OAuth를 사용한 REST API 클라이언트 인증

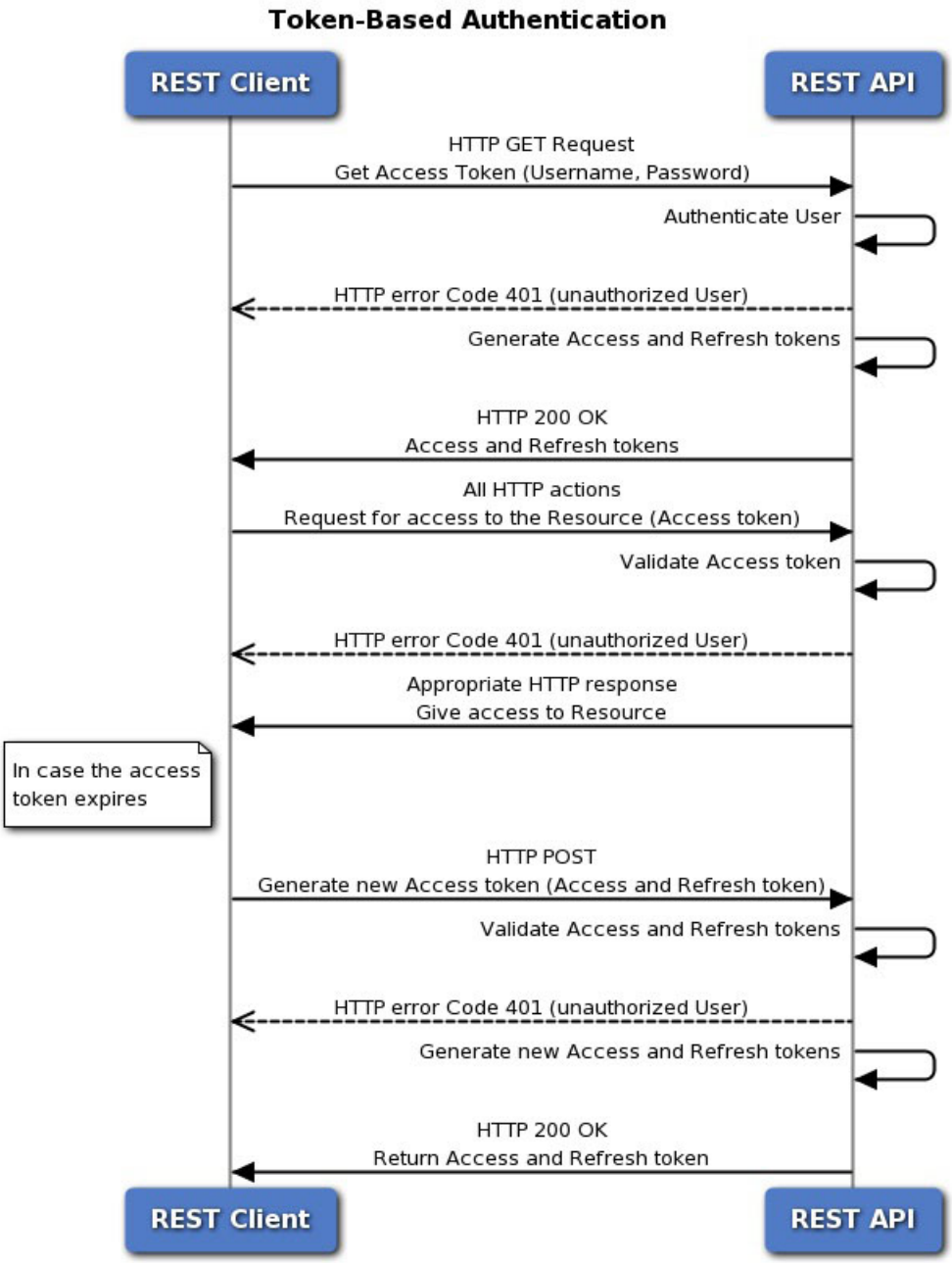
위협 방어 REST API에서는 API 클라이언트의 호출을 인증하는 데 Oauth 2.0을 사용합니다. OAuth는 액세스 토큰 기반 메서드이며, 위협 방어에서는 스키마에 JSON 웹 토큰을 사용합니다. 관련 표준은 다음과 같습니다.

- RFC6749, OAuth 2.0 Authorization Framework, https://tools.ietf.org/html/rfc6749
- RFC7519, JWT(JSON Web Token), https://tools.ietf.org/html/rfc7519

다음 주제에서는 필수 토큰을 획득하고 사용하는 방법에 대해 설명합니다.

