



API を使用します

StorageGRID

NetApp
October 03, 2025

目次

API を使用します	1
グリッド管理 API を使用します	1
トップレベルのリソース	1
問題 API 要求	1
グリッド管理 API の処理	4
グリッド管理 API のバージョン管理	5
現在のリリースでサポートされている API のバージョンを確認します	6
要求の API バージョンを指定します	7
クロスサイトリクエストフォージェリ（CSRF）の防止	7
シングルサインオンが有効な場合は、API を使用します	8
シングルサインオンが有効な場合（Active Directory）は API を使用	8
シングルサインオンが有効な場合（Azure）は API を使用	15
シングルサインオンが有効な場合は API を使用（PingFederate）	16

API を使用します

グリッド管理 API を使用します

Grid Manager のユーザインターフェイスの代わりにグリッド管理 REST API を使用して、システム管理タスクを実行できます。たとえば、API を使用して処理を自動化したり、ユーザなどの複数のエンティティを迅速に作成したりできます。

トップレベルのリソース

グリッド管理 API で使用可能な最上位のリソースは次のとおりです。

- /grid : アクセスは Grid Manager ユーザに制限され、設定されたグループ権限に基づいています。
- /org: テナントアカウントのローカル LDAP グループまたはフェデレーション LDAP グループに属しているユーザにのみアクセスが許可されます詳細については、[を参照してください](#) [テナントアカウントを使用する](#)。
- 「/private」 : アクセスは Grid Manager ユーザに制限され、設定されたグループ権限に基づいて行われます。プライベート API は予告なく変更される場合があります。StorageGRID プライベートエンドポイントは、要求の API バージョンも無視します。

問題 API 要求

グリッド管理 API では、Swagger オープンソース API プラットフォームを使用します。Swagger のわかりやすいユーザインターフェイスを使用して、開発者および一般のユーザは StorageGRID で API を使用してリアルタイムの処理を実行できます。

Swagger のユーザインターフェイスでは、各 API 処理に関する詳細情報とドキュメントを参照できます。

必要なもの

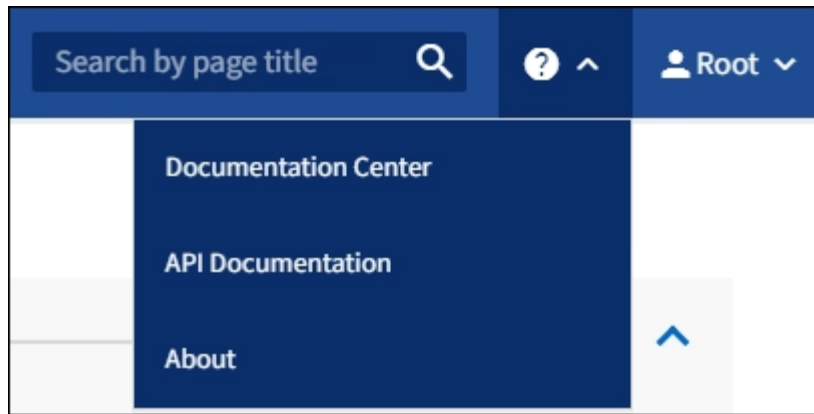
- を使用して Grid Manager にサインインします [サポートされている Web ブラウザ](#)。
- 特定のアクセス権限が必要です。



API Docs Web ページを使用して実行する API 処理はすべてその場で実行されます。設定データやその他のデータを誤って作成、更新、または削除しないように注意してください。

手順

1. Grid Manager ヘッダーでヘルプアイコンを選択し、* API ドキュメント * を選択します。



2. プライベート API を使用して操作を実行するには、StorageGRID 管理 API ページで * プライベート API ドキュメントへ移動 * を選択します。

