

# FMC REST API 상호 작용에 대한 인증 토큰을 생성하는 방법

## 소개

이 문서에서는 API(Application Programming Interface) 관리자가 FMC(Firepower Management Center)를 인증하고 토큰을 생성하며 추가 API 상호 작용에 사용하는 방법에 대해 설명합니다.

## 사전 요구 사항

### 요구 사항

다음 주제에 대한 지식을 보유하고 있으면 유용합니다.

- FMC(Firepower Management Center) 기능 및 구성([구성 가이드](#))
- 다양한 REST API 호출 이해([REST API란 무엇입니까?](#))
- FMC [API 빠른 시작 가이드](#)를 검토합니다.

### 사용되는 구성 요소

- REST API가 활성화된 REST API(버전 6.1 이상)를 지원하는 Firepower Management Center
- Postman, Python 스크립트, CURL 등의 REST 클라이언트

## 배경 정보

REST API는 네트워크 관리자가 네트워크를 구성 및 관리하는 데 사용할 수 있는 가벼운 프로그래밍 방식 때문에 점점 더 인기를 끌고 있습니다. FMC는 모든 REST 클라이언트를 사용하여 컨피그레이션 및 관리를 지원하며, 내장된 API 탐색기도 사용합니다.

## 구성

### FMC에서 REST API 활성화

1단계. System(시스템)>Configuration(구성)>REST API Preferences(**REST API 환경 설정**)>Enable REST API(**REST API 활성화**)로 이동합니다.

2단계. Enable REST API(**REST API 활성화**) 확인란을 선택합니다.

3단계. 저장을 누릅니다. 이미지에 표시된 것처럼 REST API가 활성화되면 [저장 성공] 대화 상자가 표시됩니다.

- Access List
- Access Control Preferences
- Audit Log
- Audit Log Certificate
- CLI Timeout
- Change Reconciliation
- DNS Cache
- Dashboard
- Database
- Email Notification
- External Database Access
- HTTPS Certificate
- Information
- Intrusion Policy Preferences
- Language
- Login Banner
- Management Interfaces
- Network Analysis Policy Preferences
- Process
- ▶ REST API Preferences

Enable REST API

## FMC에서 사용자 생성

FMC에서 API 인프라를 사용하는 모범 사례는 UI 사용자와 스크립트 사용자를 분리하는 것입니다. 다양한 사용자 역할에 대한 이해 및 새 사용자 [생성](#)에 대한 지침은 FMC용 사용자 계정 가이드를 참조하십시오.

## 인증 토큰 요청 단계

1단계. REST API 클라이언트를 엽니다.

2단계. POST 명령을 만들도록 클라이언트를 설정합니다.

URL: [https://<management center IP or name>/api/fmc\\_platform/v1/auth/generatetoken](https://<management center IP or name>/api/fmc_platform/v1/auth/generatetoken).

3단계. 사용자 이름과 비밀번호를 기본 인증 헤더로 포함합니다. POST 본문은 비어 있어야 합니다.

예를 들어 Python을 사용하는 인증 요청:

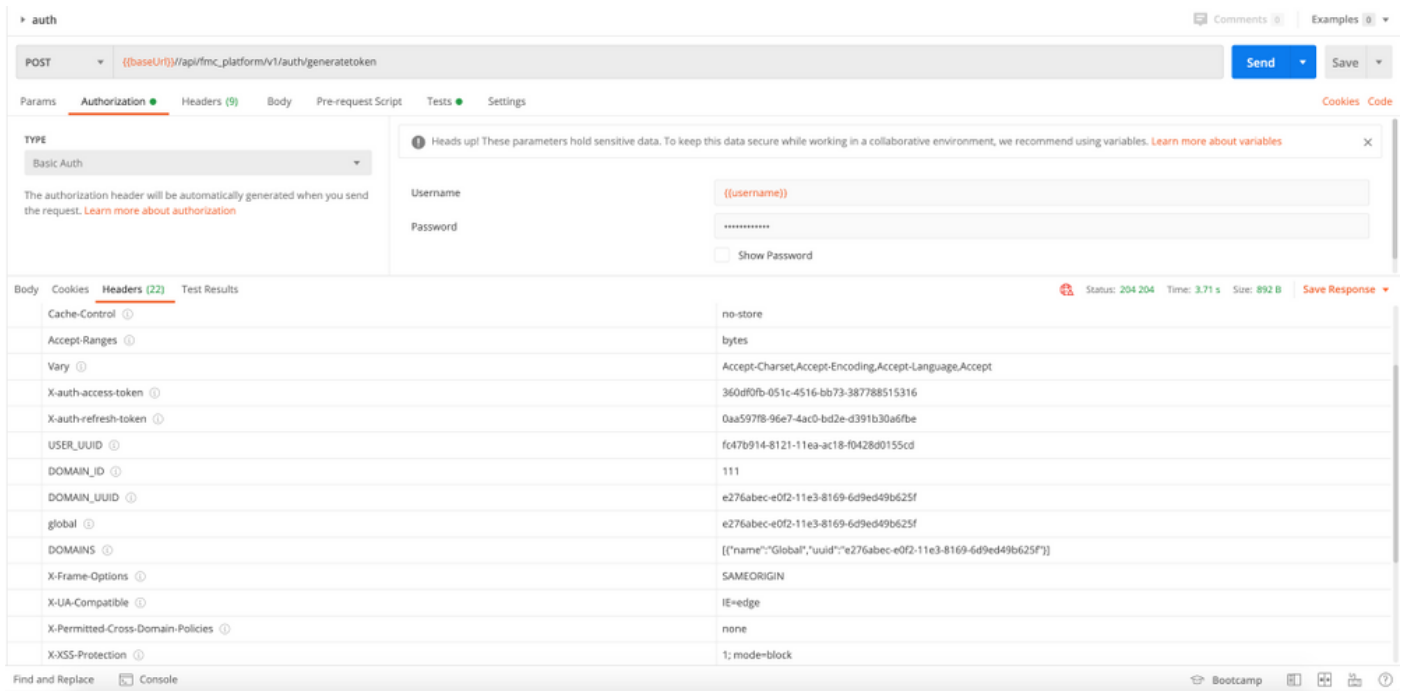
```
import requests
url = "https://10.10.10.1//api/fmc_platform/v1/auth/generatetoken"
payload = {}
headers = { 'Authorization': 'Basic Y2lzY291c2VyOmNpc2NwYXBpdXNlcg==' }
response = requests.request("POST", url, headers=headers, data = payload, verify=False)
print(response.headers)
```

