OAuth を使用した REST API クライアントの認証

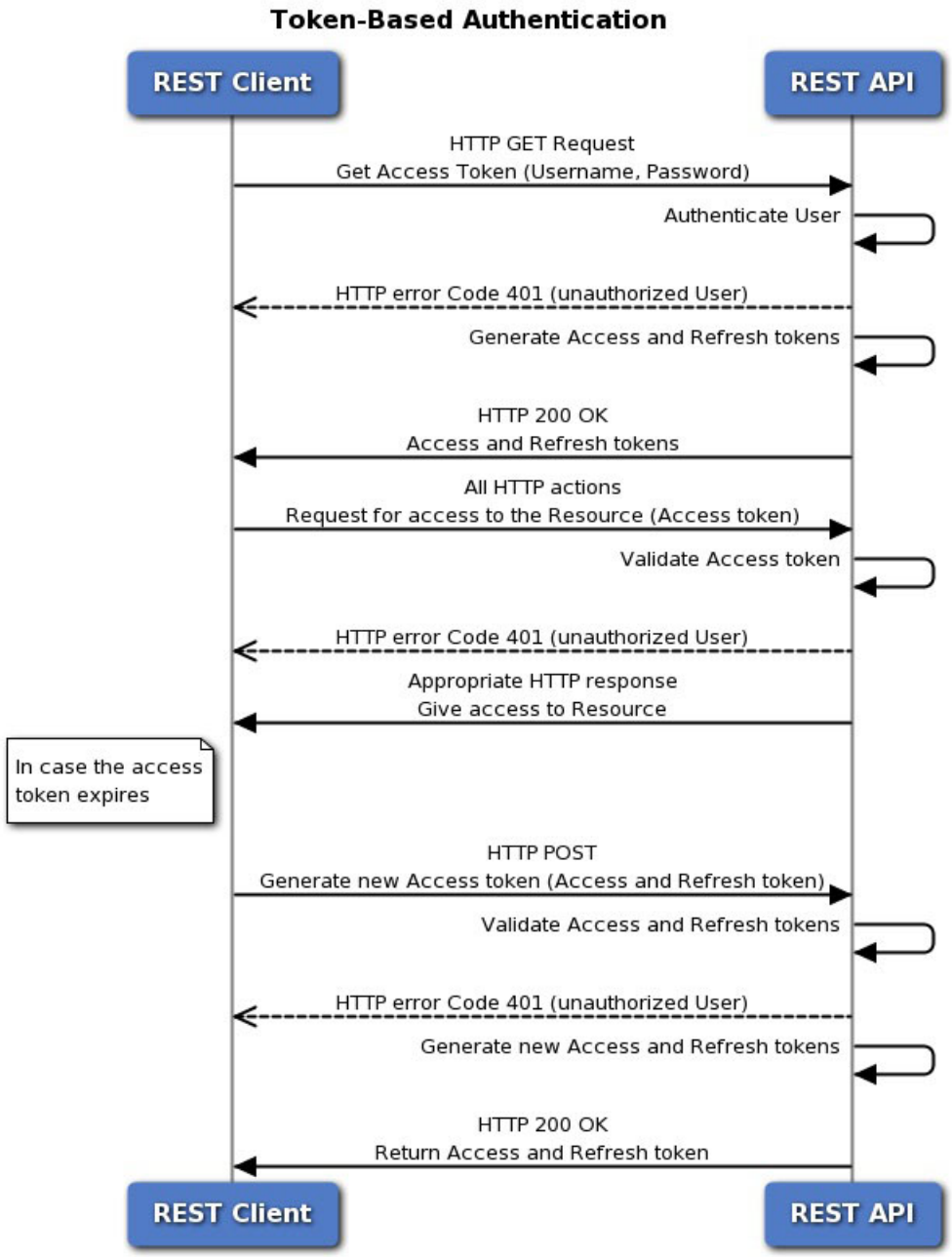
脅威に対する防御 REST API は、Oauth 2.0 を使用して API クライアントからのコールを認証します。OAuth はアクセストークンベースの方法であり、脅威に対する防御 はスキーマに JSON Web トークンを使用します。関連する規格は次のとおりです。