プライベート API は予告なく変更される場合があります。StorageGRID プライベートエンドポイントは、要求の API バージョンも無視します。

3. 目的の処理を選択します。

API 処理を拡張すると、GET、PUT、UPDATE、DELETE など、使用可能な HTTP アクションを確認できます。

4. HTTP アクションを選択して、要求の詳細を確認します。これには、エンドポイント URL、必須またはオプションのパラメータのリスト、要求の本文の例（必要な場合）、想定される応答が含まれます。

GET /grid/groups Lists Grid Administrator Groups

Parameters Try it out

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

Responses Response content type application/json

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

- グループやユーザの ID など、要求で追加のパラメータが必要かどうかを確認します。次に、これらの値を取得します。必要な情報を取得するために、先に別の API 要求の問題が必要になることがあります。
- 要求の本文の例を変更する必要があるかどうかを判断します。その場合は、* Model * を選択して各フィールドの要件を確認できます。
- [* 試してみてください *] を選択します。
- 必要なパラメータを指定するか、必要に応じて要求の本文を変更します。
- [* Execute] を選択します。
- 応答コードを確認し、要求が成功したかどうかを判断します。

グリッド管理 API の処理

グリッド管理 API では、使用可能な処理が次のセクションに分類されます。



このリストには、パブリック API で使用可能な処理のみが含まれます。

- ***accounts*** — 新規アカウントの作成や特定の使用状況の取得など 'ストレージ・テナント・アカウントを管理するためのオペレーション
- ***alarms*** - 現在のアラーム（レガシーシステム）をリストし、現在のアラートやノード接続状態の概要など、グリッドの健全性に関する情報を返す処理。
- ***alert-history*** — 解決済みアラートに関する操作。
- ***alert-Receiver*** — アラート通知受信者（電子メール）に関する操作。
- ***alert-rules*** — アラートルールに関する操作
- ***alert-silences*** -- アラートのサイレンスに関するオペレーション。
- ***alerts*** — アラートの処理。
- ***audit*** — 監査構成をリストおよび更新する処理。
- **auth** — ユーザセッション認証を実行するための操作。

グリッド管理 API は、ベアトークン認証方式をサポートしています。サインインするには、認証要求（「POST /api/v3/authorize」）の JSON の本文でユーザ名とパスワードを指定します。ユーザが認証されると、セキュリティトークンが返されます。このトークンは、後続の API 要求（「Authorization : Bearer_token_」）のヘッダーで指定する必要があります。



StorageGRID システムでシングルサインオンが有効になっている場合は、別の手順による認証が必要です。「シングルサインオンが有効な場合の API へのサインイン」を参照してください。

認証セキュリティの向上については、「クロスサイトリクエストフォージェリに対する保護」を参照してください。

- ***client-certificates*** — 外部監視ツールを使用して StorageGRID に安全にアクセスできるようにクライアント証明書を設定する処理。
- **config** — 製品リリースと Grid Management API のバージョンに関連する操作。製品のリリースバージョンおよびそのリリースでサポートされているグリッド管理 API のメジャーバージョンをリストし、廃止されたバージョンの API を無効にすることができます。
- ***deactivated-features*** — 非アクティブ化された可能性のある機能を表示する操作。
- ***dns-servers*** — 設定済みの外部 DNS サーバをリストおよび変更する処理。
- ***endpoint-domain-names*** — エンドポイントドメイン名をリストおよび変更する処理。
- ***erasure-coding*** — イレイジャーコーディングプロファイルに対する処理。
- ***expansion*** -- 拡張の操作 (プロシージャレベル) 。
- ***expansion-nodes*** - 拡張処理 (ノードレベル) 。
- ***expansion-sitites*** — 拡張の操作 (サイトレベル) 。

- * grid-networks * — グリッドネットワークリストをリストおよび変更する処理。
- * grid-password * - グリッドパスワード管理の操作。
- * groups * — ローカルグリッド管理者グループを管理し、フェデレーテッドグリッド管理者グループを外部 LDAP サーバから取得するための処理。
- * identity-source * — 外部のアイデンティティソースを設定する処理、およびフェデレーテッドグループとユーザ情報を手動で同期する処理。
- * ilm * — 情報ライフサイクル管理 (ILM; 情報ライフサイクル管理) の操作。
- **license** — StorageGRID ライセンスを取得および更新する処理。
- **logs** — ログファイルを収集してダウンロードするための操作。
- * メトリクス * — ある時点での瞬時の指標クエリや、一定期間にわたる指標クエリなど、StorageGRID メトリックに対する処理。グリッド管理 API は、バックエンドのデータソースとして Prometheus システム監視ツールを使用します。Prometheus クエリの構築については、Prometheus の Web サイトを参照してください。



