API를 사용합니다

StorageGRID

NetApp
April 10, 2024

목차

API를 사용합니다

Grid Management API를 사용합니다

Grid Manager 사용자 인터페이스 대신 Grid Management REST API를 사용하여 시스템 관리 작업을 수행할 수 있습니다. 예를 들어, API를 사용하여 작업을 자동화하거나 사용자와 같은 여러 엔터티를 더 빠르게 생성할 수 있습니다.

최고 수준의 리소스

Grid Management API는 다음과 같은 최상위 리소스를 제공합니다.

- '/grid': 그리드 관리자 사용자로 액세스가 제한되며 구성된 그룹 권한에 따라 달라집니다.
- '/org': 테넌트 계정의 로컬 또는 통합 LDAP 그룹에 속한 사용자로 액세스가 제한됩니다. 자세한 내용은 을 참조하십시오 테넌트 계정을 사용합니다.
- '/private': 액세스 권한은 Grid Manager 사용자로 제한되며 구성된 그룹 권한에 따라 결정됩니다. 사설 API는 사전 통보 없이 변경될 수 있습니다. StorageGRID 전용 엔드포인트도 요청의 API 버전을 무시합니다.

API 요청을 발행합니다

Grid Management API는 Swagger 오픈 소스 API 플랫폼을 사용합니다. Swagger는 개발자와 개발자가 아닌 사용자가 API를 사용하여 StorageGRID에서 실시간 작업을 수행할 수 있도록 직관적인 사용자 인터페이스를 제공합니다.

Swagger 사용자 인터페이스는 각 API 작동에 대한 전체 세부 정보와 문서를 제공합니다.

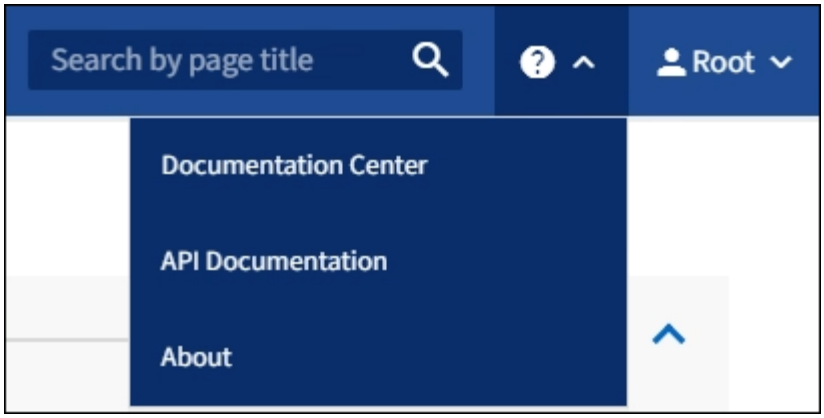
필요한 것

- 를 사용하여 그리드 관리자에 로그인했습니다 지원되는 웹 브라우저.
- 특정 액세스 권한이 있습니다.

> API Docs 웹 페이지를 사용하여 수행하는 모든 API 작업은 라이브 작업입니다. 실수로 구성 데이터나 기타 데이터를 작성, 업데이트 또는 삭제하지 않도록 주의하십시오.