- RFC6749、OAuth 2.0 認証フレームワーク、https://tools.ietf.org/html/rfc6749。
- RFC7519、JSON Web トークン（JWT）、https://tools.ietf.org/html/rfc7519。

ここでは、必要なトークンを取得して使用する方法について説明します。

API クライアント認証プロセスの概要

脅威に対する防御 デバイスを使用して API クライアントを認証する方法のエンドツーエンド ビューを以下に示します。



始める前に

各トークンは、HTTPS ログインセッションを表します。これにより、API セッションと Device Manager セッションがカウントされます。最大 5 つのアクティブな HTTPS セッションが可能です。この制限を超えると、最も古いセッション（Device Manager ログインまたは API トークン）が期限切れになり、新しいセッションが許可されます。したがって、必要なトークンのみを取得し、期限切れになるまで各トークンを再利用してから、それらを更新することが重要です。API コールごとに新しいトークンを取得すると、深刻なセッションチャーンが発生し、

ユーザーが Device Manager からロックアウトされる可能性があります。これらの制限は、SSH セッションには適用されません。

手順

ステップ 1 必要な任意の方法を使用して API クライアント ユーザを認証します。

クライアントにはユーザーを認証する責任があるため、クライアントが 脅威に対する防御 デバイスにアクセスして変更する権限を持っていることを確認します。認証権限に基づいた差別化機能を提供する場合は、それをクライアントに構築する必要があります。

たとえば、読み取り専用アクセスを許可する場合は、必要な認証サーバ、ユーザアカウントなどを設定する必要があります。その後、読み取り専用権限を持つユーザがクライアントにログインすると、GET コールのみを発行するようにする必要があります。API v1 では、このタイプの変数アクセスは 脅威に対する防御 デバイス自体では制御できません。API v2以降では、外部ユーザーを使用していて、ユーザー認証に基づいてコールを調整していない場合、ユーザー認証と試行したコール間に不一致があるとエラーが表示されます。

v1 の場合、デバイスと通信するときは、[管理者（admin）] ユーザーアカウントを 脅威に対する防御 デバイスで使用する必要があります。[管理者（admin）] アカウントは、ユーザ設定可能なすべてのオブジェクトに対する完全な読み取り/書き込み権限を持っています。

ステップ 2 [管理者（admin）] アカウントを使用して、ユーザ名/パスワードに基づくパスワードが付与されたアクセス トークンを要求します。

パスワード付与アクセス トークンの要求 （4 ページ）を参照してください。

ステップ 3 必要に応じて、クライアントのカスタム アクセス トークンを要求します。

カスタム トークンを使用すると、有効期間を明示的に要求し、トークンにサブジェクト名を割り当てることができます。カスタム アクセス トークンの要求 （5 ページ）を参照してください。

ステップ 4 [認証：ベアラー（Authorization: Bearer）] ヘッダーで API コールのアクセス トークンを使用します。

API コールでのアクセス トークンの使用 （7 ページ）を参照してください。

ステップ 5 アクセス トークンの有効期限が切れる前にトークンを更新します。

アクセス トークンの更新 （8 ページ）を参照してください。

ステップ 6 完了したら、トークンの有効期限が切れていない場合は無効にします。

アクセス トークンの無効化 （10 ページ） を参照してください。

パスワード付与アクセス トークンの要求

発信者が要求されたアクションを実行する権限を持っていることを確認するための認証トークンが、すべての REST API コールに含まれている必要があります。最初に、[管理者（admin）] のユーザ名およびパスワードを指定してアクセス トークンを取得する必要があります。これはパスワード付与アクセス トークン（grant_type = password）と呼ばれます。

手順

ステップ 1 パスワード付与アクセス トークン付与のための JSON オブジェクトを作成します。

```
{
  "grant_type": "password",
  "username": "string",
  "password": "string"
}
```

[管理者（admin）] のユーザ名および正しいパスワードを指定します。たとえば、次のようになります。

```
{
  "grant_type": "password",
  "username": "admin",
  "password": "Admin123"
}
```