名前に「*private*」を含むメトリックは内部専用です。これらの指標は、StorageGRID のリリース間で予告なく変更される可能性があります。

- * node-details * - ノードの詳細に対する処理。
- * node-health * - ノードのヘルスステータスに関する処理。
- * ntp-servers * — 外部ネットワークタイムプロトコル (NTP) サーバをリストまたは更新する処理。
- * objects * — オブジェクトおよびオブジェクトメタデータに対する処理。
- **recovery** — リカバリ手順 の処理。
- * recovery-package * — リカバリパッケージをダウンロードする処理。
- * regions * — 領域の表示と作成のための操作。
- * s3-object-lock * — グローバルな S3 オブジェクトロック設定に対する処理。
- * server-certificate * — Grid Manager サーバ証明書を表示および更新する処理。
- * snmp * — 現在の SNMP 設定に対する操作。
- * traffic-classes * -- トラフィック分類ポリシーの操作。
- * untrusted-client-network * — 信頼されていないクライアントネットワーク構成に対する操作。
- * users * — Grid Manager ユーザーを表示および管理する操作。

グリッド管理 API のバージョン管理

グリッド管理 API では、バージョン管理を使用して無停止アップグレードがサポートされます。

たとえば、次の要求 URL ではバージョン 3 の API が指定されています。

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

旧バージョンとの互換性がない * `_not compatible_` * の変更が行われると、テナント管理 API のメジャーバー

ジョンが上がります。以前のバージョンと互換性がある `_*` の変更を行うと、テナント管理 API のマイナーバージョンが上がります。互換性のある変更には、新しいエンドポイントやプロパティの追加などがあります。次の例は、変更のタイプに基づいて API バージョンがどのように更新されるかを示しています。

API に対する変更のタイプ	古いバージョン	新しいバージョン
旧バージョンと互換性があります	2.1	2.2.
旧バージョンとの互換性はありません	2.1	3.0

StorageGRID ソフトウェアを初めてインストールした時点では、グリッド管理 API の最新のバージョンのみが有効になっています。ただし、StorageGRID の新機能リリースにアップグレードした場合、少なくとも StorageGRID の機能リリース 1 つ分の間は、古い API バージョンにも引き続きアクセスできます。



グリッド管理 API を使用して、サポートされるバージョンを設定できます。詳細については、Swagger API のドキュメントの「config」セクションを参照してください。すべての Grid 管理 API クライアントを新しいバージョンを使用するように更新したら、古いバージョンのサポートを無効にする必要があります。

古い要求は、次の方法で廃止とマークされます。

- 応答ヘッダーが「Deprecated : true」となる。
- JSON 応答の本文に「deprecated : true」が追加される
- 廃止の警告が `nms.log` に追加される。例：

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

現在のリリースでサポートされている API のバージョンを確認します

サポートされている API のメジャーバージョンのリストを返すには、次の API 要求を使用します。

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


要求の API バージョンを指定します

API バージョンは 'パス・パラメータ (/api/v3)' またはヘッダー ('api-Version:3') を使用して指定できます両方の値を指定した場合は、ヘッダー値がパス値よりも優先されます。

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

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

クロスサイトリクエストフォージェリ（CSRF）の防止

CSRF トークンを使用してクッキーによる認証を強化すると、StorageGRID に対するクロスサイトリクエストフォージェリ（CSRF）攻撃を防ぐことができます。Grid Manager と Tenant Manager はこのセキュリティ機能を自動的に有効にします。他の API クライアントは、サインイン時にこの機能を有効にするかどうかを選択できます。

攻撃者が別のサイト（たとえば、HTTP フォーム POST を使用して）への要求をトリガーできる場合、サインインしているユーザのクッキーを使用して特定の要求を原因が送信できます。

StorageGRID では、CSRF トークンを使用して CSRF 攻撃を防ぐことができます。有効にした場合、特定のクッキーの内容が特定のヘッダーまたは特定の POST パラメータの内容と一致する必要があります。

