# メソッドとリソースの使用

次のトピックではさまざまなメソッドとリソースを使用する一般的な方法を説明します。



## メソッドの試行と結果の解釈

API エクスプローラを使用すると、さまざまなメソッドをテストすることができます。このトピックでは、一般的なプロセスについて、システムが返す応答の説明とともに説明します。メソッドに関連する具体的なテクニックは、各メソッドタイプのトピックを参照してください。

各メソッド/リソースの [試してみよう!（Try It Out!）] ボタンにより、システムと直接対話します。GET は、実際のデータを取得し、POST/PUT は実際のリソースを作成または変更し、DELETE は実際のオブジェクトを削除します。変更はすぐには展開されませんが、システムで実際の設定変更を行っています。変更をアクティブにするには、POST /operational/deploy リソースを使用して展開ジョブを開始します。

メソッド/リソースを開くと、[応答メッセージ（Response Message）] セクションの後に、[試してみよう!（Try It Out!）] ボタンを見つけることができます。メソッド/リソースによっては、テストするにはオブジェクト ID を入力する必要があります。この場合、通常、まず親リソースで GET を実行する必要があります。詳細については、オブジェクト ID（objId）と親 ID の検索を参照してください。

POST/PUT でも、JSON モデルに必要な値を入力する必要があります。

[試してみよう!（Try It Out!）] をクリックした後、API エクスプローラは、ボタンの後のページに結果を追加します。この応答には次のセクションが含まれます。

### Curl

コールを行うために使用する **curl** コマンド。たとえば、GET /object/networks リソースで **[試してみよう!（Try It Out!）]** をクリックすると、次のようなメッセージが返されます。パスの要素「v」は、新しいバージョンの API によって変わります。

```
curl -X GET --header 'Accept: application/json'
'https://ftd.example.com/api/fdm/[最新（latest）]/object/networks'
```

（注） これには [認証：ベアラー（Authorization: Bearer）] ヘッダーは含まれません。これはクライアントからの API 呼び出しで必要になります。

### 要求 URL

要求を行うクライアントから発行する URL です。たとえば、GET /object/networks の場合：

```
https://ftd.example.com/api/fdm/[最新（latest）]/object/networks
```

### レスポンス ボディ

システムがクライアントに返すオブジェクト。リソースに複数のオブジェクト（/object/network など）を含めることができる場合、GET 要求でアイテムのリストを取得します。POST/PUT/DELETE の応答は、ほとんど単一のオブジェクトです。

返される特定のコンテンツは、リソース モデルに基づきます。たとえば、GET /object/networks は次のような各オブジェクトを含む、オブジェクトのリストを返します（最初のアイテムのリストも表示されます）。リンク/自己値は、このオブジェクトを参照するために使用する URL を示すことに注意してください。オブジェクト ID は URL に含まれます。

```
{
  "items": [
    {
      "version": "900f8558-7d19-11e7-bf7b-3dcaf0c58345",
      "name": "AIM_SERVERS-205.188.1.132",
      "description": null,
      "subType": "HOST",
      "value": "205.188.1.132",
      "isSystemDefined": true,
      "id": "900fac69-7d19-11e7-bf7b-d9417b20e59e",
      "type": "networkobject",
      "links": {
        "self": "https://ftd.example.com/api/fdm/[最新（latest）]/
object/networks/900fac69-7d19-11e7-bf7b-d9417b20e59e"
      }
    },
```

GET 要求もページングのセクションに含まれます。これはGET：システムからのデータの取得 （3 ページ）で説明されています。

### 応答コード

HTTP 呼び出しの数値 HTTP ステータス コードが返されます。これらは、標準の HTTP ステータス コードで、RFC やウィキペディア（https://en.wikipedia.org/wiki/List_of_HTTP_status_codes など）で検索できます。たとえば、200 (OK) は、GET/PUT/POST 呼び出しの成功を示し、204 は DELETE 呼び出しの成功を示します。