CURL을 사용하는 인증 요청의 또 다른 예:

```
$ curl --request POST 'https://10.10.10.1/api/fmc_platform/v1/auth/generatetoken' --header
'Authorization: Basic Y2lzY291c2VyOmNpc2NwYXBpdXNlcg==' -k -i HTTP/1.1 204 204 Date: Tue, 11 Aug
2020 02:54:06 GMT Server: Apache Strict-Transport-Security: max-age=31536000; includeSubDomains
Cache-Control: no-store Accept-Ranges: bytes Vary: Accept-Charset, Accept-Encoding, Accept-
Language, Accept X-auth-access-token: aa6f8326-0a0c-4f48-9d85-7a920c0fdca5 X-auth-refresh-token:
674e87d1-1572-4cd1-b86d-3abec04ca59d USER_UUID: fc47b914-8121-11ea-ac18-f0428d0155cd DOMAIN_ID:
```

111 DOMAIN\_UUID: e276abec-e0f2-11e3-8169-6d9ed49b625f global: e276abec-e0f2-11e3-8169-6d9ed49b625f DOMAINS: [{"name": "Global", "uuid": "e276abec-e0f2-11e3-8169-6d9ed49b625f"}] X-Frame-Options: SAMEORIGIN X-UA-Compatible: IE=edge X-Permitted-Cross-Domain-Policies: none X-XSS-Protection: 1; mode=block Referrer-Policy: same-origin Content-Security-Policy: base-uri 'self' X-Content-Type-Options: nosniff

이미지에 표시된 대로 Postman과 같은 GUI 기반 클라이언트의 예:



## 후속 API 요청 전송

**참고:**출력에 표시되는 내용은 응답 본문이 아니라 응답 헤더입니다.실제 응답 본문은 비어 있습니다. 추출해야 하는 중요한 헤더 정보는 X-auth-access-token, X-auth-refresh-token 및 DOMAIN\_UUID입니다.

FMC에 성공적으로 인증하고 토큰을 추출하면 추가 API 요청을 위해 아래 정보를 활용해야 합니다.

- 헤더 X-auth-access-token <authentication token value>를 요청의 일부로 추가합니다.
- 토큰 새로 고침 요청에 X-auth-access-token <authentication token value> 및 X-auth-refresh-token<refresh token value> 헤더를 추가합니다.
- 서버에 대한 모든 REST 요청의 인증 토큰에서 Domain\_UUID를 사용합니다.

이 헤더 정보를 사용하여 REST API를 사용하여 FMC와 성공적으로 상호 작용할 수 있습니다.

## 일반적인 문제 해결

- 인증을 위해 전송된 POST의 요청 및 응답 본문이 비어 있습니다.요청 헤더에 기본 인증 매개변수를 전달해야 합니다.모든 토큰 정보는 응답 헤더를 통해 반환됩니다.
- REST 클라이언트를 사용할 때 자체 서명 인증서로 인해 SSL 인증서 문제와 관련된 오류가 표시될 수 있습니다.사용 중인 클라이언트에 따라 이 유효성 검사를 해제할 수 있습니다.
- 사용자 자격 증명은 REST API 및 GUI 인터페이스에 동시에 사용할 수 없으며, 두 인터페이스에 모두 사용할 경우 경고 없이 로그아웃됩니다.

- FMC REST API 인증 토큰은 30분 동안 유효하며 최대 3회 새로 고칠 수 있습니다.