この機能を有効にするには '認証時に csrfToken パラメータを true に設定しますデフォルトは「false」です。

```
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"
```

true に設定すると 'GridCsrfToken' クッキーが Grid Manager へのサインインにランダムな値を使用して設定され 'AccountCsrfToken' クッキーが Tenant Manager へのサインインにランダムな値を使用して設定されます

クッキーが存在する場合は、システムの状態を変更できるすべての要求（POST、PUT、PATCH、DELETE）には次のいずれかが含まれている必要があります。

- CSRF トークンクッキーの値が設定された 'X-Csrf-Token' ヘッダー
- フォームエンコードされた本文を受け入れるエンドポイントの場合：フォームエンコードされた要求本文パラメータ「csrfToken」。

その他の例および詳細については、オンラインの API ドキュメントを参照してください。



CSRF トークンクッキーが設定されている要求では、本文に JSON が必要なすべての要求に対して「Content-Type : application/json」ヘッダーも適用され、CSRF 攻撃からの保護がさらに強化されます。

シングルサインオンが有効な場合は、API を使用します

シングルサインオンが有効な場合（**Active Directory**）は **API** を使用

ある場合 **シングルサインオン（SSO）の設定と有効化** また、Active Directory を SSO プロバイダとして使用する場合は、一連の API 要求を **問題** で実行して、グリッド管理 API またはテナント管理 API で有効な認証トークンを取得する必要があります。

シングルサインオンが有効な場合は、**API** にサインインします

ここで説明する手順は、Active Directory を SSO アイデンティティプロバイダとして使用する場合に該当します。

必要なもの

- StorageGRID ユーザグループに属するフェデレーテッドユーザの SSO ユーザ名とパスワードが必要です。
- テナント管理 API にアクセスする場合は、テナントアカウント ID を確認しておきます。

このタスクについて

認証トークンを取得するには、次のいずれかの例を使用します。

- StorageGRID インストールファイルディレクトリ (Red Hat Enterprise Linux または CentOS の場合は「`./rpms`」、Ubuntu または Debian の場合は「`./debs`」、VMware の場合は「`./vsphere-vsphere`」) にある「`storagegrid-ssoauth.py`」 `p`python` スクリプト。
- cURL 要求のワークフローの例。

cURL ワークフローは、実行に時間がかかりすぎるとタイムアウトする場合があります。「A valid SubjectConfirmation was not found on this Response」というエラーが表示される可能性があります。



cURL ワークフローの例では、パスワードが他のユーザに表示されないように保護されていません。

URL エンコーディング問題 を使用している場合は、「Unsupported SAML version」というエラーが表示される可能性があります。

手順

1. 認証トークンを取得するには、次のいずれかの方法を選択します。
 - 「`storagegrid -ssoauth.py`」 Python スクリプトを使用します。手順 2 に進みます。
 - `curl` 要求を使用します。手順 3 に進みます。
2. 「`storagegrid -ssoauth.py`」スクリプトを使用する場合は、Python インタープリタにスクリプトを渡してスクリプトを実行します。

プロンプトが表示されたら、次の引数の値を入力します。

- SSO 方式。ADFS または ADFS と入力します。
- SSO ユーザ名
- StorageGRID がインストールされているドメイン
- StorageGRID のアドレス
- テナント管理 API にアクセスする場合は、テナントアカウント 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
```

StorageGRID 認証トークンが出力に表示されます。SSO を使用していない場合の API の使用方法と同様に、トークンを他の要求に使用できるようになりました。

3. cURL 要求を使用する場合は、次の手順を使用します。

a. サインインに必要な変数を宣言します。

```
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'
```



Grid Management API にアクセスするには、0 を「TENANTACCOUNTID」として使用します。

b. 署名付き認証 URL を受信するには、問題 A POST 要求を「/api/v3/authorize-saml」に送信し、応答から JSON エンコードを削除します。

次の例は 'TENANTACCOUNTID' の署名済み認証 URL に対する POST 要求を示しています結果は 'python-m JSOT' に渡され 'JSON エンコーディングが削除されます

```
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
```

この例の応答には、URL エンコードされた署名済み URL が含まれていますが、JSON エンコードされたレイヤは含まれていません。

