberfläche angezeigt, und jede API-Anfrage, die versucht, das Root-Passwort eines Mandanten zu ändern, ist erfolgreich, sofern der Benutzer über die Verwaltungsberechtigung **Root access** oder **Change tenant root password** verfügt.

Das vorherige Beispiel bewirkt, dass *alle* deaktivierten Funktionen wieder aktiviert werden. Wenn weitere Funktionen deaktiviert wurden, die deaktiviert bleiben sollen, müssen diese explizit in der PUT-Anfrage angegeben werden. Um beispielsweise die Funktion „Change tenant root password“ wieder zu aktivieren und die storageAdmin Management-Berechtigung weiterhin deaktiviert zu lassen, ist folgende PUT-Anfrage zu senden: `{ "grid": {"storageAdmin": true} }`