ステップ 2 POST /fdm/token を使用して、アクセストークンを取得します。

たとえば、**curl** コマンドは次のようになります。

```
curl -X POST --header 'Content-Type: application/json' --header
'Accept: application/json' -d '{
   "grant_type": "password",
   "username": "admin",
   "password": "Admin123"
 }' 'https://ftd.example.com/api/fdm/[最新（latest）]/fdm/token'
```

ステップ 3 応答からアクセス トークンと更新トークンを取得します。

正常な応答（ステータス コード 200）は次のようになります。

```
{
  "access_token": "eyJhbGciOiJIUzI1NiJ9.eyJpYXQiOjE1MDI4MzI2NjcsInN
1YiI6ImFkbWluIiwianRpIjoiMGM3ZDBmNDgtODIwMS0xMWU3LWE4MWMtMDcwZmYzOW
U3ZjQ0IiwibmJmIjoxNTAyODMyNjY3LCJleHAiOjE1MDI4MzQ0NjcsInJlZnJlc2hU
b2tlbkV4cGlyZXNBdCI6MTUwMjgzNTA2NzQxOSwidG9rZW5UeXBlIjoiSldUX0FjY2
VzcyIsIm9yaWdpbiI6InBhc3N3b3JkIn0.b2hI6fVA_GbmhCOPM-ZUx6IC8SgCk1Ak
HXI-llV0r7s",
  "expires_in": 1800,
  "token_type": "Bearer",
```

```
  "refresh_token": "eyJhbGciOiJIUzI1NiJ9.eyJpYXQiOjE1MDI4MzI2NjcsI
nN1YiI6ImFkbWluIiwia nRpIjoiMGM3ZDBmNDgtODIwMS0xMWU3LWE4MWMtMDcwZmY
zOWU3ZjQ0IiwibmJmIjoxNTAyODMyNjY3LCJleHAiOjE1MDI4MzUwNjcsImFjY2Vzc1
Rva2VuRXhwaXJlc0F0IjoxNTAyODM0NDY3NDE5LCJyZWZyZXNoQ291bnQiOi0xLCJ0b2
tlblR5cGUiOiJKV1RfUmVmcmVzaCIsIm9yaWdpbiI6InBhc3N3b3JkIn0.iLNqz1c1Xl
vcq0j9pQYW4gwYsvUCcSyaiDRXGutAz_o",
  "refresh_expires_in": 2400
}
```

説明：

- **access_token** は、APIコールに含める必要があるベアラートークンです。API コールでのアクセス トークンの使用 （7 ページ） を参照してください。
- **expires_in** は、アクセストークンが有効である、トークン発行時からの秒数です。
- **refresh_token** は、更新要求で使用するトークンです。アクセス トークンの更新 （8 ページ） を参照してください。
- **refresh_expires**_in は、更新トークンが有効である秒数です。これは、常にアクセス トークンの有効期間より長くなります。

カスタム アクセス トークンの要求

パスワード付与アクセス トークンを使用することができます。カスタム アクセス トークンを要求することもできます。カスタムトークンを使用すると、トークン使用を区別しやすくするために（自分自身のために）サブジェクト名を指定できます。パスワードトークン用に返されたデフォルト値が要件を満たさない場合は、特定の有効期間を要求することもできます。

始める前に

カスタム トークンを取得する前に、パスワード付与アクセス トークンを取得する必要があります。パスワード付与アクセス トークンの要求 （4 ページ） を参照してください。

また、次の点に注意してください。

- ローカルユーザーである場合のみ、カスタムトークンを要求できます。外部ユーザーはカスタムトークンを要求できません。
- 入手したユニットでのみカスタムトークンを使用できます。ハイ アベイラビリティ グループのピア デバイスではトークンを使用できません。

手順

ステップ 1 カスタム アクセス トークン付与のための JSON オブジェクトを作成します。

```
{
  "grant_type": "custom_token",
```

```
  "access_token": "string",
  "desired_expires_in": 0,
  "desired_refresh_expires_in": 0,
  "desired_subject": "string",
  "desired_refresh_count": 0
}
```