단계

1. Grid Manager 헤더에서 도움말 아이콘을 선택하고 * API Documentation * 을 선택합니다.



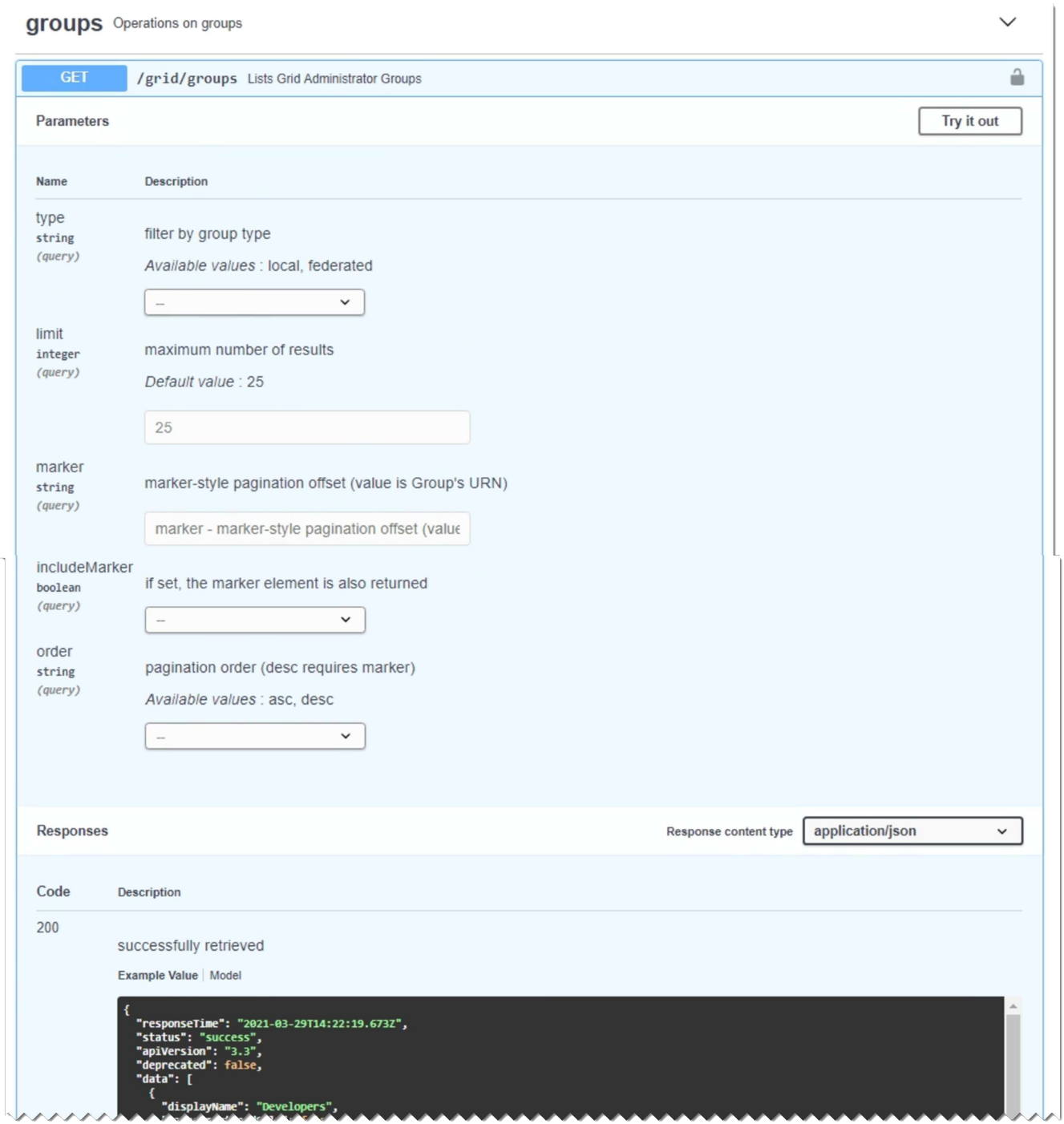
2. 전용 API로 작업을 수행하려면 StorageGRID 관리 API 페이지에서 * 전용 API 설명서 * 로 이동 * 을 선택합니다.

   사설 API는 사전 통보 없이 변경될 수 있습니다. StorageGRID 전용 엔드포인트도 요청의 API 버전을 무시합니다.

3. 원하는 작업을 선택합니다.

   API 작업을 확장하면 가져오기, 가져오기, 업데이트 및 삭제와 같은 사용 가능한 HTTP 작업을 볼 수 있습니다.

4. 끝점 URL, 필수 또는 선택적 매개 변수 목록, 요청 본문(필요한 경우) 예제 및 가능한 응답을 비롯한 요청 세부 정보를 보려면 HTTP 작업을 선택합니다.



5. 요청에 그룹 또는 사용자 ID와 같은 추가 매개 변수가 필요한지 확인합니다. 그런 다음 이 값을 구합니다. 필요한 정보를 얻기 위해 먼저 다른 API 요청을 발급해야 할 수도 있습니다.
6. 예제 요청 본문을 수정해야 하는지 확인합니다. 이 경우 * Model * 을 선택하여 각 필드의 요구 사항을 확인할 수 있습니다.
7. 체험하기 * 를 선택합니다.
8. 필요한 매개 변수를 제공하거나 요청 본문을 필요에 따라 수정합니다.
9. Execute * 를 선택합니다.
10. 응답 코드를 검토하여 요청이 성공했는지 확인합니다.

Grid Management API 작업

Grid Management API는 사용 가능한 작업을 다음 섹션으로 구성합니다.

> 이 목록에는 공용 API에서 사용할 수 있는 작업만 포함됩니다.