### レスポンス ヘッダー

HTTP 応答のパケット ヘッダーです。たとえば、GET /object/networks には、次のようなヘッダーがあります。

```
{
  "date": "Thu, 10 Aug 2017 19:19:16 GMT",
  "content-encoding": "gzip",
  "x-content-type-options": "nosniff",
  "transfer-encoding": "chunked",
  "connection": "Keep-Alive",
  "vary": "Accept-Encoding",
  "x-xss-protection": "1; mode=block",
  "pragma": "no-cache",
  "server": "Apache",
  "x-frame-options": "SAMEORIGIN",
  "strict-transport-security": "max-age=31536000 ; includeSubDomains",
  "content-type": "application/json;charset=UTF-8",
  "cache-control": "no-cache, no-store, max-age=0, must-revalidate",
  "accept-ranges": "bytes",
  "keep-alive": "timeout=5, max=99",
  "expires": "0"
}
```

# GET：システムからのデータの取得

デバイスから情報を読み取るには、GET メソッドを使用します。

リソースに複数のオブジェクトが含まれている可能性がある場合は、応答でオブジェクトのリストを取得します。URL にクエリ パラメータを含めて、返されるオブジェクトの数を制御することができます。デフォルトでは、オブジェクト リストの先頭から 10 個のオブジェクトが返されます。

次の手順では、API エクスプローラで GET コールを使用する一般的なアプローチを説明します。API クライアントのコード例を使用します。

### 手順

**ステップ 1** APIエクスプローラでGETメソッドを開きます（最初に、グループを開いてメソッドとリソースを参照します）。

**ステップ 2** 使用するメソッドがURL内にオブジェクトまたは親のIDを必要とする場合は、親メソッドを使用して必要なID を取得します。

たとえば、GET /objects/networks/{objId} は特定のオブジェクトの ID を必要とします。GET /objects/networks メソッドを使用してネットワーク オブジェクトのリストを取得し、調べるオブジェクトの id 値を探します。この場合、GET /object/networks コールで返される情報は GET /objects/network/{objId} の場合と同じになることに注意してください。オブジェクト ID（objId）と親 ID の検索を参照してください。

**ステップ 3** **Parameters** セクションで、以下のオプションを設定します。

- **objId**：オブジェクト ID は、URL 内で必要な場合は常に必要です。たとえば、900fac69-7d19-11e7-bf7b-d9417b20e59e です。
- **parentId**：親 ID はオブジェクト ID と同等ですが、単に階層内で高い位置にある親に対するものです。たとえば、GET /policy/intrusion は侵入ポリシーのリストを返しますが、GET /policy/intrusion/{parentId}/intrusionrules はそれらのポリシーの 1 つで定義されているルールを返します。親 ID は GET /policy/intrusion から取得します。
- **offset**：複数のオブジェクトをサポートするリソースの場合、オブジェクトを返すのを開始するリスト内の位置。デフォルトは 0 で、リストの先頭を示します。
- **limit**：応答で返されるオブジェクトの最大数。デフォルトは 10 です。最大値は 1000 です。無効な値を入力すると、自動的に 1000 に変更されます。
- **sort**：応答で返されるオブジェクトを並べ替える方法。デフォルトの並べ替え順は、**name** 値でアルファベット順です。並べ替え順を変更するには、並べ替えに使用するリソース内の属性の名前を入力します。たとえば、ネットワーク オブジェクトで **sort=value** を使用すると、value 属性（つまり IP アドレス）で並べ替えることができます。逆順に並べ替えるには、マイナス記号を含めます（例：**sort=-name**）。
- **filter**（一部のリソースでは使用不可）：フィルタ条件に一致する項目のみを返します。フィルタ値の形式は {キー}{演算子}{値} で、キーは属性名、値はフィルタに使用する文字列です。項目間にスペースはありません。フィルタ処理できるフィールドは、API エクスプローラの **filter** パラメータの説明に一覧表示されています。フィールドはオブジェクトごとに異なります。複数フィールドのフィルタリングをサポートしているオブジェクトの場合、複数の値をセミコロン（;）で区切って **filter** パラメータに含めることができます。たとえば、GET /policy/intrusionpolicies/{parentId}/intrusionrules の gid:1;sid:105 をフィルタ処理できます。使用可能な演算子は次のとおりです。
  - **:**：等号。たとえば、**filter=name!Canada** です。
  - **!**：不等号。たとえば、**filter=name!Canada** です。
  - **~**：類似。たとえば、**filter=name~United** です。
- **filter=fts~***string*（一部のリソースでは使用不可）：フィルタに一致する項目のみを返します。**Fts~** オプションはフルテキスト検索を指定します。オブジェクト内の全属性で文字列が検索されます。オプションで部分文字列を使用できます。アスタリスク（*）をワイルドカードとして使用し、1 文字以上と一致させます。次の文字を含めないでください。検索文字列の一部としてサポートされていません。?~!{}<>:%。次の文字は無視されます。;#&。