API 클라이언트 인증 프로세스의 개요

다음은 API 클라이언트를 위협 방어 디바이스로 인증하는 방법에 대한 엔드 투 엔드 보기입니다.



시작하기 전에

각 토큰은 API 세션 및 device manager 세션에 포함되는 HTTPS 로그인 세션을 나타냅니다. 액티브 HTTPS 세션은 최대 5개까지 가능합니다. 이 제한을 초과하면 가장 오래된 세션인 device manager 로그인 또는 API 토큰이 만료되어 새 세션을 허용합니다. 따라서 필요한 토큰만 가져와 각 토큰이 만료될 때까지 재사용한 다음 갱신해야 합니다. 각 API 호출에 대해 새 토큰을 가져오면 심각한 세션 이탈이 발생하여 사용자를 device manager에서 잠글 수 있습니다. 이러한 제한은 SSH 세션에 적용되지 않습니다.

프로시저

단계 **1** 필요한 방법을 사용하여 API 클라이언트 사용자를 인증합니다.

클라이언트는 사용자를 인증해야 할 의무가 있으며 위협 방어 디바이스에 대한 액세스 및 수정 권한이 있어야 합니다. 부여 권한에 따라 차별적으로 기능을 제공하려면 해당 기능을 클라이언트에 구축해야 합니다.

예를 들어, 읽기 전용 액세스를 허용하려면 필요한 인증 서버, 사용자 어카운트 등을 설정해야 합니다. 그리고 나서 읽기 전용 권한을 가진 사용자가 클라이언트에 로그인할 때 GET 호출만 실행해야 합니다. API v1에서 이러한 유형의 변수 액세스는 위협 방어 디바이스에서 자체적으로 제어할 수 없습니다. API v2부터는 외부 사용자를 사용 중이며 사용자 권한 부여를 기준으로 호출을 정리하지 않는 경우 시도하는 호출과 사용자 권한 부여가 불일치하면 오류가 발생합니다.

v1의 경우 디바이스와 통신할 때 위협 방어 디바이스의 관리 사용자 계정을 사용해야 합니다. **admin** 어카운트에는 사용자가 구성 가능한 모든 개체에 대한 전체 읽기/쓰기 권한이 있습니다.

단계 **2** **admin(**관리자**)** 어카운트를 사용하여 사용자 이름/비밀번호를 기반으로 비밀번호 부여 액세스 토큰을 요청합니다.

비밀번호 부여 액세스 토큰 요청, 3 페이지의 내용을 참조하십시오.

단계 **3** 선택적으로 클라이언트용 맞춤형 액세스 토큰을 요청합니다.

맞춤형 토큰을 사용하면 유효 기간을 명시적으로 요청하고 토큰에 주체 이름을 할당할 수 있습니다.
맞춤형 액세스 토큰 요청, 5 페이지의 내용을 참조하십시오.

단계 **4** Authorization: Bearer(권한 부여: 전달자) 헤더의 API 호출에 액세스 토큰을 사용합니다.

API 호출에서 액세스 토큰 사용, 7 페이지의 내용을 참조하십시오.

단계 **5** 액세스 토큰이 만료되기 전에 토큰을 새로 고칩니다.

액세스 토큰 새로 고침, 8 페이지의 내용을 참조하십시오.

단계 **6** 작업을 완료한 경우 액세스 토큰이 아직 만료되지 않았다면 토큰을 취소합니다.

액세스 토큰 취소, 9 페이지의 내용을 참조하십시오.

비밀번호 부여 액세스 토큰 요청

모든 REST API 호출에는 호출자가 요청한 작업을 수행할 수 있는 권한이 있는지 확인하는 인증 토큰이 포함되어 있어야 합니다. 처음에는 **admin** 사용자 이름/비밀번호를 제공하여 액세스 토큰을 획득해야 합니다. 이를 비밀번호 부여 액세스 토큰(즉, grant_type = password)이라고 합니다.

프로시저

단계 **1** 비밀번호 부여 액세스 토큰을 부여받기 위해 JSON 개체를 생성합니다.

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

admin 사용자 이름 및 올바른 비밀번호를 지정합니다. 아래에 예시가 나와 있습니다.

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

단계 **2** POST /fdm/token을 사용하여 액세스 토큰을 획득합니다.

예를 들어 **curl** 명령은 다음과 같이 표시될 수 있습니다.

```
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/최신/fdm/token'
```

단계 **3** 액세스를 검색하고 응답에서 토큰을 새로 고칩니다.

양호한 응답(상태 코드 200)은 다음과 같이 표시됩니다.

```
{
  "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
}
```

여기서 각 항목은 다음을 나타냅니다.