```
{
  "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. 後続のコマンドで使用するために ' 応答から SAMLRequest を保存します

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

- d. AD FS からクライアント要求 ID を含む完全な URL を取得します。

1 つは、前の応答の URL を使用してログインフォームを要求する方法です。

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

応答にはクライアント要求 ID が含まれています。

```
<form method="post" id="loginForm" autocomplete="off"
novalidate="novalidate" onKeyPress="if (event && event.keyCode == 13)
Login.submitLoginRequest();" action="/adfs/ls/?
SAMLRequest=fZHRT0MwFIZfhh...UJikvo77sXPw%3D%3D&RelayState=12345&clie
nt-request-id=00000000-0000-0000-ee02-0080000000de" >
```

- e. 応答からクライアント要求 ID を保存します。

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

- f. 前の応答のフォームアクションにクレデンシャルを送信します。

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

AD FS からヘッダーに追加情報が含まれた 302 リダイレクトが返されます。



SSO システムで多要素認証 (MFA) が有効になっている場合、フォームポストには 2 つ目のパスワードまたはその他のクレデンシャルも含まれます。

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

- g. 応答から MSISAuth クッキーを保存します。

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

- h. 認証 POST からクッキーを使用して、指定した場所に GET 要求を送信します。

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

応答ヘッダーには、あとでログアウトに使用する AD FS セッション情報が含まれます。応答の本文には、非表示のフォームフィールドに SAMLResponse が含まれています。

```

HTTP/1.1 200 OK
Cache-Control: no-cache,no-store
Pragma: no-cache
Content-Length: 5665
Content-Type: text/html; charset=utf-8
Expires: -1
Server: Microsoft-HTTPAPI/2.0
P3P: ADFS doesn't have P3P policy, please contact your site's admin
for more details
Set-Cookie:
SamlSession=a3dpbnRlcnMtUHJpbWFyeS1BZG1pbi0xNzgmRmFsc2Umcng4NnJDZmFKV
XFxVWx3bk1lMnFuUSUzZCUzZCYmJiYmXzE3MjAyZTA5LTNmMDgtNDRkZC04Yzg5LTQ3ND
UxYzA3ZjkzYw==; path=/adfs; HttpOnly; Secure
Set-Cookie: MSISAuthenticated=MTEvNy8yMDE4IDQ6MzI6NTkgUE0=;
path=/adfs; HttpOnly; Secure
Set-Cookie: MSISLoopDetectionCookie=MjAxOC0xMS0wNzoxNjoxMjMjOjVpcMQ==;
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. 非表示フィールドから SAMLResponse を保存します

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

j. 保存した SAMLResponse を使用して、StorageGRID 認証トークンを生成する StorageGRID の「/api/saml-response` 要求」を作成します。

「RelayState」の場合はテナントアカウント ID を使用し、Grid 管理 API にサインインする場合は 0 を使用します。

```

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

```

応答には認証トークンが含まれています。

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

- a. 認証トークンを応答に「MYTOKEN」として保存します。

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

これで、SSO を使用していない場合の API の使用方法と同じように、他の要求に「MYTOKEN」を使用できます。

シングルサインオンが有効な場合は、**API** からサインアウトします

シングルサインオン（SSO）が有効になっている場合は、グリッド管理 API またはテナント管理 API からサインアウトするための一連の API 要求を問題 で処理する必要があります。ここで説明する手順は、Active Directory を SSO アイデンティティプロバイダとして使用する場合に該当します

このタスクについて

必要に応じて、組織のシングルログアウトページからログアウトするだけで、StorageGRID API からサインアウトできます。または、StorageGRID からシングルログアウト（SLO）を実行することもできます。この場合、有効な StorageGRID ベアラトークンが必要です。

手順

1. 署名されたログアウト要求を生成するには、「cookie" sso=true"」を SLO API に渡します。