説明：

- **access_token** は、有効なパスワード付与アクセストークンです。
- **desired_expires_in** は、カスタムアクセストークンが有効である秒数を表す整数です。比較すると、パスワード付与トークンは 1800 秒間有効です。
- **desired_refresh_expires_in** は、カスタム更新トークンが有効である秒数を表す整数です。更新トークンを取得する場合は、この値が **desired_expires_in** 値より大きいことを確認してください。比較すると、パスワード付与更新トークンは 2400 秒間有効です。**desired_refresh_count** に 0 を指定した場合は、このパラメータは不要です。
- **desired_subject** は、カスタムトークンに付ける名前です。
- **desired_refresh_count** は、トークンを更新できる回数です。更新トークンを取得しない場合は、0 を指定します。更新トークンがない場合は、既存のアクセストークンの有効期限が切れるときに新しいアクセス トークンを取得する必要があります。

以下は、2400 秒で有効期限が切れる API クライアントのカスタム トークン、および 3000 秒で有効期限が切れる更新トークンを要求します。トークンは 3 回更新することができます。

```
{{
  "grant_type": "custom_token",
  "access_token": "eyJhbGciOiJIUzI1NiJ9.eyJpYXQiOjE1MDI4MzI2NjcsInN
1YiI6ImFkbWluIiwianRpIjoiMGM3ZDBmNDgtODIwMS0xMWU3LWE4MWMtMDcwZmYzOW
U3ZjQ0IiwibmJmIjoxNTAyODMyNjY3LCJleHAiOjE1MDI4MzQ0NjcsInJlZnJlc2hU
b2tlbkV4cGlyZXNBdCI6MTUwMjgzNTA2NzQxOSwidG9rZW5UeXBlIjoiSldUX0FjY2
VzcyIsIm9yaWdpbiI6InBhc3N3b3JkIn0.b2hI6fVA_GbmhCOPM-ZUx6IC8SgCk1Ak
HXI-llV0r7s",
  "desired_expires_in": 2400,
  "desired_refresh_expires_in": 3000,
  "desired_subject": "api-client",
  "desired_refresh_count": 3
}
```

ステップ 2 POST /fdm/token を使用して、アクセストークンを取得します。

たとえば、**curl** コマンドは次のようになります。

```
curl -X POST --header 'Content-Type: application/json' --header
'Accept: application/json' -d '{
   "grant_type": "custom_token",
   "access_token": "eyJhbGciOiJIUzI1NiJ9.eyJpYXQiOjE1MDI4MzU5N
jgsInN1YiI6ImFkbWluIiwianRpIjoiYmMyNjM4N2EtODIwOC0xMWU3LWE4MWM
tYzNlYTZkZjJjZThjIiwibmJmIjoxNTAyODM1OTY4LCJleHAiOjE1MDI4Mzc3N
jgsInJlZnJlc2hUb2tlbkV4cGlyZXNBdCI6MTUwMjgzODM2ODYwNiwidG9rZW5
UeXBlIjoiSldUX0FjY2VzcyIsIm9yaWdpbiI6InBhc3N3b3JkIn0.acOE_Y4SE
ds-NE4Qw99fQlUzdoSkhsjInaCh0a9WK38",
   "desired_expires_in": 2400,
```

```
    "desired_refresh_expires_in": 3000,
    "desired_subject": "api-client",
    "desired_refresh_count": 3
}' 'https://ftd.example.com/api/fdm/[最新（latest）]/fdm/token'
```

ステップ 3 応答からアクセス トークンと更新トークンを取得します。

正常な応答（ステータス コード 200）は次のようになります。

```
{
  "access_token": "eyJhbGciOiJIUzI1NiJ9.eyJpYXQiOjE1MDI4MzU5O
TEsInN1YiI6ImFwaS1jbGllbnQiLCJqdGkiOiJjOWIxYzdjYi04MjA4LTExZT
ctYTgxYy02YmY0NzY3ZmRmZGUiLCJuYmYiOjE1MDI4MzU5OTEsImV4cCI6MTU
wMjgzODM5MSwicmVmcmVzaFRva2VuRXhwaXJlc0F0IjoxNTAyODM4OTkxMzMx
LCJ0b2tlblR5cGUiOiJKV1RfQWNjZXNzIiwib3JpZ2luIjoiY3VzdG9tIn0.9
IVzLjGffVQffHAWdrNkrYfvuO6TgpJ7Zi_z3RYubN8",
  "expires_in": 2400,
  "token_type": "Bearer",
  "refresh_token": "eyJhbGciOiJIUzI1NiJ9.eyJpYXQiOjE1MDI4MzU5
OTEsInN1YiI6ImFwaS1jbGllbnQiLCJqdGkiOiJjOWIxYzdjYi04MjA4LTExZ
TctYTgxYy02YmY0NzY3ZmRmZGUiLCJuYmYiOjE1MDI4MzU5OTEsImV4cCI6MT
UwMjgzODk5MSwiYWNjZXNzVG9rZW5FeHBpcmVzQXQiOjE1MDI4MzgzOTEzMzE
sInJlZnJlc2hDb3VudCI6MywidG9rZW5UeXBlIjoiSldUX1JlZnJlc2giLCJv
cmlnaW4iOiJjdXN0b20ifQ.qseqjg3Uo183YvfN_77iJZELEqwpWw5AbKAqAn
CIcSA",
  "refresh_expires_in": 3000
}
```