- * ACCOUNT * — 새 계정 생성 및 지정된 계정의 스토리지 사용량 검색을 포함하여 스토리지 테넌트 계정을 관리하는 작업입니다.
- * ALARMS * — 현재 경보(레거시 시스템)를 나열하고, 현재 경고와 노드 연결 상태 요약을 포함하여 그리드의 상태에 대한 정보를 반환하는 작업.
- * alert-history * — 해결된 경고에 대한 작업.
- 알림 메시지 수신자 * — 경고 알림 수신자(이메일)에 대한 작업.
- * alert-rules * — 경고 규칙에 대한 작업.
- * alert-silences * — 경고 작동 중.
- * 경고 * — 경고 작업.
- * 감사 * — 감사 구성을 나열하고 업데이트하는 작업.
- * auth * — 사용자 세션 인증을 수행하기 위한 작업.

  Grid Management API는 Bearer Token Authentication Scheme을 지원한다. 로그인하려면 인증 요청의 JSON 본문('POST/API/v3/authorize')에 사용자 이름과 암호를 입력합니다. 사용자가 성공적으로 인증되면 보안 토큰이 반환됩니다. 이 토큰은 후속 API 요청 헤더("Authorization:Bearer_token_")에 제공되어야 합니다.

  > StorageGRID 시스템에 대해 Single Sign-On이 활성화된 경우 인증을 위해 다른 단계를 수행해야 합니다. SSO(Single Sign-On)가 활성화된 경우 API에 대한 인증"을 참조하십시오.

  인증 보안 개선에 대한 자세한 내용은 사이트 간 요청 위조 방지 를 참조하십시오.

- * client-certificates * — 외부 모니터링 도구를 사용하여 StorageGRID에 안전하게 액세스할 수 있도록 클라이언트 인증서를 구성하는 작업.
- * config * — 그리드 관리 API 제품 릴리스 및 버전과 관련된 작업. 제품 릴리스 버전과 해당 릴리스에서 지원하는 Grid Management API의 주요 버전을 나열할 수 있으며 더 이상 사용되지 않는 API 버전을 사용하지 않도록 설정할 수 있습니다.
- * deactivated - features * — 비활성화된 기능을 보기 위한 작업.
- * DNS-서버 * — 구성된 외부 DNS 서버를 나열하고 변경하는 작업.
- **endpoint-domain-names** — 끝점 도메인 이름을 나열하고 변경하는 작업.
- * 삭제 코딩 * — 삭제 코딩 프로파일에서 작업.
- * 확장 * — 확장 작업(절차 수준).
- * expansion-nodes * — 확장 시 작업(노드 레벨).
- * 확장 사이트 * — 확장 시 운영(사이트 레벨)
- * GRID-NETWORKS * — 그리드 네트워크 목록을 나열하고 변경하는 작업.
- * GRID-Passwords * — 그리드 암호 관리 작업.

- * 그룹 * — 로컬 그리드 관리자 그룹을 관리하고 외부 LDAP 서버에서 통합 그리드 관리자 그룹을 검색하는 작업.
- * identity-source * — 외부 ID 소스를 구성하고 통합 그룹 및 사용자 정보를 수동으로 동기화하는 작업.
- * ILM * — 정보 수명 주기 관리(ILM)에 대한 운영.
- * license * — StorageGRID 라이센스를 검색하고 업데이트하는 작업.
- * 로그 * — 로그 파일을 수집하고 다운로드하기 위한 작업.
- * 메트릭 * — 일정 기간 동안 단일 시점 및 범위 메트릭 쿼리에 대한 즉석 메트릭 쿼리를 비롯한 StorageGRID 메트릭의 운영 Grid Management API는 Prometheus 시스템 모니터링 도구를 백엔드 데이터 소스로 사용합니다. Prometheus 쿼리 구성에 대한 자세한 내용은 Prometheus 웹 사이트를 참조하십시오.

> 이름에 '*private*'가 포함된 메트릭은 내부용으로만 사용됩니다. 이러한 메트릭은 사전 통지 없이 StorageGRID 릴리스 간에 변경될 수 있습니다.

- * node-details * — 노드 세부 정보에 대한 작업.
- * 노드 상태 * — 노드 상태에 대한 작업
- * NTP-서버 * — 외부 NTP(Network Time Protocol) 서버를 나열하거나 업데이트하는 작업.
- * 오브젝트 * — 오브젝트 및 오브젝트 메타데이터 작업
- * 복구 * — 복구 절차를 위한 작업.
- * recovery-package * — 복구 패키지를 다운로드하기 위한 작업.
- * 지역 * — 영역을 보고 작성하는 작업.
- * S3-오브젝트 잠금 * — 글로벌 S3 오브젝트 잠금 설정에서 작업.
- * server-certificate * — Grid Manager 서버 인증서를 보고 업데이트하는 작업.
- * SNMP * — 현재 SNMP 구성에 대한 작업.
- * traffic-classes * — 트래픽 분류 정책을 위한 운영.
- * 신뢰할 수 없는 클라이언트-네트워크 * — 신뢰할 수 없는 클라이언트 네트워크 구성에서의 작업.
- * 사용자 * — 그리드 관리자 사용자를 보고 관리하는 작업.