```
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
```

ログアウト URL が返されます。

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

2. ログアウト URL を保存します。

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

3. 要求をログアウト URL に送信し、SLO を実行して StorageGRID にリダイレクトします。

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

302 応答が返されます。リダイレクト先は API のみのログアウトには適用されません。

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

4. StorageGRID Bearer トークンを削除します。

StorageGRID Bearer トークンを削除すると、SSO を使用しない場合と同じように動作します。「cookie」 sso=true' が指定されていない場合、ユーザーは SSO 状態に影響を与えることなく StorageGRID からログアウトされます。

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

「204 No Content」 応答は、ユーザがサインアウトしたことを示します。

シングルサインオンが有効な場合（Azure）は API を使用

ある場合 [シングルサインオン（SSO）の設定と有効化](#) また、Azure を SSO プロバイダとして使用している場合は、2つのサンプルスクリプトを使用して、グリッド管理 API またはテナント管理 API で有効な認証トークンを取得できます。

Azure シングルサインオンが有効な場合は、**API** にサインインします

以下の手順は、Azure を SSO アイデンティティプロバイダとして使用する場合に該当します

必要なもの

- StorageGRID ユーザグループに属するフェデレーテッドユーザの SSO E メールアドレスとパスワードが必要です。
- テナント管理 API にアクセスする場合は、テナントアカウント ID を確認しておきます。

このタスクについて

認証トークンを取得するには、次のサンプルスクリプトを使用します。

- 「storagegrid-ssoauth-caz.py」 Python スクリプト
- 「storagegrid-ssoauth-azure.js」 スクリプト

どちらのスクリプトも、StorageGRID インストールファイルディレクトリ (Red Hat Enterprise Linux または CentOS 用の場合は「./rpms」、Ubuntu または Debian 用の場合は「./debs」、VMware 用の「./vsphere」) にあります。

独自の API 統合を Azure に記述するには、「storagegrid-ssoauth-azure.py」スクリプトを参照してください。Python スクリプトは、StorageGRID に対して 2 つの要求を直接実行し（まず SAMLRequest を取得し、あとで認証トークンを取得するため）、さらに Node.js スクリプトを呼び出して、SSO 処理を実行します。

SSO 処理は一連の API 要求を使用して実行できますが、実行するのは簡単ではありません。puppeteer Node.js モジュールは、Azure SSO インターフェイスを破棄するために使用します。

URL エンコーディング問題を使用している場合は、「Unsupported SAML version」というエラーが表示される可能性があります。

手順

1. 必要な依存関係を次のようにインストールします。
 - a. Node.js をインストールします（を参照） ["https://nodejs.org/en/download/"](https://nodejs.org/en/download/)）。
 - b. 必要な Node.js モジュール（puppeteer および jsdom）を取り付けます。

```
'NPM install-g <module>'
```

2. Python スクリプトを Python インタープリタに渡して、スクリプトを実行します。

Python スクリプトは、対応する Node.js スクリプトを呼び出して、Azure SSO のインタラクションを実

行します。

3. プロンプトが表示されたら、次の引数の値を入力します（または、パラメータを使用して渡します）。
 - Azure へのサインインに使用する SSO E メールアドレス
 - StorageGRID のアドレス
 - テナント管理 API にアクセスする場合は、テナントアカウント ID
4. プロンプトが表示されたら、パスワードを入力し、要求された場合に Azure に対する MFA 認証を提供できるように準備します。