説明：

- **access_token** は、API コールに含める必要があるベアラートークンです。API コールでのアクセス トークンの使用 （7 ページ） を参照してください。
- **expires_in** は、アクセストークンが有効である、トークン発行時からの秒数です。
- **refresh_token** は、更新要求で使用するトークンです。アクセス トークンの更新 （8 ページ） を参照してください。
- **refresh_expires_**in は、更新トークンが有効である秒数です。これは、常にアクセス トークンの有効期間より長くなります。

API コールでのアクセス トークンの使用

パスワード付与またはカスタム アクセス トークンを取得した後は、それを HTTPS 要求への [認証：ベアラー（Authorization: Bearer）] ヘッダーの各 API コールに含める必要があります。

たとえば、GET /object/networks を実行するための **curl** コマンドは次のようになります。

```
curl -k -X GET -H 'Accept: application/json'
-H 'Authorization: Bearer eyJhbGciOiJIUzI1NiJ9.yJp
YXQiOjE1MDI4MzU5OTEsInN1YiI6ImFwaS1jbGllbnQiLCJqdG
kiOiJjOWIxYzdjYi04MjA4LTExZTctYTgxYy02YmY0NzY3ZmRm
```

```
ZGUiLCJuYmYiOjE1MDI4MzU5OTEsImV4cCI6MTUwMjgzODM5MS
wicmVmcmVzaFRva2VuRXhwaXJlc0F0IjoxNTAyODM4OTkxMzMx
LCJ0b2tlblR5cGUiOiJKV1RfQWNjZXNzIiwib3JpZ2luIjoiY3
VzdG9tIn0.9IVzLjGffVQffHAWdrNkrYfvuO6TgpJ7Zi_z3RYu
bN8'
'https://ftd.example.com/api/fdm/[最新（latest）]/object/networks'
```

（注） APIエクスプローラを使用してメソッドおよびリソースを試す場合、表示される **curl** コマンドには **[認証**：**ベアラー（Authorization: Bearer）]** ヘッダーは含まれません。しかし、API クライアントから呼び出しを行う際にこのヘッダーを追加する必要があります。

アクセス トークンの更新

アクセス トークンの有効期限が切れたら、元の付与で提供された更新トークンを使用して更新する必要があります。更新されたアクセス トークンは、実際には元のアクセス トークンとは異なります。「更新」では、実際にアクセス トークンと更新トークンの新しいペアが提供され、古いアクセス トークンの期間が延長されるだけではありません。

手順

ステップ 1 更新トークン付与のための JSON オブジェクトを作成します。

```
{
  "grant_type": "refresh_token",
  "refresh_token": "string"
}
```

refresh_token はパスワード付与、またはカスタムアクセストークン付与から取得できます。

次に例を示します。

```
{
  "grant_type": "refresh_token",
  "refresh_token": "eyJhbGciOiJIUzI1NiJ9.eyJpYXQ
iOjE1MDI4MzU5OTEsInN1YiI6ImFwaS1jbGllbnQiLCJqdGk
iOiJjOWIxYzdjYi04MjA4LTExZTctYTgxYy02YmY0NzY3ZmR
mZGUiLCJuYmYiOjE1MDI4MzU5OTEsImV4cCI6MTUwMjgzODk
5MSwiYWNjZXNzVG9rZW5FeHBpcmVzQXQiOjE1MDI4MzgzOTE
zMzEsInJlZnJlc2hDb3VudCI6MywidG9rZW5UeXBlIjoiSld
UX1JlZnJlc2giLCJvcmlnaW4iOiJjdXN0b20ifQ.qseqjg3U
o183YvfN_77iJZELEqwpWw5AbKAqAnCIcSA"
}
```