Grid Management API 버전 관리

Grid Management API는 버전 관리를 사용하여 무중단 업그레이드를 지원합니다.

예를 들어 이 요청 URL은 API의 버전 3을 지정합니다.

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

테넌트 관리 API의 주요 버전은 이전 버전과 * _호환되지 않는 _ * 변경 사항이 있을 때 충돌합니다. 테넌트 관리 API의 부 버전은 * _ 이(가) 이전 버전과 호환된다는 변경 사항이 있을 때 충돌합니다. 호환 가능한 변경 사항에는 새 끝점 또는 새 속성 추가가 포함됩니다. 다음 예제에서는 변경 유형에 따라 API 버전을 충돌하는 방법을 보여 줍니다.

API 변경 유형입니다	이전 버전	새 버전
이전 버전과 호환 가능합니다	2.1	2.2

API 변경 유형입니다	이전 버전	새 버전
이전 버전과 호환되지 않습니다	2.1	3.0

StorageGRID 소프트웨어를 처음 설치하는 경우 가장 최신 버전의 그리드 관리 API만 활성화됩니다. 그러나 StorageGRID의 새 기능 릴리즈로 업그레이드하면 하나 이상의 StorageGRID 기능 릴리즈에 대한 이전 API 버전에 계속 액세스할 수 있습니다.

> Grid Management API를 사용하여 지원되는 버전을 구성할 수 있습니다. 자세한 내용은 Swagger API 설명서의 ""구성"" 섹션을 참조하십시오. 최신 버전을 사용하도록 모든 Grid Management API 클라이언트를 업데이트한 후에는 이전 버전에 대한 지원을 비활성화해야 합니다.

오래된 요청은 다음과 같은 방법으로 더 이상 사용되지 않는 것으로 표시됩니다.

- 응답 헤더가 "DEPRECATED:TRUE"입니다.
- JSON 응답 본문에는 "DEPRECATED"가 포함됩니다. TRUE
- 더 이상 사용되지 않는 경고가 NMS.log에 추가됩니다. 예를 들면 다음과 같습니다.

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

현재 릴리즈에서 지원되는 API 버전을 확인합니다

다음 API 요청을 사용하여 지원되는 API 주요 버전 목록을 반환합니다.

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

요청에 대한 API 버전을 지정합니다

PATH 파라미터('/api/v3')나 header('api-Version:3')를 이용하여 API 버전을 지정할 수 있다. 두 값을 모두 제공하면 헤더 값이 경로 값을 재정의합니다.

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

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

사이트 간 요청 위조(CSRF)로부터 보호

CSRF 토큰을 사용하여 쿠키를 사용하는 인증을 강화하면 StorageGRID에 대한 CSRF(사이트 간 요청 위조) 공격으로부터 보호할 수 있습니다. Grid Manager 및 Tenant Manager는 이 보안 기능을 자동으로 활성화합니다. 다른 API 클라이언트는 로그인할 때 활성화 여부를 선택할 수 있습니다.

HTTP 양식 POST와 같이 다른 사이트에 대한 요청을 트리거할 수 있는 공격자는 로그인한 사용자의 쿠키를 사용하여 특정 요청을 만들 수 있습니다.

StorageGRID는 CSRF 토큰을 사용하여 CSRF 공격으로부터 보호합니다. 활성화된 경우 특정 쿠키의 내용은 특정 헤더 또는 특정 POST 본문 매개 변수의 내용과 일치해야 합니다.

이 기능을 활성화하려면 인증 중에 csrfToken 매개 변수를 true로 설정하십시오. 기본값은 false 입니다.

```
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"
```

true이면 Grid Manager에 로그인할 때 임의의 값으로 GridCsrfToken 쿠키가 설정되고 테넌트 관리자에 로그인할 때 임의의 값으로 AccountCsrfToken 쿠키가 설정됩니다.

쿠키가 있는 경우 시스템 상태(POST, PUT, 패치, 삭제)를 수정할 수 있는 모든 요청에 다음 중 하나가 포함되어야 합니다.