たとえば、GET /object/networks?filter=fts~10 を使用して、最初のオクテットとして 10 を持つすべてのネットワークオブジェクトを見つけることができます。新しく作成された、または更新されたオブジェクトのインデックス化には 3 ～ 5 秒かかるため、それらのオブジェクトのフルテキスト検索を実行する前に少し待つ必要があります。

- **filter=fetchZeroHitCount:{true | false}**（アクセスルールのみで使用可能）：**includeHitCounts=true** を指定した場合、このフィルタオプションを使用して、ヒットしていない（つまりヒットカウントがゼロである）ルールを含める（**true**）または除外（**false**）することができます。デフォルトは **true** です。
- **includeHitCounts**（アクセスルールのみで使用可能）：ポリシーにルールのヒットカウント情報を含めるかどうか。ヒットカウントを取得するには、**includeHitCounts=true** を指定します。**false**（デフォルト）を指定してヒットカウントを除外します。ヒットカウント情報は、返されたオブジェクトの hitCount 属性で返されます。
- **time_duration**（トレンド レポートの場合にのみ使用可能）：過去何秒間までのレポートを含めるか。たとえば、1800 は過去 30 分のレポートを返します。

（注） {objId} と {parentId} は URL パスの一部ですが、**offset**、**limit**、**sort**、**filter**、**includeHitCounts**、および **time_duration** の各パラメータは URL の末尾に **?** 文字に続けて追加します。

**ステップ 4** [試してみよう!（Try It Out!）] ボタンをクリックし、応答を調べます。

コールが成功すると（戻りコード 200）、応答本文にはコールの内容に応じて 1 つのオブジェクトまたはオブジェクトのリストが含まれています。応答の一般的な構造と内容については、メソッドの試行と結果の解釈 （1 ページ）を参照してください。

GET 要求にはページング セクションが含まれています。コールで返されたオブジェクトより多くのオブジェクトが存在する場合、**prev** および **next** の値はその前または後のオブジェクトセットを取得する方法を示しています。**count** 値はオブジェクト全体の数を示しています。**limit** 値は応答で返される項目数を示しています。**offset** 値は返されるオブジェクトの開始位置を示しており、0 はリストの先頭を示しています。

```
"paging": {
    "prev": [],
    "next": [
     "https://ftd.example.com/api/fdm/[最新（latest）]/object/networks?limit=10&offset=10"

    ],
    "limit": 10,
    "offset": 0,
    "count": 22,
    "pages": 0
  }
```

# POST：新しいオブジェクトの作成

あるリソース タイプの新しいオブジェクトを作成するには、POST メソッドを使用します。たとえば、POST を使用して新しいネットワーク オブジェクトを作成します。

次の手順では、API エクスプローラで POST コールを使用する一般的なアプローチを説明します。API クライアントのコード例を使用します。

### 手順

**ステップ 1** API エクスプローラで POST メソッドを開きます（最初に、グループを開いてメソッドとリソースを参照します）。

**ステップ 2** [応答クラス（Response Class）] 見出しで [モデル（Model）] をクリックし、リソース属性のデータ型および値について読みます。

**ステップ 3** **[パラメータ（Parameters）]** 見出しの下で、以下のオプションを設定します（設定可能な場合）。

- **[parentId]**：このオブジェクトを含む親オブジェクトの ID。たとえば、SSL ルールを追加する場合は、SSL 復号ポリシーの ID。
- **[at]**：番号付きリストでオブジェクトを編成する親に属するオブジェクト（SSL 復号ポリシーなど）の場合は、その場所をオブジェクトに挿入します。整数を使用して場所を示します（リストの先頭は 0）。デフォルトでは、新しいオブジェクトはリストの最後に追加されます。