ステップ 2 POST /fdm/token を使用して、更新されたアクセストークンを取得します。

たとえば、**curl** コマンドは次のようになります。

```
curl -X POST --header 'Content-Type: application/json' --header
'Accept: application/json' -d '{
   "grant_type": "refresh_token",
   "refresh_token": "eyJhbGciOiJIUzI1NiJ9.eyJpYXQiOjE1M
DI4MzU5OTEsInN1YiI6ImFwaS1jbGllbnQiLCJqdGkiOiJjOWIxYzdj
Yi04MjA4LTExZTctYTgxYy02YmY0NzY3ZmRmZGUiLCJuYmYiOjE1MDI
4MzU5OTEsImV4cCI6MTUwMjgzODk5MSwiYWNjZXNzVG9rZW5FeHBpcm
VzQXQiOjE1MDI4MzgzOTEzMzEsInJlZnJlc2hDb3VudCI6MywidG9rZ
W5UeXBlIjoiSldUX1JlZnJlc2giLCJvcmlnaW4iOiJjdXN0b20ifQ.q
seqjg3Uo183YvfN_77iJZELEqwpWw5AbKAqAnCIcSA"
 }' 'https://ftd.example.com/api/fdm/[最新（latest）]/fdm/token'
```

ステップ 3 応答からアクセス トークンと更新トークンを取得します。

正常な応答（ステータス コード200）は次のようになります。この例では、更新トークンはカスタム トークン用でした。有効期間は、元のカスタム アクセス トークン要求の値に基づいています。

```
{
  "access_token": "eyJhbGciOiJIUzI1NiJ9.eyJpYXQ
iOjE1MDI4Mzc1MTAsInN1YiI6ImFwaS1jbGllbnQiLCJqdG
kiOiJjOWIxYzdjYi04MjA4LTExZTctYTgxYy02YmY0NzY3Z
mRmZGUiLCJuYmYiOjE1MDI4Mzc1MTAsImV4cCI6MTUwMjgz
OTkxMSwicmVmcmVzaFRva2VuRXhwaXJlc0F0IjoxNTAyODQ
wNTEwNzQxLCJ0b2tlblR5cGUiOiJKV1RfQWNjZXNzIiwib3
JpZ2luIjoiY3VzdG9tIn0.fAAreX0DdnuqnM0Bs0NXYnI-9
jkpyW1pWDMwgwO_h7A",
  "expires_in": 2400,
  "token_type": "Bearer",
  "refresh_token": "eyJhbGciOiJIUzI1NiJ9.eyJpYX
QiOjE1MDI4Mzc1MTAsInN1YiI6ImFwaS1jbGllbnQiLCJqd
GkiOiJjOWIxYzdjYi04MjA4LTExZTctYTgxYy02YmY0NzY3
ZmRmZGUiLCJuYmYiOjE1MDI4Mzc1MTAsImV4cCI6MTUwMjg
0MDUxMCwiYWNjZXNzVG9rZW5FeHBpcmVzQXQiOjE1MDI4Mz
k5MTEwNzIsInJlZnJlc2hDb3VudCI6MiwidG9rZW5UeXBlI
joiSldUX1JlZnJlc2giLCJvcmlnaW4iOiJjdXN0b20ifQ.p
Adc2N0oun7Yyw872qK12pFlix4arAwyMETD1ErKu5c",
  "refresh_expires_in": 3000
}
```

説明：

- **access_token** は、API コールに含める必要があるベアラートークンです。API コールでのアクセス トークンの使用 （7 ページ） を参照してください。
- **expires_in** は、アクセストークンが有効である、トークン発行時からの秒数です。
- **refresh_token** は、更新要求で使用するトークンです。
- **refresh_expires_**in は、更新トークンが有効である秒数です。これは、常にアクセス トークンの有効期間より長くなります。