- **access_token** API 호출에서 포함해야 하는 전달자 토큰입니다. API 호출에서 액세스 토큰 사용, 7 페이지의 내용을 참조하십시오.

- **expires_in** 토큰이 발행된 시간 이후부터 액세스 토큰이 유효한 시간(초)입니다.
- **refresh_token** 새로 고침 요청에서 사용할 토큰입니다. 액세스 토큰 새로 고침, 8 페이지의 내용을 참조하십시오.
- **refresh_expires_**in 새로 고침 토큰이 유효한 시간(초)입니다. 이는 항상 액세스 토큰 유효 기간보다 깁니다.

맞춤형 액세스 토큰 요청

비밀번호 부여 액세스 토큰을 사용할 수 있습니다. 그러나 맞춤형 액세스 토큰도 요청할 수 있습니다. 맞춤형 토큰을 사용하면 주체 이름을 제공하여 고유한 목적에 대해 차별적으로 토큰을 사용하는 데 도움이 될 수 있습니다. 비밀번호 토큰에 반환된 기본값이 요건에 맞지 않는 경우에는 특정한 유효 기간을 요청할 수도 있습니다.

시작하기 전에

맞춤형 토큰을 획득하기 전에 먼저 비밀번호 부여 액세스 토큰을 획득해야 합니다. 비밀번호 부여 액세스 토큰 요청, 3 페이지의 내용을 참조하십시오.

그 외에도,

- 로컬 사용자인 경우 맞춤형 토큰을 요청할 수 있습니다. 외부 사용자는 맞춤형 토큰을 요청할 수 없습니다.
- 토큰을 가져올 수 있는 유닛에서만 맞춤형 토큰을 사용할 수 있습니다. 고가용성 그룹의 피어 디바이스에서는 토큰을 사용할 수 없습니다.

프로시저

단계 **1** 맞춤형 액세스 토큰을 부여받기 위해 JSON 개체를 생성합니다.

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

여기서 각 항목은 다음을 나타냅니다.

- **access_token** 유효한 비밀번호 부여 액세스 토큰입니다.

- **desired_expires_in** 맞춤형 액세스 토큰이 유효한 시간(초)을 나타내는 정수입니다. 한편, 비밀번호 부여 토큰은 1800초 동안 유효합니다.
- **desired_refresh_expires_in** 맞춤형 새로 고침 토큰이 유효한 시간(초)을 나타내는 정수입니다. 새로고침 토큰을 가져오는 경우, 이 값이 **desired_expires_in** 값보다 커야 합니다. 한편, 비밀번호 부여 새로 고침 토큰은 2400초 동안 유효합니다. **desired_refresh_count**에 대해 0을 지정하는 경우에는 이 파라미터가 필요 없습니다.
- **desired_subject** 맞춤형 토큰에 지정한 이름입니다.
- **desired_refresh_count** 토큰을 새로 고칠 수 있는 횟수입니다. 새로 고침 토큰을 획득하지 않으려면 0으로 지정합니다. 새로 고침 토큰이 없는 경우 기존 토큰이 만료되면 새 액세스 토큰을 획득해야 합니다.

예를 들어, 다음은 2400초 후 만료되는 api-client에 대한 맞춤형 토큰을 요청하는 내용입니다. 이 예에서 새로 고침 토큰은 3000초 후 만료됩니다. 토큰은 3번을 새로 고칠 수 있습니다.

```
{{
  "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
}
```

단계 **2** POST /fdm/token을 사용하여 액세스 토큰을 획득합니다.

예를 들어 **curl** 명령은 다음과 같이 표시될 수 있습니다.

```
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/최신/fdm/token'
```

단계 **3** 액세스를 검색하고 응답에서 토큰을 새로 고칩니다.

양호한 응답(상태 코드 200)은 다음과 같이 표시됩니다.

```
{
```

```
  "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
}
```


여기서 각 항목은 다음을 나타냅니다.

- **access_token** API 호출에서 포함해야 하는 전달자 토큰입니다. API 호출에서 액세스 토큰 사용, 7 페이지의 내용을 참조하십시오.
- **expires_in** 토큰이 발행된 시간 이후부터 액세스 토큰이 유효한 시간(초)입니다.
- **refresh_token** 새로 고침 요청에서 사용할 토큰입니다. 액세스 토큰 새로 고침, 8 페이지의 내용을 참조하십시오.
- **refresh_expires**_in 새로 고침 토큰이 유효한 시간(초)입니다. 이는 항상 액세스 토큰 유효 기간보다 깁니다.