- CSRF 토큰 쿠키의 값으로 설정된 헤더의 X-CSRF-Token 헤더입니다.
- 폼 인코딩된 본문을 허용하는 끝점의 경우 "csrfToken" 형식 인코딩된 요청 본문 매개 변수입니다.

추가 예제 및 세부 정보는 온라인 API 설명서를 참조하십시오.

> CSRF 토큰 쿠키 세트를 가진 요청은 또한 JSON 요청 본문을 CSRF 공격에 대한 추가 보호로서 기대하는 모든 요청에 대해 ""Content-Type:application/json"" 헤더를 적용합니다.

SSO(Single Sign-On)가 활성화된 경우 API를 사용합니다

SSO(Single Sign-On)가 활성화된 경우 API 사용(Active Directory)

있는 경우 SSO(Single Sign-On) 구성 및 활성화 Active Directory를 SSO 공급자로 사용하는 경우, 그리드 관리 API 또는 테넌트 관리 API에 유효한 인증 토큰을 얻기 위해 일련의 API 요청을 실행해야 합니다.

SSO(Single Sign-On)가 활성화된 경우 API에 로그인합니다

Active Directory를 SSO ID 공급자로 사용하는 경우 다음 지침이 적용됩니다.

필요한 것

- StorageGRID 사용자 그룹에 속한 페더레이션 사용자의 SSO 사용자 이름과 암호를 알고 있습니다.
- 테넌트 관리 API에 액세스하려면 테넌트 계정 ID를 알고 있어야 합니다.

이 작업에 대해

인증 토큰을 얻으려면 다음 예 중 하나를 사용할 수 있습니다.

- StorageGRID 설치 파일 디렉토리(Red Hat Enterprise Linux 또는 CentOS의 경우 ./rpms, Ubuntu 또는 Debian의 경우 ./debs, VMware의 경우 ./vsphere)에 있는 toragegrid-ssoauth.py Python 스크립트입니다.
- curl 요청의 워크플로 예

  컬을 너무 느리게 수행하면 컬링 작업 시간이 초과될 수 있습니다. "이 응답에 유효한 SubjectConfirmation을 찾을 수 없습니다."라는 오류가 표시될 수 있습니다.

  > 예제 curl 워크플로는 다른 사용자가 암호를 볼 수 없도록 보호하지 않습니다.

URL 인코딩 문제가 있는 경우 '지원되지 않는 SAML 버전' 오류가 표시될 수 있습니다.

단계

1. 인증 토큰을 얻으려면 다음 방법 중 하나를 선택합니다.
   - toragegrid-soauth.py Python 스크립트를 사용하십시오. 2단계로 이동합니다.
   - curl 요청을 사용합니다. 3단계로 이동합니다.
2. 'toragegrid-ssoauth.py' 스크립트를 사용하려면 스크립트를 Python 해석기로 전달하고 스크립트를 실행합니다.

   프롬프트가 표시되면 다음 인수에 대한 값을 입력합니다.

   - SSO 방법 ADFS 또는 ADFS를 입력합니다.
   - SSO 사용자 이름입니다
   - StorageGRID가 설치된 도메인입니다
   - StorageGRID의 주소입니다
   - 테넌트 관리 API에 액세스하려는 경우 테넌트 계정 ID입니다.

```
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 인증 토큰은 출력에 제공됩니다. 이제 SSO가 사용되지 않는 경우 API를 사용하는 방법과 유사하게 다른 요청에 토큰을 사용할 수 있습니다.

3. curl 요청을 사용하려면 다음 절차를 따르십시오.

   a. 로그인에 필요한 변수를 선언합니다.

```
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 API에 액세스하려면 0을 "TENANTACCOUNTID"로 사용합니다.

   b. 서명된 인증 URL을 받으려면 '/api/v3/authorize-SAML'에 POST 요청을 보내고 응답에서 추가 JSON 인코딩을 제거합니다.

   이 예제에서는 "TENANTACCOUNTID"에 대한 서명된 인증 URL에 대한 POST 요청을 보여 줍니다. 결과는 JSON 인코딩을 제거하기 위해 python-m json.tool으로 전달됩니다.

```
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
```

   이 예제의 응답에는 URL로 인코딩된 서명된 URL이 포함되어 있지만 추가 JSON 인코딩 계층은 포함되지 않습니다.

```
{
    "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. 후속 명령어에 사용하기 위해 응답에서 'AMLRequest'를 저장한다.

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

d. AD FS에서 클라이언트 요청 ID가 포함된 전체 URL을 가져옵니다.

한 가지 옵션은 이전 응답의 URL을 사용하여 로그인 양식을 요청하는 것입니다.

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

응답에는 클라이언트 요청 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. 응답에서 클라이언트 요청 ID를 저장합니다.

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

f. 이전 응답에서 양식 작업으로 자격 증명을 보냅니다.