```
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'}
```



このスクリプトでは、MFA が Microsoft Authenticator を使用して実行されていることを前提として他の形式の MFA（テキストメッセージで受信したコードの入力など）をサポートするために、スクリプトの変更が必要になる場合があります。

StorageGRID 認証トークンが出力に表示されます。SSO を使用していない場合の API の使用方法と同様に、トークンを他の要求に使用できるようになりました。

シングルサインオンが有効な場合は **API** を使用（**PingFederate**）

ある場合 **シングルサインオン（SSO）の設定と有効化** また、SSO プロバイダとして PingFederate を使用するには、グリッド管理 API またはテナント管理 API で有効な認証トークンを取得するための一連の API 要求を問題 で処理する必要があります。

シングルサインオンが有効な場合は、**API** にサインインします

これらの手順は、SSO アイデンティティプロバイダとして PingFederate を使用している場合に適用されま

す

必要なもの

- StorageGRID ユーザグループに属するフェデレーテッドユーザの SSO ユーザ名とパスワードが必要です。
- テナント管理 API にアクセスする場合は、テナントアカウント ID を確認しておきます。

このタスクについて

認証トークンを取得するには、次のいずれかの例を使用します。

- StorageGRID インストールファイルディレクトリ (Red Hat Enterprise Linux または CentOS の場合は「`./rpms`」、Ubuntu または Debian の場合は「`./debs`」、VMware の場合は「`./vsphere-vsphere`」) にある「`storagegrid-ssoauth.py`」 p`python スクリプト。
- cURL 要求のワークフローの例。

cURL ワークフローは、実行に時間がかかりすぎるとタイムアウトする場合があります。「A valid SubjectConfirmation was not found on this Response」というエラーが表示される可能性があります。



cURL ワークフローの例では、パスワードが他のユーザに表示されないように保護されていません。

URL エンコーディング問題を使用している場合は、「Unsupported SAML version」というエラーが表示される可能性があります。

手順

1. 認証トークンを取得するには、次のいずれかの方法を選択します。
 - 「storagegrid -ssoauth.py」 Python スクリプトを使用します。手順 2 に進みます。
 - curl 要求を使用します。手順 3 に進みます。
2. 「storagegrid -ssoauth.py」スクリプトを使用する場合は、Python インタープリタにスクリプトを渡してスクリプトを実行します。

プロンプトが表示されたら、次の引数の値を入力します。

- SSO 方式。「PingFederate」（PingFederate、PingFederate など）の任意のバリエーションを入力できます。
- SSO ユーザ名
- StorageGRID がインストールされているドメイン。このフィールドは PingFederate には使用されません。空白のままにするか、任意の値を入力できます。
- StorageGRID のアドレス
- テナント管理 API にアクセスする場合は、テナントアカウント 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
```

StorageGRID 認証トークンが出力に表示されます。SSO を使用していない場合の API の使用方法と同様に、トークンを他の要求に使用できるようになりました。

3. cURL 要求を使用する場合は、次の手順を使用します。
 - a. サインインに必要な変数を宣言します。

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



Grid Management API にアクセスするには、0 を「TENANTACCOUNTID」として使
用します。

- b. 署名付き認証 URL を受信するには、問題 A POST 要求を「/api/v3/authorize-saml」に送信し、応答
から JSON エンコードを削除します。

次の例は、TENANTACCOUNTID の署名済み認証 URL を取得するための POST 要求です。結果は
python-m json ツールに渡され、JSON エンコードが削除されます。

```
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
```

この例の応答には、URL エンコードされた署名済み URL が含まれていますが、JSON エンコードさ
れたレイヤは含まれていません。

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

- c. 後続のコマンドで使用するために ' 応答から SAMLRequest を保存します

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

- d. 応答とクッキーをエクスポートし、応答をエコーします。

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

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

- e. 'pf.adapterID' 値をエクスポートし、応答をエコーします。

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

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

- f. 「href」 値をエクスポートし（末尾のスラッシュ / を削除）、応答をエコーします。

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

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

- g. 「action」 の値をエクスポートします。

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

- h. クレデンシャルとともに Cookie を送信する：

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

- i. 非表示フィールドから SAMLResponse を保存します

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

- j. 保存した SAMLResponse を使用して、StorageGRID 認証トークンを生成する StorageGRID の「/api/saml-response` 要求」を作成します。

「RelayState」の場合はテナントアカウント ID を使用し、Grid 管理 API にサインインする場合は 0 を使用します。

```
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
```

応答には認証トークンが含まれています。

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

- a. 認証トークンを応答に「MYTOKEN」として保存します。

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

これで、SSO を使用していない場合の API の使用方法と同じように、他の要求に「MYTOKEN」を使用できます。

シングルサインオンが有効な場合は、**API** からサインアウトします

シングルサインオン（SSO）が有効になっている場合は、グリッド管理 API またはテナント管理 API からサインアウトするための一連の API 要求を問題 で処理する必要があります。これらの手順は、SSO アイデンティティプロバイダとして PingFederate を使用している場合に適用されます