（注） {objId} と {parentId} は URL パスの一部ですが、**at**パラメータを URL の末尾の **?** 文字に続けて追加します。

**ステップ 4** また、**[パラメータ（Parameters）]** 見出しの下で、**body** パラメータの **[データ型（Data Type）]** > **[サンプル値（Example Value）]** 列に表示される JSON モデルをクリックします。

このボックスをクリックすると、JSON モデルが **body** パラメータの [値（Value）] 列にロードされます。たとえば、POST /object/networks リソースのボックスをクリックすると、次の本文がロードされます。

```
{
  "name": "string",
  "description": "string",
  "subType": "HOST",
  "value": "string",
  "type": "networkobject"
}
```

**ステップ 5** **body** JSON オブジェクト属性の必須値を入力します。

列挙値の場合は、必ず**[応答クラス（Response Class）]**>**[モデル（Model）]**で許可される値を確認してください。たとえば、値を入力して[subType]のデフォルト値を変更することにより、サブネット（ホストではない）アドレスのネットワーク オブジェクトを作成できます。

```
{
  "name": "new_network_object",
  "description": "A subnet object created using the REST API.",
  "subType": "NETWORK",
  "value": "10.100.10.0/24",
  "type": "networkobject"
}
```

**ステップ 6** [試してみよう!（Try It Out!）] ボタンをクリックし、応答を調べます。

システムの更新に使用される**curl**コマンドを調べます。追加のヘッダーに注意してください。APIクライアントを作成するときは、これらのヘッダーフィールドと値も含める必要があります。たとえば、サンプルオブジェクトを作成するための**curl**コマンドは次のとおりです。Content-Type ヘッダーと Accept ヘッダーに注意してください。

```
curl -X POST --header 'Content-Type: application/json' \
   --header 'Accept: application/json' -d '{ \
   "name": "new_network_object", \
   "description": "A subnet object created using the REST API.", \
   "subType": "NETWORK", \
   "value": "10.100.10.0/24", \
   "type": "networkobject" \
 }' 'https://ftd.example.com/api/fdm/[最新（latest）]/object/networks'
```

コールが成功すると（戻りコード200）、応答本文には作成した完全なオブジェクトが含まれていて、**version**や**id**などのその他のシステム生成値も含まれています。バージョンおよびIDの値は、その後オブジェクトを変更するためにPUTを使用する場合に必要になるため、特に重要です。応答の一般的な構造と内容については、メソッドの試行と結果の解釈（1ページ）を参照してください。

応答本文には、作成したオブジェクトのURLである**links/self**値も含まれています。たとえば、サンプル オブジェクトの応答本文は次のとおりです。

```
{
  "version": "f6d8da48-7ed5-11e7-9bfd-d96183b5f5f1",
  "name": "new_network_object",
  "description": "A subnet object created using the REST API.",
  "subType": "NETWORK",
  "value": "10.100.10.0/24",
  "isSystemDefined": false,
  "id": "f6d8da49-7ed5-11e7-9bfd-27136f5686ad",
  "type": "networkobject",
  "links": {
    "self": "https://ftd.example.com/api/fdm/[最新（latest）]/object/networks/
f6d8da49-7ed5-11e7-9bfd-27136f5686ad"
  }
}
```

# PUT：既存のオブジェクトの変更

既存のオブジェクトの属性を変更するには、PUT メソッドを使用します。たとえば、PUT を使用して、既存のネットワーク オブジェクトに含まれているアドレスを、オブジェクトの ID を変更せずに変更します。

