

# Erstellen Sie benutzerdefinierte Geräte über die REST-API

Veröffentlicht: 2024-03-20

Sie können über die REST-API benutzerdefinierte Geräte erstellen, die den Netzwerkverkehr über mehrere IP-Adressen und Ports verfolgen. Möglicherweise möchten Sie beispielsweise für jede Zweigstelle ein benutzerdefiniertes Gerät hinzufügen. Wenn Sie die Geräte über ein Skript erstellen, können Sie die Geräteliste aus einer CSV-Datei lesen. In diesem Thema werden wir Methoden sowohl für die REST-API als auch für den ExtraHop REST API Explorer demonstrieren.

## Bevor Sie beginnen

- Sie müssen sich anmelden bei Sensor mit einem Konto, das über System - und Zugriffsadministrationsrechte verfügt, um einen API-Schlüssel zu generieren.
- Sie benötigen einen gültigen API-Schlüssel, um Änderungen über die REST-API vornehmen und die folgenden Verfahren ausführen zu können. (siehe [Generieren Sie einen API-Schlüssel](#).)
- Machen Sie sich mit dem vertraut [ExtraHop REST-API-Leitfaden](#) um zu erfahren, wie Sie im ExtraHop REST API Explorer navigieren.

## Erstellen Sie ein benutzerdefiniertes Gerät über den REST API Explorer

Sie können ein benutzerdefiniertes Gerät erstellen und das benutzerdefinierte Gerät mit einer Liste von IP-Adressen oder CIDR-Blöcken verknüpfen, indem Sie **POST /benutzerdefinierte Geräte** Betrieb.

1. Navigieren Sie in einem Browser zum REST API Explorer.

Die URL ist der Hostname oder die IP-Adresse Ihres Sensor, gefolgt von `/api/v1/explore/`. Wenn Ihr Hostname beispielsweise `seattle-eda` ist, lautet die URL `https://seattle-eda/api/v1/explore/`.

2. Klicken Sie **Benutzerdefiniertes Gerät**, und klicken Sie dann auf **POST /benutzerdefinierte Geräte**.
3. Geben Sie im Feld Eigenschaften für das benutzerdefinierte Gerät an, das Sie erstellen möchten.

Beispielsweise ordnet der folgende Text das benutzerdefinierte Gerät den CIDR-Blöcken `192.168.0.0/26`, `192.168.0.64/27`, `192.168.0.96/30` und `192.168.0.100/32` zu:

```
{
  "description": "The location of our office in Washington",
  "name": "Seattle",
  "criteria": [
    {
      "ipaddr": "192.168.0.0/26"
    },
    {
      "ipaddr": "192.168.0.64/27"
    },
    {
      "ipaddr": "192.168.0.96/30"
    },
    {
      "ipaddr": "192.168.0.100/32"
    }
  ]
}
```

## Rufen Sie das Python-Beispielskript ab und führen Sie es aus

Das ExtraHop GitHub-Repository enthält ein Python-Skript, das benutzerdefinierte Geräte erstellt, indem es Kriterien aus einer CSV-Datei liest.

1. Gehe zum [GitHub-Repository mit ExtraHop-Codebeispielen](#) und laden Sie die `create_custom_devices/create_custom_devices.py` Datei auf Ihrem lokalen Computer.
2. Erstellen Sie eine CSV-Datei mit Zeilen, die die folgenden Spalten in der angegebenen Reihenfolge enthalten:

Name	ID	Beschreibung	IP-Adresse oder CIDR-Block
------	----	--------------	----------------------------



**Hinweis** Die `create_custom_devices` Verzeichnis enthält eine CSV-Beispieldatei mit dem Namen `device_list.csv`.

Das Skript akzeptiert keine Kopfzeile in der CSV-Datei. Die Anzahl der Spalten in der Tabelle ist unbegrenzt. Jede Spalte nach den ersten vier gibt eine zusätzliche IP-Adresse für das Gerät an. Die ersten vier Spalten sind für jede Zeile erforderlich.

3. Öffnen Sie in einem Texteditor den `create_custom_devices.py` archivieren und ersetzen Sie die folgenden Konfigurationsvariablen durch Informationen aus Ihrer Umgebung:
  - **GASTGEBER:** Die IP-Adresse oder der Hostname des Sensor.
  - **API-SCHLÜSSEL:** Der API-Schlüssel.
  - **CSV\_DATEI:** Der Pfad der CSV-Datei relativ zum Speicherort der Skriptdatei.
4. Führen Sie den folgenden Befehl aus:

```
python3 create_custom_devices.py
```



**Hinweis** Wenn das Skript eine Fehlermeldung zurückgibt, dass die SSL-Zertifikatsüberprüfung fehlgeschlagen ist, stellen Sie sicher, dass [Ihrem Sensor oder Ihrer Konsole wurde ein vertrauenswürdiges Zertifikat hinzugefügt](#). Alternativ können Sie das hinzufügen `verify=False` Option zur Umgehung der Zertifikatsüberprüfung. Diese Methode ist jedoch nicht sicher und wird nicht empfohlen. Der folgende Code sendet eine HTTP GET-Anfrage ohne Zertifikatsüberprüfung:

```
requests.get(url, headers=headers, verify=False)
```