このタスクについて

必要に応じて、組織のシングルログアウトページからログアウトするだけで、StorageGRID API からサインアウトできます。または、StorageGRID からシングルログアウト（SLO）を実行することもできます。この場合、有効な StorageGRID ベアラトークンが必要です。

手順

1. 署名されたログアウト要求を生成するには、「cookie" sso=true"」を SLO API に渡します。

```
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
```

ログアウト URL が返されます。

```
{
  "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. ログアウト URL を保存します。

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

3. 要求をログアウト URL に送信し、SLO を実行して StorageGRID にリダイレクトします。

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

302 応答が返されます。リダイレクト先は API のみのログアウトには適用されません。

```
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. StorageGRID Bearer トークンを削除します。

StorageGRID Bearer トークンを削除すると、SSO を使用しない場合と同じように動作します。「cookie」sso=true' が指定されていない場合、ユーザーは SSO 状態に影響を与えることなく StorageGRID からログアウトされます。

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

「204 No Content」応答は、ユーザがサインアウトしたことを示します。

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


著作権に関する情報

Copyright © 2025 NetApp, Inc. All Rights Reserved. Printed in the U.S.このドキュメントは著作権によって保護されています。著作権所有者の書面による事前承諾がある場合を除き、画像媒体、電子媒体、および写真複写、記録媒体、テープ媒体、電子検索システムへの組み込みを含む機械媒体など、いかなる形式および方法による複製も禁止します。

ネットアップの著作物から派生したソフトウェアは、次に示す使用許諾条項および免責条項の対象となります。

このソフトウェアは、ネットアップによって「現状のまま」提供されています。ネットアップは明示的な保証、または商品性および特定目的に対する適合性の暗示的保証を含み、かつこれに限定されないいかなる暗示的な保証も行いません。ネットアップは、代替品または代替サービスの調達、使用不能、データ損失、利益損失、業務中断を含み、かつこれに限定されない、このソフトウェアの使用により生じたすべての直接的損害、間接的損害、偶発的損害、特別損害、懲罰的損害、必然的損害の発生に対して、損失の発生の可能性が通知されていたとしても、その発生理由、根拠とする責任論、契約の有無、厳格責任、不法行為（過失またはそうでない場合を含む）にかかわらず、一切の責任を負いません。

ネットアップは、ここに記載されているすべての製品に対する変更を随時、予告なく行う権利を保有します。ネットアップによる明示的な書面による合意がある場合を除き、ここに記載されている製品の使用により生じる責任および義務に対して、ネットアップは責任を負いません。この製品の使用または購入は、ネットアップの特許権、商標権、または他の知的所有権に基づくライセンスの供与とはみなされません。

このマニュアルに記載されている製品は、1つ以上の米国特許、その他の国の特許、および出願中の特許によって保護されている場合があります。

権利の制限について：政府による使用、複製、開示は、DFARS 252.227-7013（2014年2月）およびFAR 5252.227-19（2007年12月）のRights in Technical Data -Noncommercial Items（技術データ - 非商用品目に関する諸権利）条項の(b)(3)項、に規定された制限が適用されます。

本書に含まれるデータは商用製品および/または商用サービス（FAR 2.101の定義に基づく）に関係し、データの所有権はNetApp, Inc.にあります。本契約に基づき提供されるすべてのネットアップの技術データおよびコンピュータソフトウェアは、商用目的であり、私費のみで開発されたものです。米国政府は本データに対し、非独占的かつ移転およびサブライセンス不可で、全世界を対象とする取り消し不能の制限付き使用权を有し、本データの提供の根拠となった米国政府契約に関連し、当該契約の裏付けとする場合にのみ本データを使用できます。前述の場合を除き、NetApp, Inc.の書面による許可を事前に得ることなく、本データを使用、開示、転載、改変するほか、上演または展示することはできません。国防総省にかかる米国政府のデータ使用权については、DFARS 252.227-7015(b)項（2014年2月）で定められた権利のみが認められます。

商標に関する情報

NetApp、NetAppのロゴ、<http://www.netapp.com/TM>に記載されているマークは、NetApp, Inc.の商標です。その他の会社名と製品名は、それを所有する各社の商標である場合があります。