```
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 FS는 헤더에 추가 정보가 포함된 302 리디렉션을 반환합니다.

> SSO 시스템에 대해 MFA(다중 요소 인증)가 활성화된 경우 양식 게시물에는 두 번째 암호 또는 다른 자격 증명도 포함됩니다.

```
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. 응답에서 MISAuth 쿠키를 저장합니다.

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

h. 인증 POST에서 쿠키를 사용하여 지정된 위치로 GET 요청을 보냅니다.

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

응답 헤더에는 나중에 로그아웃 사용을 위한 AD FS 세션 정보가 포함되며 응답 본문에는 숨겨진 양식 필드에 SALMLResponse가 포함됩니다.

```
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. 숨겨진 필드에서 '응답'을 저장합니다.

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

j. 저장된 'SAMLResponse'를 사용하여 StorageGRID 인증 토큰을 생성하기 위한 StorageGRID '/API/SAML-RESPONSE' 요청을 생성합니다.

RelayState의 경우, Grid Management API에 로그인하려면 테넌트 계정 ID를 사용하거나 0을 사용하십시오.

```
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
```

응답에는 인증 토큰이 포함됩니다.

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

a. 응답에 인증 토큰을 MYTOKEN으로 저장합니다.

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

이제 다른 요청에는 MYTOKEN을 사용할 수 있습니다. SSO를 사용하지 않을 경우 API를 사용하는 방법과 비슷합니다.

SSO(Single Sign-On)가 활성화된 경우 **API**에서 로그아웃합니다

SSO(Single Sign-On)가 활성화된 경우 그리드 관리 API 또는 테넌트 관리 API에서 로그아웃하기 위해 일련의 API 요청을 실행해야 합니다. Active Directory를 SSO ID 공급자로 사용하는 경우 다음 지침이 적용됩니다

이 작업에 대해

필요한 경우 조직의 단일 로그아웃 페이지에서 로그아웃하기만 하면 StorageGRID API에서 로그아웃할 수 있습니다. 또는 StorageGRID에서 유효한 StorageGRID 베어러 토큰이 필요한 단일 로그아웃(SLO)을 트리거할 수 있습니다.

단계

1. 서명된 로그아웃 요청을 생성하려면 SLO API에 쿠키 "SSO=true""를 전달합니다.

```
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
```

로그아웃 URL이 반환됩니다.

```
{
    "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. 로그아웃 URL을 저장합니다.

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

3. 로그아웃 URL에 요청을 보내 SLO를 트리거하고 StorageGRID로 다시 리디렉션합니다.

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

302 응답이 반환됩니다. 리디렉션 위치는 API 전용 로그아웃에는 적용되지 않습니다.

```
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. StorageGRID bearer token을 삭제한다.

StorageGRID 베어러 토큰을 삭제하는 것은 SSO를 사용하지 않는 것과 동일한 방식으로 작동합니다. cookie "sso=true"'를 제공하지 않으면 SSO 상태에 영향을 주지 않고 StorageGRID에서 로그아웃됩니다.

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

204 콘텐츠 없음 응답은 사용자가 로그아웃되었음을 나타냅니다.

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

SSO(Single Sign-On)가 활성화된 경우 API 사용(Azure)

있는 경우 SSO(Single Sign-On) 구성 및 활성화 Azure를 SSO 공급자로 사용하는 경우, 두 개의 예제 스크립트를 사용하여 Grid Management API 또는 Tenant Management API에 유효한 인증 토큰을 얻을 수 있습니다.

Azure Single Sign-On이 활성화된 경우 API에 로그인합니다

Azure를 SSO ID 공급자로 사용하는 경우 다음 지침이 적용됩니다

필요한 것

- StorageGRID 사용자 그룹에 속한 페더레이션 사용자의 SSO 전자 메일 주소와 암호를 알고 있습니다.
- 테넌트 관리 API에 액세스하려면 테넌트 계정 ID를 알고 있어야 합니다.

이 작업에 대해

인증 토큰을 얻으려면 다음 예제 스크립트를 사용할 수 있습니다.

- 토라게르흐산스소auth-soauth-so.py 파이썬 스크립트
- toragegrid-soauth-Azure.js의 Node.js 스크립트

두 스크립트는 모두 StorageGRID 설치 파일 디렉토리(Red Hat Enterprise Linux 또는 CentOS의 경우 ./rpms, Ubuntu 또는 Debian의 경우 ./debs, VMware의 경우 ./vsphere)에 있습니다.