PUT メソッドはオブジェクト全体を置き換えます。単に 1 つの属性を変更することはできません。したがって、保存する古い値が JSON オブジェクトに含まれていることを確認する必要があります。

次の手順では、API エクスプローラで PUT コールを使用する一般的なアプローチを説明します。API クライアントのコード例を使用します。

### 始める前に

オブジェクトの既存の状態のコピーを取得するには、親リソースに対して GET メソッドを使用します（GET：システムからのデータの取得 （3 ページ）を参照）。

少なくとも以下のパラメータに対する正しい値、および変更しないユーザ指定値が必要です。

- **version**
- **id**

### 手順

**ステップ 1** API エクスプローラで PUT メソッドを開きます（最初に、グループを開いてメソッドとリソースを参照します）。

**ステップ 2** **[パラメータ（Parameters）]** 見出しの下で、以下のオプションを設定します。

- **[objId]**：オブジェクトの **id** 値。たとえば、900fac69-7d19-11e7-bf7b-d9417b20e59e です。
- **[parentId]**：別のオブジェクト内にあるオブジェクトの場合、このオブジェクトを含む親オブジェクトの ID。たとえば、SSL ルールを変更する場合は、SSL 復号ポリシーの ID。
- **[at]**：番号付きリストでオブジェクトを編成する親に属するオブジェクト（SSL 復号ポリシーなど）の場合は、その場所をオブジェクトに挿入します。整数を使用して場所を示します（リストの先頭は 0）。デフォルトでは、オブジェクトはリストの最後に挿入されます。

（注） {objId} と {parentId} は URL パスの一部ですが、**at**パラメータを URL の末尾の **?** 文字に続けて追加します。

**ステップ 3** また、**[パラメータ（Parameters）]** 見出しの下で、**body** パラメータの **[データ型（Data Type）]** > **[サンプル値（Example Value）]** 列に表示される JSON モデルをクリックします。

このボックスをクリックすると、JSON モデルが **body** パラメータの [値（Value）] 列にロードされます。たとえば、PUT /object/networks リソースのボックスをクリックすると、次の本文が

ロードされます。これは、同じリソースに対するPOSTバージョンとは少し異なっていることに注意してください。PUT本文には**version**属性が含まれています。

```
{
  "version": "string",
  "name": "string",
  "description": "string",
  "subType": "HOST",
  "value": "string",
  "type": "networkobject"
}
```

**ステップ4** **body** JSONオブジェクト属性の必須値を入力します。

変更しない古い値を必ず複製してください。

列挙値の場合は、必ず**[応答クラス（Response Class）]**>**[モデル（Model）]**で許可される値を確認してください。オブジェクトを別のサブタイプに変更する場合を除き、古い値を繰り返し使用します。たとえば、ネットワークオブジェクトのデフォルトのPUTモデルでは**subType**はHOSTですが、サブネットオブジェクトを変更する場合は必ず**subType**をNETWORKに変更します。

たとえば、ネットワークオブジェクト内のサブネットIPアドレスを更新するには、**value**を除くすべての属性の古い値をすべて繰り返し使用します。新しいサブネットアドレスを**value**に入力します。

```
{
  "version": "f6d8da48-7ed5-11e7-9bfd-d96183b5f5f1",
  "name": "new_network_object",
  "description": "A subnet object created using the REST API.",
  "subType": "NETWORK",
  "value": "10.100.11.0/24",
  "type": "networkobject",
}
```

**ステップ5** [試してみよう!（Try It Out!）]ボタンをクリックし、応答を調べます。

システムの更新に使用される**curl**コマンドを調べます。追加のヘッダーに注意してください。APIクライアントを作成するときは、これらのヘッダーフィールドと値も含める必要があります。たとえば、サンプルオブジェクトを更新するための**curl**コマンドは次のとおりです。Content-Typeヘッダーと Acceptヘッダーに注意してください。

```
curl -X PUT --header 'Content-Type: application/json' \
--header 'Accept: application/json' -d '{ \
   "version": "f6d8da48-7ed5-11e7-9bfd-d96183b5f5f1", \
   "name": "new_network_object", \
   "description": "A subnet object created using the REST API.", \
   "subType": "NETWORK", \
   "value": "10.100.11.0/24", \
   "type": "networkobject" \
 }' 'https://ftd.example.com/api/fdm/[最新（latest）]/object/
networks/f6d8da49-7ed5-11e7-9bfd-27136f5686ad'
```

