

# REST-APIを使用したFMCへの一括オブジェクトのプッシュ

## 内容

[概要](#)

[前提条件](#)

[要件](#)

[使用するコンポーネント](#)

[制限](#)

[背景説明](#)

[設定](#)

[確認](#)

[トラブルシューティング](#)

## 概要

このドキュメントでは、アプリケーションプログラミングインターフェイス(API)管理者がネットワーク、ポート、およびURLオブジェクトをFirepower Management Center(FMC)に一括でプッシュする方法について説明します。

## 前提条件

### 要件

次の項目に関する知識があることが推奨されます。

- 各種のREST APIコールについて。( [REST APIとは](#) )
- FMC APIクイックスタートガイドの [復習](#)
- FMC再利用可能な [オブジェクトの確認](#)
- Pythonリクエストライブラリの [基礎知識](#)

### 使用するコンポーネント

- REST APIが有効なREST API ( バージョン6.1以降 ) をサポートするFirepower Management Center
- Pythonを使用したREST APIインタラクション。

### 制限

- FMCでは、オブジェクトの名前が64文字を超えることはできません。
- オブジェクト名の先頭にスペースを使用し、末尾にセミコロンを使用することはできません。
- ペイロードには、1回の一括プッシュで1,000を超えるエントリを含めることはできません。

- 1回の一括プッシュでは、ペイロードサイズを2 MBを超えることはできません。

## 背景説明

ネットワークマネージャがネットワークの設定と管理に使用できる軽量でプログラム可能なアプローチにより、REST APIの普及が進んでいます。FMCは、任意のRESTクライアントを使用し、組み込みのAPIエクスペローラを使用して、設定と管理をサポートします。

このドキュメントの例では、CSVファイルを入力として取り込み、REST APIインターフェイス経由でオブジェクトをFMCにプッシュします。このドキュメントでは、ホストネットワークの一括プッシュのみを扱い、他のオブジェクトに対して同様のロジックを拡張できます。サンプルコードは、URLおよびPortオブジェクトのドキュメントに添付されます。

図に示すように、使用されるネットワークホストのPOSTのAPIリファレンスを次に示します。

**POST** /api/fmc\_config/v1/domain/{domainUUID}/object/hosts

Retrieves, deletes, creates, or modifies the host object associated with the specified ID. If no ID is specified for a GET, retrieves list of all host objects. *Check the response section for applicable examples (if any).*

Parameters Try it out

Name	Description
<b>body</b> * required object (body)	Input representation of host object.  Parameter content type application/json
bulk boolean (query)	Enables bulk create for host objects.  -
<b>domainUUID</b> * required string (path)	Domain UUID  e276abec-e0f2-11e3-8169-6d9ed49b625f

Responses Response content type application/json

Code	Description
201	Created  Example Value   Model  Request example 1 : POST /fmc_config/v1/domain/domainUUID/object/hosts ( POST to create a host object ) <pre>{   "name": "TestHost",   "type": "Host",   "value": "10.5.3.20",   "description": "Test Description" }</pre> Request example 2 : POST /fmc_config/v1/domain/domainUUID/object/hosts?bulk=true ( Bulk POST operation for Host object ) <pre>[   {     "name": "host1",     "type": "Host",     "value": "10.5.3.20",     "description": "Test Description"   },   {     "name": "Host2",     "type": "Host",     "value": "1.2.3.4",     "description": "Host object 2"   } ]</pre>

## 設定

ステップ1:REST APIを有効にし、認証トークンを生成します。設定手順と例の詳細については、『[FMCでの認証トークンの生成](#)』を参照してください。

```
import requests import csv import json from requests.auth import HTTPBasicAuth from getpass
import getpass address = input("Enter IP Address of the FMC: ") username = input ("Enter
Username: ") password = getpass("Enter Password: ") api_uri =
"/api/fmc_platform/v1/auth/generatetoken" url = "https://" + address + api_uri response =
requests.request("POST", url, verify=False, auth=HTTPBasicAuth(username, password)) accesstoken
= response.headers["X-auth-access-token"] refreshtoken = response.headers["X-auth-refresh-
token"] DOMAIN_UUID = response.headers["DOMAIN_UUID"]
```

**ステップ2:**提供されたCSVファイルをディクショナリに変換し、要求のJSONペイロードとして使  
用します。各オブジェクトタイプのサンプルCSVファイルがドキュメントに添付されます。

	A	B	C	D
1	name	description	type	value
2	Host-1	Host-1	Host	10.10.10.10
3	Host-2	Host-2	Host	10.10.10.1
4	Network-1	Network-1	Network	10.10.9.0/24
5	Host-3	Host-3	Host	10.10.10.2
6	Range-1	Rannge-1	Range	10.20.20.1-10.20.20.20
7				

```
csvFilePath = input("Please enter the CSV Filepath (For eg. : path/to/file/objects.csv) :) host
= [] with open(csvFilePath, encoding='utf-8-sig') as csvf: csvReader = csv.DictReader(csvf) for
rows in csvReader: if rows['type'] == "Host": host.append(rows) host_payload = json.dumps(host)
この段階のhost_payloadは、次の図のように表示されます。
```

```
[{ "name": "Host-1", "description": "Host-1", "type": "Host", "value": "10.10.10.10" }, {
"name": "Host-2", "description": "Host-2", "type": "Host", "value": "10.10.10.1" }, { "name":
"Host-3", "description": "Host-3", "type": "Host", "value": "10.10.10.2" } ]
```

**ステップ3:**前のステップで受信した入力から要求を作成し、ペイロードが空でない場合は要求を  
送信します。

```
host_api_uri = "/api/fmc_config/v1/domain/" + DOMAIN_UUID + "/object/hosts?bulk =true" host_url
= "https://" + address + host_api_uri headers = { 'Content-Type': 'application/json', 'x-auth-
access-token': accesstoken } if host != []: response = requests.request("POST", host_url,
headers=headers, data = host_payload, verify = False) else : print("Please Validate that the CSV
file provided is correct or at correct location")
```

## 確認

- 次に示すように、要求が成功したか失敗したかを確認するために、応答のステータスコードを印刷します。

```
if response.status_code == 201 or response.status_code == 202: print("Host Objects successfully
pushed") else: print("Host Object creation failed")
```

- FMCにログイン図に示すように、[Object] > [Object Management] > [Network]に移動し、ホストオブジェクトを確認します。

## Network

Add Network

A network object represents one or more IP addresses. Network objects are used in various places, including access control policies, network variables, intrusion discovery rules, event searches, reports, and so on.

Name	Value	Type
Host-1	10.10.10.10	Host
Host-2	10.10.10.1	Host
Host-3	10.10.10.2	Host

## トラブルシューティング

- RESTクライアントを使用すると、自己署名証明書が原因でSSL証明書の問題に関連するエラーが表示されることがあります。この検証は、使用しているクライアントに応じてオフにできます。
- FMC REST API認証トークンは30分間有効で、最大3回更新できます。
- 要求に関するエラーは、応答本文から抽出できる。これは、トラブルシューティングに役立つログファイルとして収集できます。

```
logfile = "requestlog.txt" log = open(logfile, "w+") log.write(response.text) log.close
```

- すべてのREST要求は、FMC上のこれらの2つのログファイルに記録されます。URLを検索します(例: ../object/hosts)を使用して正しい操作を行います(GET操作のエラーを探している場合は、ログがGET ...のように始まることを確認します。オブジェクト/ホスト)

```
tail -f /var/opt/CSCOpX/MDC/tomcat/logs/stdout.logs tail -f /var/opt/CSCOpX/MDC/log/operation/usmsharredsvcs.log
```