Azure와의 API 통합 기능을 직접 작성하려면 'toragegrid-soauth-Azure.py' 스크립트를 참조하십시오. Python 스크립트는 StorageGRID에 직접 두 개의 요청을 하고(먼저 SAMLRequest를 받고 나중에 인증 토큰을 얻기 위해) Node.js 스크립트를 호출하여 Azure와 상호 작용하여 SSO 작업을 수행합니다.

SSO 작업은 일련의 API 요청을 사용하여 실행할 수 있지만, 그렇게 하는 것은 간단하지 않습니다. Puppeteer Node.js 모듈은 Azure SSO 인터페이스를 스크레핑하는 데 사용됩니다.

URL 인코딩 문제가 있는 경우 '지원되지 않는 SAML 버전' 오류가 표시될 수 있습니다.

단계

1. 다음과 같이 필요한 종속성을 설치합니다.
   a. Node.js를 설치합니다(참조) "https://nodejs.org/en/download/")를 클릭합니다.
   b. 필요한 Node.js 모듈(puppeteer 및 jsdom)을 설치합니다.

      "NPM INSTALL-g<MODULE>'

2. Python 스크립트를 Python 인터프리터로 전달하여 스크립트를 실행합니다.

   그런 다음 Python 스크립트는 해당 Node.js 스크립트를 호출하여 Azure SSO 상호 작용을 수행합니다.

3. 프롬프트가 표시되면 다음 인수에 대한 값을 입력하거나 매개 변수를 사용하여 전달합니다.
   - Azure에 로그인하는 데 사용되는 SSO 이메일 주소입니다
   - StorageGRID의 주소입니다
   - 테넌트 관리 API에 액세스하려는 경우 테넌트 계정 ID입니다
4. 메시지가 표시되면 암호를 입력하고 요청 시 Azure에 MFA 권한을 제공할 준비를 합니다.

```
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'}
```

> 이 스크립트는 MFA가 Microsoft Authenticator를 사용하여 수행된 것으로 가정합니다. 텍스트 메시지를 통해 받은 코드를 입력하는 등 다른 형태의 MFA를 지원하도록 스크립트를 수정해야 할 수도 있습니다.

StorageGRID 인증 토큰은 출력에 제공됩니다. 이제 SSO가 사용되지 않는 경우 API를 사용하는 방법과 유사하게 다른 요청에 토큰을 사용할 수 있습니다.

SSO(Single Sign-On)가 활성화된 경우 API 사용(PingFederate)

있는 경우 SSO(Single Sign-On) 구성 및 활성화 그리고 PingFederate를 SSO 공급자로 사용하는 경우 일련의 API 요청을 발급하여 Grid Management API 또는 Tenant Management API에 유효한 인증 토큰을 얻어야 합니다.

SSO(Single Sign-On)가 활성화된 경우 API에 로그인합니다

이 지침은 PingFederate를 SSO ID 공급자로 사용하는 경우 적용됩니다

필요한 것

- StorageGRID 사용자 그룹에 속한 페더레이션 사용자의 SSO 사용자 이름과 암호를 알고 있습니다.
- 테넌트 관리 API에 액세스하려면 테넌트 계정 ID를 알고 있어야 합니다.

이 작업에 대해

인증 토큰을 얻으려면 다음 예 중 하나를 사용할 수 있습니다.

- StorageGRID 설치 파일 디렉토리(Red Hat Enterprise Linux 또는 CentOS의 경우 ./rpms, Ubuntu 또는 Debian의 경우 ./debs, VMware의 경우 ./vsphere)에 있는 toragegrid-ssoauth.py Python 스크립트입니다.
- curl 요청의 워크플로 예

  컬을 너무 느리게 수행하면 컬링 작업 시간이 초과될 수 있습니다. "이 응답에 유효한 SubjectConfirmation을 찾을 수 없습니다."라는 오류가 표시될 수 있습니다.

  > 예제 curl 워크플로는 다른 사용자가 암호를 볼 수 없도록 보호하지 않습니다.

  URL 인코딩 문제가 있는 경우 '지원되지 않는 SAML 버전' 오류가 표시될 수 있습니다.

단계

1. 인증 토큰을 얻으려면 다음 방법 중 하나를 선택합니다.
   - toragegrid-soauth.py Python 스크립트를 사용하십시오. 2단계로 이동합니다.
   - curl 요청을 사용합니다. 3단계로 이동합니다.
2. 'toragegrid-ssoauth.py' 스크립트를 사용하려면 스크립트를 Python 해석기로 전달하고 스크립트를 실행합니다.