アクセス トークンの無効化

アクセス トークンは特定の期間有効であるため、ユーザが API クライアントからログアウトするときに、トークンを無効にすることによりクリーンアップする必要があります。これにより、脅威に対する防御デバイスへのバックドアが開いたままにならないことが確認されます。

手順

ステップ 1 無効化トークン付与のための JSON オブジェクトを作成します。

```
{
  "grant_type": "revoke_token",
  "access_token": "string",
  "token_to_revoke": "string",
  "custom_token_id_to_revoke": "string",
  "custom_token_subject_to_revoke": "string"
}
```

説明：

- **access_token** は、パスワード付与アクセストークンである必要があります。カスタム アクセス トークンを使用してトークンを無効にすることはできません。
- 次のうち 1 つ（1 つのみ）を指定する必要があります。
  - **token_to_revoke** は、無効にするパスワード付与トークンまたはカスタムトークンです。これは **access_token** と同じトークンにすることができるため、パスワード付与トークンを使用してそれ自体を無効にすることができます。
  - (使用不可.)**custom_token_id_to_revoke** は、内部の一意 ID によってカスタムアクセストークンを識別します。ただし、ユーザがこの値を取得する直接的な方法はありません。代わりにその他のオプションを使用します。
  - **custom_token_subject_to_revoke** は、無効にするカスタムアクセストークンの **desired_subject** 値です。

次に例を示します。

```
{
  "grant_type": "revoke_token",
  "access_token": "eyJhbGciOiJIUzI1NiJ9.eyJpYXQ
iOjE1MDI5MDQzMjQsInN1YiI6ImFkbWluIiwianRpIjoiZT
MzNGIxOWYtODJhNy0xMWU3LWE4MWMtNGQ3NzY2ZTExMzVkI
iwibmJmIjoxNTAyOTA0MzI0LCJleHAiOjE1MDI5MDYxMjQs
InJlZnJlc2hUb2tlbkV4cGlyZXNBdCI6MTUwMjkwNjcyNDE
xMiwidG9rZW5UeXBlIjoiSldUX0FjY2VzcyIsIm9yaWdpbi
I6InBhc3N3b3JkIn0.OVZBT9yVZc4zxZfZiiLH4SZcFclaH
yCPbZJC_Gyd5FE",
  "custom_token_subject_to_revoke": "api-client"
```

```
}
```

ステップ 2 POST/fdm/tokenを使用して、アクセストークンを無効化します。

たとえば、**curl** コマンドは次のようになります。

```
curl -X POST --header 'Content-Type: application/json'
--header 'Accept: application/json' -d '{
   "grant_type": "revoke_token",
   "access_token": "eyJhbGciOiJIUzI1NiJ9.eyJpYXQ
iOjE1MDI5MDQzMjQsInN1YiI6ImFkbWluIiwianRpIjoiZTM
zNGIxOWYtODJhNy0xMWU3LWE4MWMtNGQ3NzY2ZTExMzVkIiw
ibmJmIjoxNTAyOTA0MzI0LCJleHAiOjE1MDI5MDYxMjQsInJ
lZnJlc2hUb2tlbkV4cGlyZXNBdCI6MTUwMjkwNjcyNDExMiw
idG9rZW5UeXBlIjoiSldUX0FjY2VzcyIsIm9yaWdpbiI6InB
hc3N3b3JkIn0.OVZBT9yVZc4zxZfZiiLH4SZcFclaHyCPbZJ
C_Gyd5FE",
   "custom_token_subject_to_revoke": "api-client"
 }' 'https://ftd.example.com/api/fdm/[最新（latest）]/fdm/token'
```

ステップ 3 応答を評価して、トークンが無効になったことを確認します。

正常な応答（ステータス コード 200）は次のようになります。

```
{
  "message": "OK",
  "status_code": 200
}
```

翻訳について
このドキュメントは、米国シスコ発行ドキュメントの参考和訳です。リンク情報につきましては、日本語版掲載時点で、英語版にアップデートがあり、リンク先のページが移動/変更されている場合がありますことをご了承ください。あくまでも参考和訳となりますので、正式な内容については米国サイトのドキュメントを参照ください。