コールが成功すると（戻りコード 200）、応答本文には更新した完全なオブジェクトが含まれています。バージョン値は変更されますが、オブジェクト ID は（したがって link/self も）同じままです。オブジェクトを変更するたびに、バージョンが変更されます。応答の一般的な構造と内容については、メソッドの試行と結果の解釈 （1 ページ）を参照してください。

（注） オブジェクトに何も変更を加えなかった場合、つまり、更新されるオブジェクトが以前のバージョンと同じである場合、要求は処理されず、そのリソースに対して何も変更されていないことを伝える 204 コードが返されます。

たとえば、サンプル オブジェクト更新の応答本文は次のとおりです。

```
{
  "version": "96f5f3cc-7ede-11e7-9bfd-9b7d8a92863f",
  "name": "new_network_object",
  "description": "A subnet object created using the REST API.",
  "subType": "NETWORK",
  "value": "10.100.11.0/24",
  "isSystemDefined": false,
  "id": "f6d8da49-7ed5-11e7-9bfd-27136f5686ad",
  "type": "networkobject",
  "links": {
    "self": "https://ftd.example.com/api/fdm/[最新（latest）]/object/networks/
f6d8da49-7ed5-11e7-9bfd-27136f5686ad"
  }
}
```

# DELETE：ユーザが作成したオブジェクトの削除

自分または別のユーザが作成したオブジェクトを削除するには、DELETE メソッドを使用します。たとえば、不要になったネットワーク オブジェクトを削除するには、DELETE を使用します。

システム定義オブジェクトや存在する必要があるオブジェクトは削除できません。

また、アクセス ルールで使用されているネットワーク オブジェクトなど、別のオブジェクトによって現在使用されているオブジェクトも削除できません。使用中のオブジェクトについては、そのオブジェクトを使用しているすべてのオブジェクトを変更してからそのオブジェクトを削除します。

次の手順では、APIエクスプローラでDELETEコールを使用する一般的なアプローチを説明します。API クライアントのコード例を使用します。

### 始める前に

オブジェクトの既存の状態のコピーを取得するには、親リソースに対して GET メソッドを使用します（GET：システムからのデータの取得 （3 ページ）を参照）。

オブジェクトを削除するには、オブジェクト ID（**id** 値）が必要です。

### 手順

**ステップ 1** APIエクスプローラでDELETEメソッドを開きます（最初に、グループを開いてメソッドとリソースを参照します）。

**ステップ 2** [パラメータ（Parameters）] 見出しの下で、オブジェクトの [id] 値を [objId] フィールドに入力します。たとえば、f6d8da49-7ed5-11e7-9bfd-27136f5686ad です。

該当オブジェクトがコンテナ内に存在する場合は、親オブジェクトの ID も **[parentId]** フィールドに入力する必要があります。

**ステップ 3** [試してみよう!（Try It Out!）] ボタンをクリックし、応答を調べます。

システムからオブジェクトを削除するために使用する **curl** コマンドを調べます。追加のヘッダーに注意してください。APIクライアントを作成するときは、これらのヘッダーフィールドと値も含める必要があります。たとえば、サンプルオブジェクトを削除するための **curl** コマンドは以下のとおりです。Accept ヘッダーに注意してください。

```
curl -X DELETE --header 'Accept: application/json'
'https://ftd.example.com/api/fdm/[最新（latest）]/object/
networks/f6d8da49-7ed5-11e7-9bfd-27136f5686ad'
```

コールが成功すると（戻りコード204「コンテンツがありません」）、空の応答本文を受け取ります。これは予期されている結果です。

- **DELETE**：ユーザが作成したオブジェクトの削除

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。