   프롬프트가 표시되면 다음 인수에 대한 값을 입력합니다.

   - SSO 방법 ""핑남부연합"(PINGFEDERATE, 핑남부연합 등)의 모든 변형을 입력할 수 있습니다.
   - SSO 사용자 이름입니다

- StorageGRID가 설치된 도메인입니다. 이 필드는 PingFederate에 사용되지 않습니다. 빈 칸으로 두거나 원하는 값을 입력할 수 있습니다.
- StorageGRID의 주소입니다
- 테넌트 관리 API에 액세스하려는 경우 테넌트 계정 ID입니다.

```
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 인증 토큰은 출력에 제공됩니다. 이제 SSO가 사용되지 않는 경우 API를 사용하는 방법과 유사하게 다른 요청에 토큰을 사용할 수 있습니다.

3. curl 요청을 사용하려면 다음 절차를 따르십시오.

   a. 로그인에 필요한 변수를 선언합니다.

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

> Grid Management API에 액세스하려면 0을 "TENANTACCOUNTID"로 사용합니다.

   b. 서명된 인증 URL을 받으려면 '/api/v3/authorize-SAML'에 POST 요청을 보내고 응답에서 추가 JSON 인코딩을 제거합니다.

   이 예제에서는 TENANTACCOUNTID에 대한 서명된 인증 URL에 대한 POST 요청을 보여 줍니다. 결과는 python-m json.tool에 전달되어 JSON 인코딩을 제거합니다.

```
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
```

   이 예제의 응답에는 URL로 인코딩된 서명된 URL이 포함되어 있지만 추가 JSON 인코딩 계층은 포함되지 않습니다.

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

c. 후속 명령어에 사용하기 위해 응답에서 'AMLRequest'를 저장한다.

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

d. 응답과 쿠키를 내보내고 응답을 에코합니다.

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

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

e. 'pf.adapterId' 값을 내보내고 응답을 에코합니다.

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

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

f. 'href' 값을 내보내고(후행 슬래시/ 제거) 응답을 에코합니다.

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

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

g. '조치' 값 내보내기:

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

h. 자격 증명과 함께 쿠키 보내기:

```
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. 숨겨진 필드에서 '응답'을 저장합니다.

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

j. 저장된 'SAMLResponse'를 사용하여 StorageGRID 인증 토큰을 생성하기 위한 StorageGRID '/API/SAML-RESPONSE' 요청을 생성합니다.

RelayState의 경우, Grid Management API에 로그인하려면 테넌트 계정 ID를 사용하거나 0을 사용하십시오.

```
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
```

응답에는 인증 토큰이 포함됩니다.

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

a. 응답에 인증 토큰을 MYTOKEN으로 저장합니다.

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

이제 다른 요청에는 MYTOKEN을 사용할 수 있습니다. SSO를 사용하지 않을 경우 API를 사용하는 방법과 비슷합니다.

SSO(Single Sign-On)가 활성화된 경우 **API**에서 로그아웃합니다

SSO(Single Sign-On)가 활성화된 경우 그리드 관리 API 또는 테넌트 관리 API에서 로그아웃하기 위해 일련의 API 요청을 실행해야 합니다. 이 지침은 PingFederate를 SSO ID 공급자로 사용하는 경우 적용됩니다

이 작업에 대해

필요한 경우 조직의 단일 로그아웃 페이지에서 로그아웃하기만 하면 StorageGRID API에서 로그아웃할 수 있습니다. 또는 StorageGRID에서 유효한 StorageGRID 베어러 토큰이 필요한 단일 로그아웃(SLO)을 트리거할 수 있습니다.

단계

1. 서명된 로그아웃 요청을 생성하려면 SLO API에 쿠키 "SSO=true""를 전달합니다.

```
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
```

로그아웃 URL이 반환됩니다.

```
{
    "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. 로그아웃 URL을 저장합니다.

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

3. 로그아웃 URL에 요청을 보내 SLO를 트리거하고 StorageGRID로 다시 리디렉션합니다.

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

302 응답이 반환됩니다. 리디렉션 위치는 API 전용 로그아웃에는 적용되지 않습니다.

```
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 token을 삭제한다.

StorageGRID 베어러 토큰을 삭제하는 것은 SSO를 사용하지 않는 것과 동일한 방식으로 작동합니다. cookie "sso=true"'를 제공하지 않으면 SSO 상태에 영향을 주지 않고 StorageGRID에서 로그아웃됩니다.

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

204 콘텐츠 없음 응답은 사용자가 로그아웃되었음을 나타냅니다.

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