API 호출에서 액세스 토큰 사용

비밀번호 부여 액세스 토큰 또는 맞춤형 액세스 토큰 중 하나를 획득한 후에는 HTTPS 요청에 대한 **Authorization: Bearer(**권한 부여**:** 전달자**)** 헤더의 각 API 호출에 이 토큰을 포함해야 합니다.

예를 들어 GET /object/networks 작업을 수행하기 위한 **curl** 명령은 다음과 같이 표시될 수 있습니다.


```
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/최신/object/networks'
```

참고 API Explorer를 사용하여 메서드 및 리소스를 시도하는 경우, 표시되는 **curl** 명령에는 **Authorization: Bearer(**권한 부여**:** 전달자**)** 헤더가 포함되어 있지 않습니다. 그러나 API 클라이언트에서 호출할 때는 이 헤더를 추가해야 합니다.

액세스 토큰 새로 고침

액세스 토큰이 만료되고 나면 기존에 부여받았던 새로 고침 토큰을 사용하여 해당 토큰을 새로 고쳐야 합니다. 새로 고친 액세스 토큰은 실제로 기존의 액세스 토큰과 다릅니다. "새로 고침"을 수행하면 단순히 기존의 액세스 토큰의 수명이 연장되는 것이 아니라 실제로 새로운 쌍의 액세스 토큰과 새로 고침 토큰이 제공됩니다.

프로시저

단계 **1** 새로 고침 토큰을 부여받기 위해 JSON 개체를 생성합니다.

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

refresh_token은 암호 부여 액세스 토큰 또는 맞춤형 액세스 토큰에서 부여받을 수 있습니다.

예를 들면 다음과 같습니다.

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

단계 **2** POST /fdm/token을 사용하여 새로 고쳐진 액세스 토큰을 획득합니다.

예를 들어 **curl** 명령은 다음과 같이 표시될 수 있습니다.

```
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/최신/fdm/token'
```

단계 **3** 액세스를 검색하고 응답에서 토큰을 새로 고칩니다.

양호한 응답(상태 코드 200)은 다음과 같이 표시됩니다. 이 예에서는 새로 고침 토큰이 맞춤형 토큰에 사용되었습니다. 만료 기간은 기존 맞춤형 액세스 토큰 요청의 값을 기반으로 합니다.

```
{
  "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
}
```

여기서 각 항목은 다음을 나타냅니다.

- **access_token** API 호출에서 포함해야 하는 전달자 토큰입니다. API 호출에서 액세스 토큰 사용, 7 페이지의 내용을 참조하십시오.
- **expires_in** 토큰이 발행된 시간 이후부터 액세스 토큰이 유효한 시간(초)입니다.
- **refresh_token** 새로 고침 요청에서 사용할 토큰입니다.
- **refresh_expires**_in 새로 고침 토큰이 유효한 시간(초)입니다. 이는 항상 액세스 토큰 유효 기간보다 깁니다.

액세스 토큰 취소

액세스 토큰은 특정 기간 동안 유효하므로 API 클라이언트에서 로그아웃할 때 토큰을 취소하여 정리해야 합니다. 이렇게 하면 백도어가 위협 방어 디바이스로 열려 있지 않게 할 수 있습니다.

프로시저

단계 **1** 취소 토큰을 부여받기 위해 JSON 개체를 생성합니다.

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

여기서 각 항목은 다음을 나타냅니다.

- **access_token** 비밀번호 부여 액세스 토큰이어야 합니다. 맞춤형 액세스 토큰을 사용해서는 토큰을 취소할 수 없습니다.
- 다음 중 하나만 지정해야 합니다.
  - **token_to_revoke** 취소하려는 비밀번호 부여 토큰 또는 맞춤형 토큰입니다. 이것은 **access_token**과 동일한 토큰일 수 있으며, 이 경우에는 암호 부여 토큰을 사용하여 토큰을 자체적으로 취소할 수 있습니다.
  - (사용하지 않음) **custom_token_id_to_revoke**에서는 내부 고유 ID로 맞춤형 액세스 토큰을 식별합니다. 그러나 이 값을 얻을 수 있는 직접적인 방법은 없으므로 다른 옵션을 대신 사용합니다.
  - **custom_token_subject_to_revoke**는 취소하려는 맞춤형 액세스 토큰에 대한 **desired_subject** 값입니다.

예를 들면 다음과 같습니다.

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

단계 **2** POST /fdm/token을 사용하여 액세스 토큰을 취소합니다.

예를 들어 **curl** 명령은 다음과 같이 표시될 수 있습니다.

```
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/최신/fdm/token'
```

단계 **3** 응답을 평가하여 토큰이 취소되었는지 확인합니다.

양호한 응답(상태 코드 200)은 다음과 같이 표시됩니다.

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